Softw Syst Model (2003) 2: 120–133 / Digital Object Identifier (DOI) 10.1007/s10270-003-0022-1

Modeling learning technology systems as business systems

Paris Avgeriou[1,*], **Symeon Retalis**[2], **Nikolaos Papaspyrou**[1]

[1] National Technical University of Athens, Department of Electrical and Computer Engineering, Software Engineering Laboratory, 15780 Zografou, Athens, Greece; E-mail: nickie@softlab.ntua.gr
[2] University of Cyprus, Department of Computer Science, 75 Kallipoleos Str., P.O. Box 20537CY-1678 Nicosia, Cyprus; Email: retal@ucy.ac.cy

Received: 27 September 2002/Accepted: 19 February 2003
Published online: 3 June 2003 –

Abstract. The design of Learning Technology Systems, and the Software Systems that support them, is largely conducted on an intuitive, ad hoc basis, thus resulting in inefficient systems that defectively support the learning process. There is now justifiable, increasing effort in formalizing the engineering of Learning Technology Systems in order to achieve better learning effectiveness as well as development efficiency. This paper presents such an approach for designing Learning Technology Systems and their most popular specialization, the Web-based Learning Systems, by modeling them as business systems, using business-modeling methods. The aim is to provide an in-depth analysis and comprehension of the Learning Technology Systems and Web-based Learning Systems' domain, that can be used for improving the systems themselves, as well as for building the supporting software systems. Our work is based upon the Learning Technology Systems Architecture standard of IEEE LTSC, on the empirical results of designing Web-based Learning Systems for university courses and on the practices of the Rational Unified Process and the Unified Modeling Language.

Keywords: Business model – Learning technology system – Unified modeling language – Rational unified process – Web-based learning systems – Open and distance learning – e-learning – Learning technology systems architecture

1 Introduction

Governments, authorities and organizations comprehend the potential of the Information Technology to transform teaching and learning, and envisage a knowledge-based future where acquiring and acting on information and knowledge is the primary operation of all learners. In order to facilitate the realization of this vision, the use of Learning Technology Systems is being exponentially augmented and broadened to cover all fields of the new economy demands. **Learning Technology Systems (LTS)** are learning, education and training systems that are supported by the Information Technology [1]. Examples of such systems are computer-based training systems, intelligent tutoring systems and Web-based Learning Systems. The latter, **Web-based Learning Systems (WLS)** are complex Learning Technology Systems that incorporate a variety of organizational, administrative, instructional and technological components [2, 3]. They are based on the state-of-the-art Internet and WWW technologies in order to provide education and training following the open and distance learning paradigm.

In [4] it is substantiated that Learning Technology Systems, and their specializations, like WLS, can be considered as *business systems* that are supported by special *software systems*, which automate some of their business processes. The term 'business system' refers to a complex system or organization that is comprised of **resources** (e.g. people, material, information, products) and **processes** (activities performed within the business), serves certain **goals** and is constrained by certain **rules** [5]. The amount and type of supporting software systems depends on the specific Learning Technology System and the business processes that are automated. For example a Web-based Learning System for a course in a university needs software systems that provide web-based learning courseware authoring and delivery, student and course data management, communication and collaboration services and so on. The architecture of these software systems should be tightly connected to the business environment, affect it and be affected by it in return,

* *Present address:* University of Cyprus, Department of Computer Science, P.O. Box 20537, 1678 Nicosia, Cyprus; E-mail: pavger@softlab.ntua.gr

within an Architecture-Business Cycle [6]. The problem is that, in the real world, the process of designing a Learning Technology System, which is an instructional design process, and the process of developing its supporting software systems, which is a software engineering process, are usually performed separately, leading to poor sustainability and robustness in change [4], poor integratability and zero reuse of business logic [5]. Especially in the case of software systems that support e-learning, experience has taught that many of the early attempts for developing such systems failed because they didn't understand the domain of technology-supported learning [7, 8]. In order to understand this domain one must take into consideration the instructional processes, the educational settings where these are applied and the learning environment in which these processes take place.

The solution that has been proposed over the past years is to model both the business system and the software system and establish an explicit link between the two models [5, 9–13]. **Business Modeling** is an engineering technique that aims not only at producing the correct set of requirements for supporting software systems but at improving the business systems themselves as well [5, 14, 15]. This paper presents an approach of modeling Learning Technology Systems and Web-based Learning Systems as business systems, by identifying and designing the concepts that characterize the domain, the information that needs to be handled, and the workflows between the 'players' in the domain. The goal of the paper is twofold: firstly to understand the domain of LTS and WLS and upon that understanding, initiate discussion for improvement in both domains; secondly to establish a sound and formal basis for building supporting software systems for WLS.

We model the generic domain of LTS because there is a lot of work taking place on the standardization of LTS from IEEE LTSC [`http://ltsc.ieee.org/`], ISO/IEC JTC1 SC36 [`http://jtc1sc36.org/`] and IMS [`http://www.imsproject.org/`]. Furthermore, we deal with WLS as well, since they are the most popular implementation of LTS nowadays. The methodology that we use in this paper is: first to define the business model for a Learning Technology System according to the IEEE LTSC Learning Technology Systems Architecture (LTSA) standard; and subsequently to extend this model or rather refine it for the domain of WLS so that the development of WLS can benefit from basing its foundations on the strong and commonly accepted background of international standards. Similar work of business engineering for a specific category of LTS has been performed in [4], where the focus is not on a single business model, but on a complete business architecture for Higher Education Institutions in the UK.

The structure of the paper is as follows: Sect. 2 presents the theoretical foundations of the paper, i.e. the IEEE LTSA draft standard, as well as key concepts of the discipline of Business Modeling. Section 3 deals with modeling of a Learning Technology System, as a business system, according to the IEEE LTSA standard. Section 4 describes the domain of WLS and extends on the previous business model, refining it for the case of a particular Web-based Learning System. Finally Sect. 5 contains some conclusions deduced from this research, as well as future work.

2 Theoretical underpinnings

2.1 Architecture of Learning Technology Systems

The largest effort on developing a Learning Technology System architecture has been carried out in the IEEE P1484.1 Architecture and Reference Model Working Group [`http://ltsc.ieee.org/wg1/`] which has developed a tentative standard: the Learning Technology Systems Architecture (LTSA) [1]. The LTSA deals with the Learning Technology System as a whole, encompassing human resources, infrastructure and learning resources as well as their interactions. The LTSA describes a high-level system architecture and layering for learning technology systems, and identifies the objectives of human activities and computer processes and their involved categories of knowledge. These are all incorporated into the 5 layers, where each layer is a refinement of the concepts in the above layer. It is noted that the LTSA specification has merely an Information Technology perspective and therefore is pedagogically-neutral, content-neutral, culturally-neutral and platform-neutral. This aims at keeping a generic perspective on LTS and leaving space for pedagogical notions to appear in a specific layer of the architecture: the Stakeholders' perspectives and priorities. The LTSA promotes the design and implementation of components and subsystems that are re-usable, cost-effective and adaptable by facilitating the development of configuration guidelines for general learning technology systems. Many systems can satisfy the requirements of the LTSA specification, i.e. conform to it, although they might not provide all the Layer 3 components, or have different organizations. Section 4 will present a conforming web-based learning system. Out of the five refinement layers of architecture specified in the LTSA, only Layer 3 (system components) is normative in this Standard, i.e. conforming implementations must declare a mapping between their components and Layer 3 components. The five LTSA Layers are the following:

Layer 1, "Learner and Environment Interactions" addresses the learner's acquisition, transfer, exchange, formulation, discovery, etc. of knowledge and/or information through interaction with the environment. This is the top layer that decomposes the system into environment, interactions, and the learner entity. Collaboration among learners is internal to the learner entity, i.e., the learner entity represents a collection of learners that collaborate among themselves.

Layer 2, "Human-Centered and Pervasive Features" addresses the human aspects of learning technology systems in terms of human-specific strengths and weaknesses. The human-centered and pervasive features are critical and have the highest instructional design risk because these human strengths and weaknesses greatly influence the design of LTS. There are five human-centered and pervasive features: (1) humans use multimedia (auditory, visual, and other sensory inputs, physical interactions, etc.) for information exchange; (2) humans are "unreliable" receivers of information; (3) humans are nomadic and frequently change teachers and institutions over a lifetime of learning; (4) humans are diverse, learn differently, and learn differently over time; and (5) humans are self-aware and can give advice about learning methods that work best for themselves. These five features are sufficient to generate the LTSA system components (LTSA layer 3) — there are no extra components outside the ones required for human-centered and pervasive features. It is noted that this Layer was considered out of the scope of the LTSA in its latest version, but we will use it for the scope of this paper, as it will identify the main LTS business processes.

Layer 3, "System Components" describes the component-based architecture, as identified in human-centered and pervasive features. The LTSA system components are:

- Processes: learner entity, evaluation, coach, delivery.
- Stores: learner records, learning resources.
- Flows: learning preferences, behavior, assessment information, performance and preference information (three times), query, catalog info, locator (twice), learning content, multimedia, interaction context.

Stores are system components used for storing and/or retrieving information; *Processes* transform their inputs to their outputs; *Flows* transfer information from one system component to another. The LTSA system components are utterly generic and map to virtually all learning technology systems. Actual implementations of learning technology systems may not fit these component boundaries exactly, but represent implementation variations.

Layer 4, "Stakeholder Perspectives and Priorities" describes learning technology systems from a variety of perspectives by reference to subsets of the system components layer. Each stakeholder can have different focus (represented by a selection of a subset of LTSA system components) and can have different priorities (represented by primary and secondary design issues). The presentation of 120+ stakeholders in the LTSA is performed in order to build consensus because stakeholders can verify that their concerns are being addressed in LTSA.

Layer 5, "Operational Components and Interoperability — codings, APIs, protocols" describes the generic "plug-n-play" (interoperable) components and interfaces of an information technology-based learning technology architecture, as identified in the stakeholder perspectives. The LTSA provides a common method for analyzing and describing these operational and interoperable components. The specification of actual coding, API, protocol, etc., standards is outside the scope of LTSA and should be performed in an implementation system

The added value derived from the abstraction-implementation layers, is that the five layers represent five independent areas of technical analysis, which makes it easier to discuss each layer independently of the others.

It is stressed here that a Learning Technology System is actually considered an abstract system, which can be implemented in numerous ways by emphasizing on certain components, according to specific concerns and interests. In our case an abstract Learning Technology System is implemented as a Web-based Learning System by putting emphasis on the components that relate to web-based open and distance learning. Expressed in object-oriented terms, the concrete class "Web-based Learning System" extends the abstract class "Learning Technology System".

2.2 Essentials of Business modeling

As in the case of software modeling, there in so silver bullet in business modeling either. Today there are several approaches, such as processes, methods, notations, tools which can be used for the purposes of business modeling. Some of the recent, major approaches are: the Rational Unified Process [9, 16, 17], and the Unified Modeling Language (UML) and especially the UML Business Modeling Extensions [18]; the ARIS framework which is comprised of four different methods, combines several notations and is supported by commercial CASE tools [19–21]; the approach proposed by Eriksson and Penker which defines an extension of UML for business modeling and establishes a method for designing a 'business architecture' using 5 different views [5]; several commercial tools that aid in performing business modeling in the context of Enterprise Resource Planning, such as the ones from SAP, BAAN, Oracle and PeopleSoft. For business modeling approaches of the past, readers are encourage to look at the empirical reviews of business modeling methodologies, process and tools in [22–24].

In this paper we adopt the business modeling concepts from the Rational Unified Process and the Unified Modeling Language. The Unified Modeling Language has been chosen because it is an industrial standard in software engineering, it is vastly adopted in the software industry and it is supported by a significant number of CASE tools. Furthermore the Rational Unified Process has been selected since it adopts the UML official standard and in particular the UML Business Modeling Extensions, thus supporting the readability of the designs and the effective communication between the development stakeholders. More importantly, the Rational Unified Process offers an explicit link between the business model and the software systems that support it and aids the development team into extracting the requirements of these software systems directly from the business modeling elements, as will be explained later on. Finally UML and the Rational Unified Process were preferred for their simplicity and cost-effectiveness, as opposed to the other approaches aforementioned.

In the Rational Unified Process, the business model specifies which business processes are to be supported by the software system that is being engineered. The concepts used to define the business system are [9, 16, 17]:

- **Business Use-Cases** or **Business Processes** – The processes performed within the business during which the state of business entities change. A business use-case defines what should happen in the business when it is performed; it describes the performance of a sequence of actions that produces a valuable result to a particular business actor. A business process either generates value for the business or mitigates costs to the business.
- **Business Actors** – they represent entities of the environment external to the business system that interact with it, i.e. put demands on it, or are interested in its output. The different types of "interactors" might be for example customers, suppliers, partners, potential customers (the "market place"), local authorities etc. Business actors might be humans, other business systems, other software systems and so on.
- **Business Workers** – they represent abstractions of humans that act within the business. A business worker interacts with other business workers and manipulates business entities in order to realize a business use-case.
- **Business Entities** – they represent artifacts handled or used by the business workers as they execute a business use case. Typically, a business entity represents a document or an essential part of a product. Sometimes it represents something less tangible, like important knowledge about a market or a customer.

The goal of Business Modeling is to define these concepts and show the relationships and interactions between them in models. Business Modeling is supported by two kinds of UML models [9, 16, 17]:

- **Business use-case model** – which describes the business processes of a business in terms of business use cases and business actors. Like the use-case model for a software system, the business use-case model presents a system (here the business) from the usage perspective and outlines how it provides value to its users (here its customers, partners etc.). This model is depicted in UML use case diagrams.
- **Business object model** – which is an interior model of the business and describes how each business use case is realized by a set of workers who are using a set of business entities. This model is depicted in UML class diagrams, activity diagrams and sequence diagrams.

The result of business modeling can then be used as input in defining the requirements of the supporting software systems, as aforementioned. The Rational Unified Process proposes a mapping between the elements of the business model, i.e. business actors, business workers and business use cases and the elements of the use case model for the software system, i.e. actors and use cases. This is performed as following:

- Each *business worker* may correspond to an *actor* for the software system, if the *business worker* directly interacts with the software system while working on the realization of a *business use case*. For each *business use* case that the above *business worker* participates in, there may exist a *use case* corresponding to the functionality performed by the *business worker*.
- If the software system automates a *business use case* in full extent, then the *actor* of the software system is not one of the *business workers* working for the realization of the *business use case*, but the *business actor* connected to that *business use case*. In this case, there is no *business worker* coming in between the *business actor* and the software system; instead the *business actor* interfaces directly with the software system.

It is noted that business modeling also includes several other activities, producing other artifacts, such as defining business goals and strategy, business rules, business architecture, business patterns etc. [5, 14, 16]. For the scope and objectives of this paper, and in order to keep its size manageable the business use-case and object models are adequate to describe the business model.

Business modeling naturally makes use of a modeling language, in order to visually depict the necessary concepts. As aforementioned, we have adopted the Unified Modeling Language [25, 26]. Although UML in its first years was used mainly for modeling software systems, it is also a very suitable language for business modeling. It has the ability to describe both the structural aspects of a business, the behavioral aspects, and the business rules that affect both structure and behavior. An advantage of modeling in a language such as UML is that it **visually** depicts functions and relationships that are usually difficult to visualize clearly. Furthermore UML is a standard notation, with an exponentially increasing use in software engineering, and a massive support by tools. So the same tools used for modeling the business can be used to model its supporting software systems. Finally UML follows the object-oriented paradigm, which is a well-proven and established technique for handling large and complex systems and thus offers a short learning curve. Figure 1

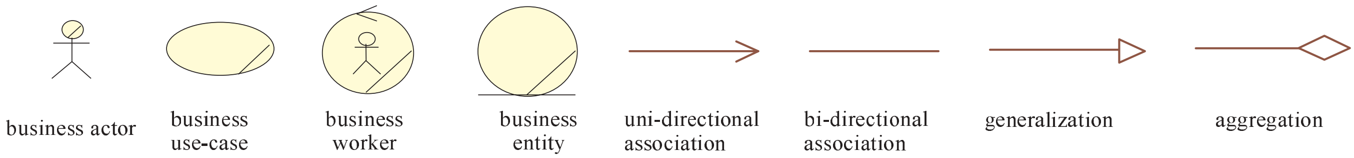

Fig. 1. Main symbols of UML business models

illustrates some of the most important symbols used in UML business models.

3 The business model of a Learning Technology System

This section combines the concepts identified in the LTSA draft standard, in order to model a Learning Technology System as a business system. The first question that arises is, which, if not all, of the 5 LTSA layers should be incorporated in the business model? Let' s take a look at the layers in order to survey which ones contain the business processes, actors, workers and entities:

- Layer 1, *Learner and Environment Interactions*, depicts the flow of information between the learner and learner environment, which is the generic business process in a Learning Technology System. It also provides the crucial information that the learner entity may be comprised of several collaborating learners.
- Layer 2, *Human-centered and pervasive features*, contains the human strengths and weaknesses that pervade the whole of the Learning Technology System. These human characteristics stimulate the actual business processes that a Learning Technology System has to carry out.
- Layer 3, *System components,* identifies the different constituents of the Learning Technology System that are derived by the human features. In business model terms, these system components are the business workers and the business entities handled by the workers, in order to realize the business use-cases.
- Layer 4, *Stakeholder perspectives and priorities*, shows different views of the LTSA depending on different stakeholders, and aims at verifying that all stakeholders' concerns are included at the LTSA. It does not define any new system components but only presents specific, stakeholder-dependant subsets of the Layer 3 components, showing where each stakeholder puts emphasis on. Therefore this layer is of no use to the business model.
- Layer 5, *Operational components and interoperability*, provides an overview of how technical standards can be related to LTSA and the development process that creates and harmonizes the technical work. The specification of actual coding, API, protocol, etc., standards is outside the scope of LTSA, hence there is nothing in this layer to incorporate in the business model.

We now move on to model a Learning Technology System, using a business use-case model that demonstrates the external usage of a Learning Technology System and a business object model, that show its internal operation in terms of business process realizations.

3.1 The business use-case model

The business use case model, as shown in Fig. 2 contains concepts from the first two layers of the LTSA. The learner entity is represented as a business actor, since it

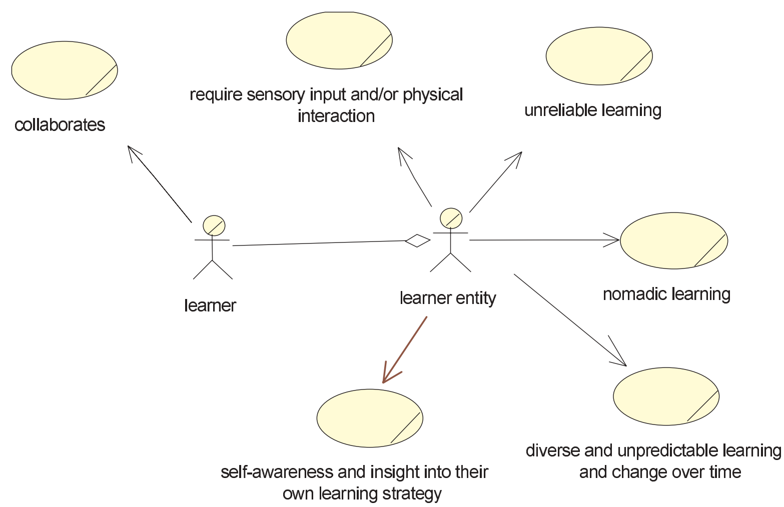

Fig. 2. The LTSA business use-case model

interacts with the Learning Technology System and receives value from it. The learner entity could also be considered part of the system, e.g. a business worker, and not an external actor, but the first layer clearly separates the learner entity from the 'learner environment', thus compelling the modeling of them as two different entities interacting with each other. It is repeated here that the LTSA is pedagogically neutral; therefore the learner entity-environment separation does not reflect a learning theory such as the learner-centered educational paradigm. The fact that the learner entity may be comprised of more than one learners collaborating is depicted in the "collaborates" business use-case that is stimulated by the "learner" business actor and provides value back to it. The 'learner' actor is itself a part of the learner entity. The 2^{nd} LTSA layer, provides the other 5 business use-cases that match the human-centered and pervasive features: "require sensory input and/or physical interaction", "unreliable learning", "nomadic learning", "diverse and 'unpredictable' learning and change over time", "self-awareness and have insight into their own best learning strategies".

3.2 The business object model

The business object model is comprised of the Layer 3 concepts, i.e. the System Components, and illustrates the realization of business use-cases, defined above, by business workers that work on business entities. The mapping between the 19 LTSA system components and the UML business model extension elements, was performed according to what the system components stand for in the Learning Technology System. The Learner Entity as aforementioned in the business use-case model, is a business actor as it instantiates the entity that interacts with the Learning Technology System. Furthermore, the 3 other processes in the LTSA, Coach, Delivery and Evaluation are represented as business workers, because they interact with each other and manipulate business entities. Finally the 2 stores and the 13 flows are signified as business entities, as they are the articles manipulated by the processes. It is emphasized that the following diagrams depict the exact LTSA system components in each use case realization as defined in the IEEE standard.

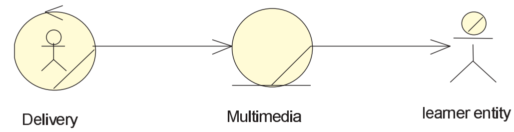

Fig. 3. The realization of the "require sensory input and/or physical interaction" business use case

The realization of the "require sensory input and/or physical interaction" business use case is depicted in Fig. 3, which implies that interactive multimedia is used for information exchange. This is the starting point of the LTSA and represents the *delivery* of information via *multimedia* to the *learner entity*. Multimedia includes audio/visual and other information, as well as physical interactions.

The realization of the "unreliable learning" business use case is depicted in Fig. 4, which shows that feedback systems may be required to avoid undesirable behavior and to target towards desirable behavior. It is called "the feedback and coaching loop", through which, the required learning experiences are maximized and the detrimental are minimized. The *learner entity* receives *multimedia* information from the *delivery* process and expresses some *behavior* that is assessed by the *evaluation* process. The *coach* may determine the "current position" from the *assessment information,* and in sequence decide on appropriate action (e.g., delivery of particular learning content) to achieve the desired target (pedagogical objectives). The coach may then send *locators* (e.g., references to lessons, experimentation tools, suggestions) to the Delivery system in order to achieve the new targets.

The realization of the "nomadic learning" business use case is depicted in Fig. 5, which illustrates that *learner records* need to be maintained, and accompany learners as they change teachers, coaches, and institutions over time. Typically, there will be more than one teacher

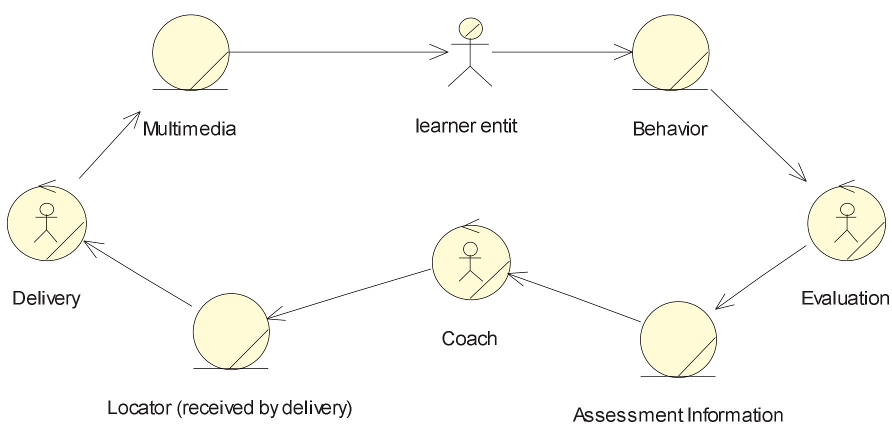

Fig. 4. The realization of the "unreliable learning" business use case

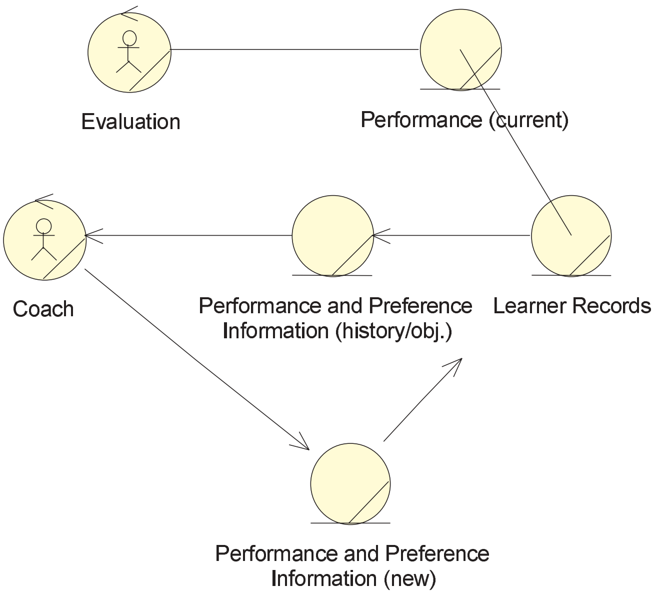

Fig. 5. The realization of the "nomadic learning" business use case

associated with a learner's lifetime of learning experience, so performance information is stored in Learner Records for the purpose of communicating it to other teachers. This way, each teacher can "pick up where the last left off", i.e., the next teacher (coach, etc.) minimizes the amount of observation (of behavior) and evaluation needed to determine where the learner "is at". Of course, learners, parents, and employers are interested in *past (historical), present,* and *future (objectives) performance information* because they can influence the learning experience, too. Similarly, *preference information* expressed by the learner herself (stored with performance information) can support better provision of learning experiences.

The realization of the "diverse and 'unpredictable' learning and change over time" business use case is depicted in Fig. 6, which exemplifies that rich learning resources need to be integrated into the feedback system, to support varying learning styles and strategies.

The *coach* performs *queries* to the *learning resources* repository and receives *catalog information*, i.e. appropriate metadata. He then chooses the suitable learning content from the catalog information and orders the *delivery* system to present them to the learner, by sending it a *locator*. The *delivery* process retrieves the actual *learning content* using a *locator* of its own and supplies it to the learner in multimedia format. Finally the delivery process provides *interaction context* to the *evaluation* process, so that the behavior that the learner expresses is properly evaluated. For example if the learner answers a multiple choice question, the evaluation process needs to know that the correct answer is no. 2, and this is codified in the interaction context flow.

The realization of the "self-awareness and have insight into their own best learning strategies" business use case is depicted in Fig. 7, demonstrating that provisions should be made to enable *learner entities* and *coaches* to negotiate the *learning styles, strategies, preferences* etc.

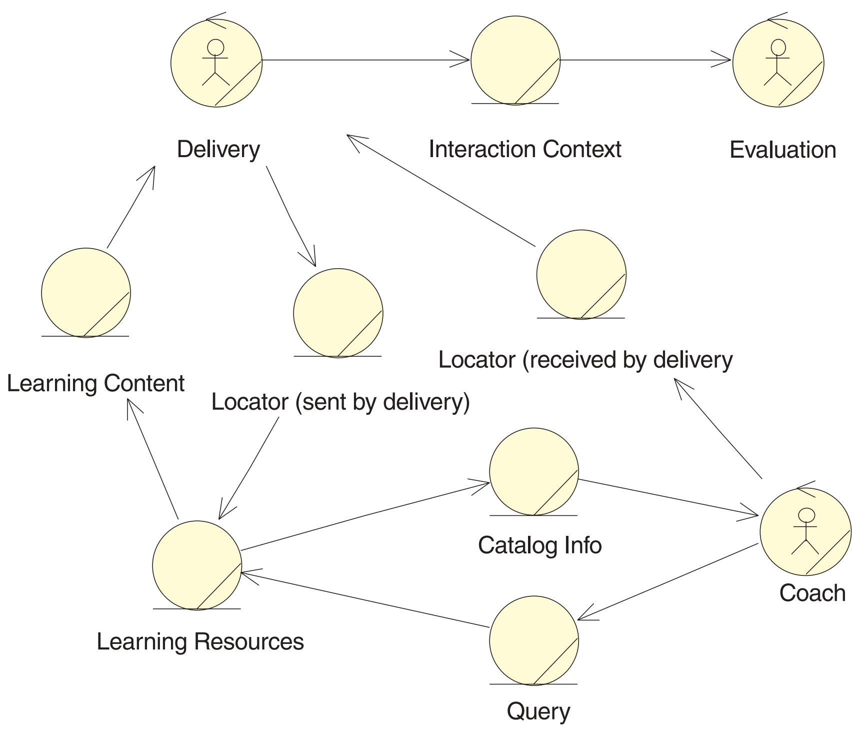

Fig. 6. The realization of the "diverse and 'unpredictable' learning and change over time" business use case

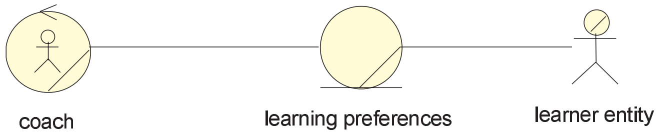

Fig. 7. The realization of the "self-awareness and have insight into their own best learning strategies" business use case

4 The business model of an LTSA-conformed implementation

WLS should be regarded as a set of three interrelated subsystems [27]:

- *The human subsystem*, which describes the roles, in as much detail as possible, for each kind of human agent involved in the instructional process [28].
- *The learning resources subsystem,* which is divided into web-based and conventional resources. Web-based resources can be course notes, slideware, study guides, self-assessment questionnaires, communication archives, learning material used for communication purposes, etc. The non web-based learning resources include textbooks, papers, audio/video cassettes, CDs, DVDs, etc.
- *The technical infrastructure subsystem,* which is divided into common and special. An instructional system basically makes use of services from common infrastructure, which is a set of *learning places*, that support student learning in general (e.g. laboratories, networking facilities, etc.). However, in order to best support the instructional process, special infrastructure should be created (e.g. multimedia conferencing systems, state of the art hardware components, a specific learning management system, etc.), which will provide services unique to a particular instructional problem [4].

This section elaborates on the business model of the previous section, by extending it for a specific Web-based Learning System that was designed to support the learning process in the subject matter of "Web Engineering" [27]. Sub-section 4.1, presents a brief description and analysis of the development of the Web-based Learning System, while sub-section 4.2 turns the previous informal description into a business model by refining the Learning Technology System business model.

4.1 The Web-based Learning System on Web Engineering

Our instructional design decision was to deliver this course in a *hybrid mode*, in the sense that: a) few lectures on specialized topics are given at lecture halls, b) few live lectures are given by distance using the facilities of a tele-conferencing seminar room, c) face-to-face tutorials are given when needed and d) asynchronous and synchronous guidance is provided via e-mail, discussion fora, on-line chat and tele-conferencing systems. The Web-based Learning System structure is shown in Fig. 8.

The human subsystem. The human agents and their roles in this course are shown in Table 1:

This hybrid/enriched way of teaching was designed to be learner-centered; the roles of the human agents of the system were based on the cognitive theories of instruction that emphasize the active exploration, construction and problem solving activities. The learners are expected to seek and choose from available information at their own pace, according to their own needs, and preferences and the instructor is merely a facilitator and a guide to the learning process.

The learning resources subsystems. The learning material for this course consisted of:

- Web-based learning resources
- E-book (electronic book) in the form of hypermedia course notes. Its structure follows the UK's Open University standards for structuring the learning material into blocks and units.

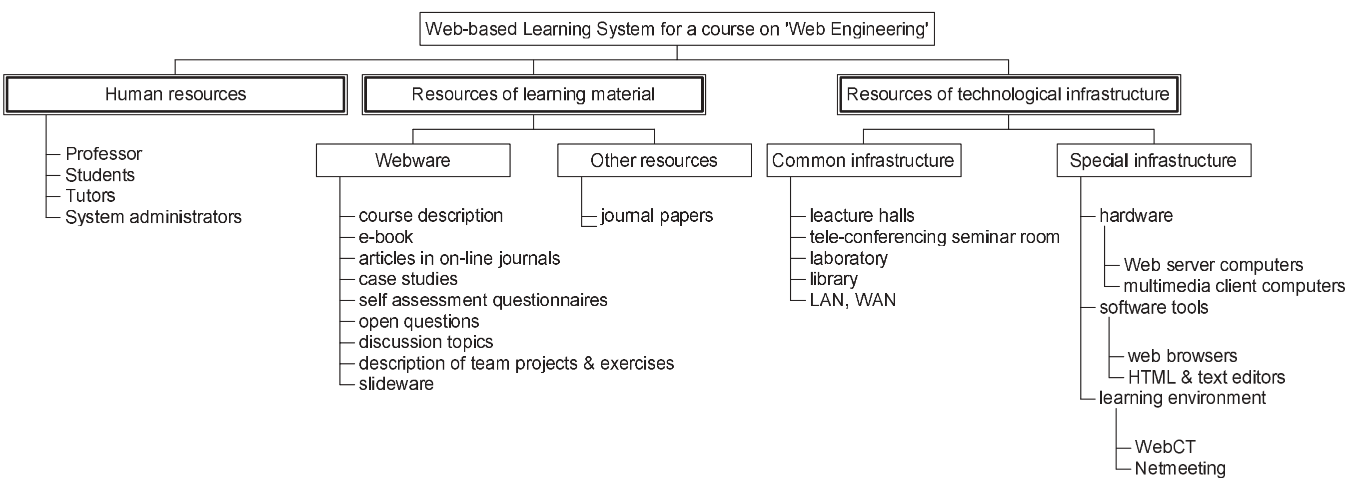

Fig. 8. The structure of the Web-based Learning System on Web Engineering

Table 1. Human agents' roles

Human agent	Roles
Learner	Attends Lectures Navigates freely within the learning resources Explores information resources Asks questions Collaborates with other classmates in team projects Seeks feedback on discussion topics Interacts with other learners either face-to-face or via e-mail
Instructor/tutor	Organizes content into learning resources Gives lectures either face-to-face or by distance Displays and updates information about the course Brings up discussion topics Provides corrective feedback (either face-to-face or computer mediated e.g. via e-mail) Advises and tutors students Assesses the students Monitors students' progress Creates and manages project teams
System Administrator	Administrates the technical infrastructure Administrates users' data Provides technical assistance

- Slideware
- Descriptions of the course and the team projects.
- Case studies of team projects from past years
- Discussion topics
- On-line journal papers
- Self-assessment exercises
- Other learning resources
- Journal papers

Technological infrastructure subsystem. The cornerstone of the special technological infrastructure subsystem was the WebCT learning environment [`http://www.webct.com/`]. This environment hosted the web-based learning resources, the details about students and instructors (personal data and records), and the data used for administration (course management). In addition it provided asynchronous communication facilities (asynchronous discussion fora and e-mail). Microsoft Netmeeting [`http://www.microsoft.com/windows/netmeeting/`] was used for synchronous tele-conferencing. An ordinary WWW browser (like Netscape Navigator, Internet Explorer) was adequate to browse through the material of the learning resources. The students could access the material either from the computers of a university laboratory or from their home or work provided that they had access to the Internet. Finally, for the purposes of their project assignments the students should also use the university library facilities or digital libraries.

4.2 The business model of the Web-based Learning System

The aim of this section is to produce the business model of the Web-based Learning System on Web Engineering by extending the business model of the Learning Technology System. The key idea of the Learning Technology System business model extension is that we leave the business use-case model intact, and we augment the business object model by defining new business entities and business workers that directly relate to Learning Technology System components. The business use-case model is left as is, because the generic business processes defined in it are global and comprehensive enough to cover our Web-based Learning System model. Therefore the extension relies on the business object model, by refining the existing business workers and entities and defining new ones that are directly associated to the Learning Technology System components. This is a practice recommended by the LTSA draft standard itself, in the sense that all conforming implementations must declare which of the system components they implement and in what way, irrelevantly of the business processes. In the following figures the WLS business workers and entities are denoted with a red color in order to distinguish them from the LTS business workers and entities.

The human subsystem. The human subsystem defined in the Web-based Learning System is depicted in Fig. 9. The *student* is considered to be a business actor, which interacts with the Web-based Learning System, and is a specialization of the LTSA Learner. The *professor* and the *tutor* are both business workers that extend the Coach process, perform some of the Evaluation process functions and manipulate the Performance (current), Learning Preferences and Assessment Information entities. Finally the *system administrator* is also a business worker

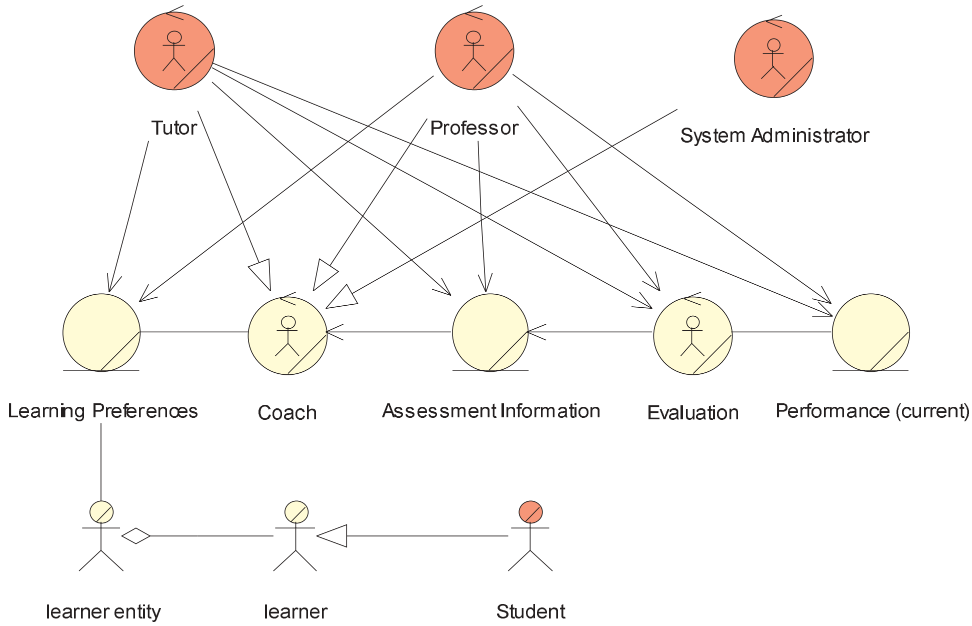

Fig. 9. The human subsystem related to the LTSA business model element

and a specialization of the Coach process as he/she performs some complementary administrative tasks.

The Learning Resources subsystems. The Learning Resources subsystem, depicted in Fig. 10 is comprised of *web-based courseware* and *conventional courseware*, which are both represented as business entities. The former, in turn, is divided into *content* (course description, e-book, articles in on-line journals, case studies, description of team projects and exercises, slideware) and *assessment* (self-assessment questionnaires, exercises, open questions). Both *content* and *conventional courseware* are a specialization of the Learning Resources store, while the *assessment* is a specialization of the Behavior flow.

Technological Infrastructure subsystem. The Technological Infrastructure subsystem, depicted in Fig. 11, is comprised of the *Common Infrastructure*, the *Special Infrastructure Except the LMS* and the *LMS* itself. The first two components are business workers and are considered part of the Delivery process. The *LMS* is also a business worker and the third part of the Delivery process. It is also responsible for the manipulation of the majority of the LTSA flows such as Interaction Context, Learning Content, Locator (both sent and received by Delivery), Multimedia, Behavior, Assessment Information, Performance (current), Evaluation, Query, Catalog Info, Learning Resources, Performance and Preference Information (both history and new) and Learner Records. It is obvious that the LMS is a major part of the Web-based Learning System and consequently encloses a great deal of its functionality.

Having finished associating the Web-based Learning System components with the LTSA ones, we can redesign the business use-case realizations of the Learning Technology System with the extra syntax and semantics of the Web-based Learning System components. Due to lack of space, we suggestively present as an example, only one of these realizations, the "require sensory input and/or physical interaction" use-case, as shown in Fig. 12.

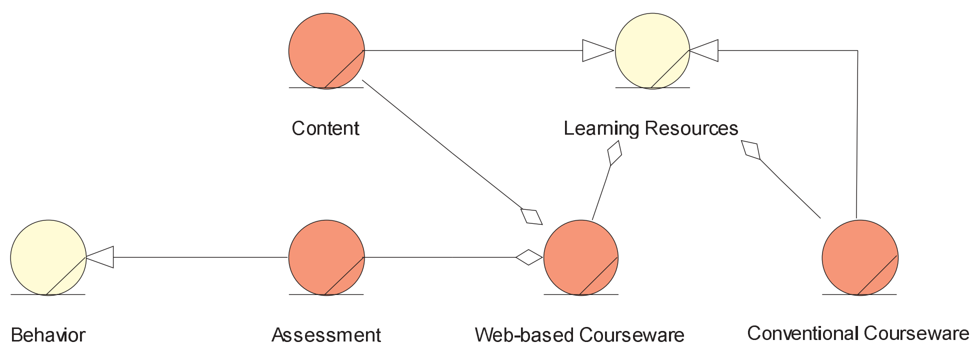

Fig. 10. The learning resources subsystem related to the LTSA business model elements

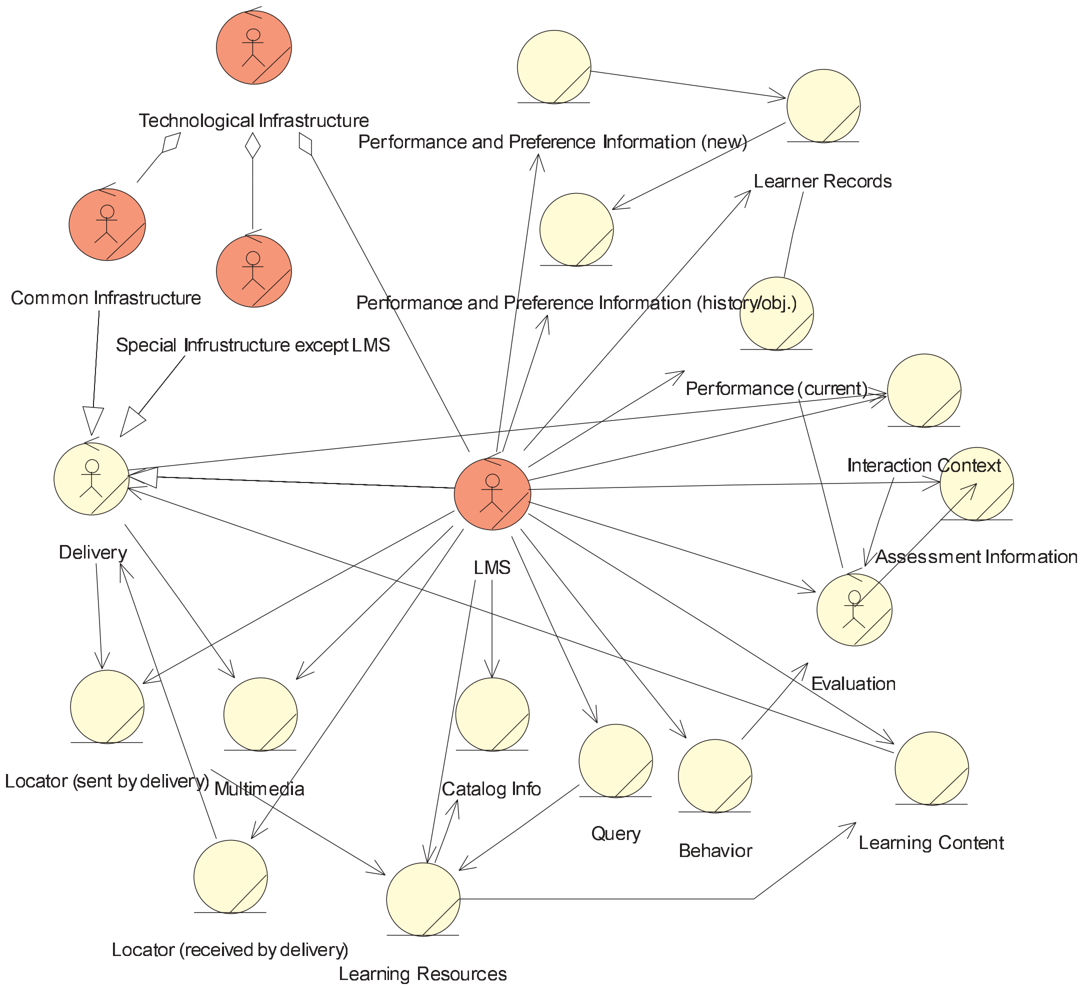

Fig. 11. The Technological Infrastructure subsystem related to the LTSA business model elements

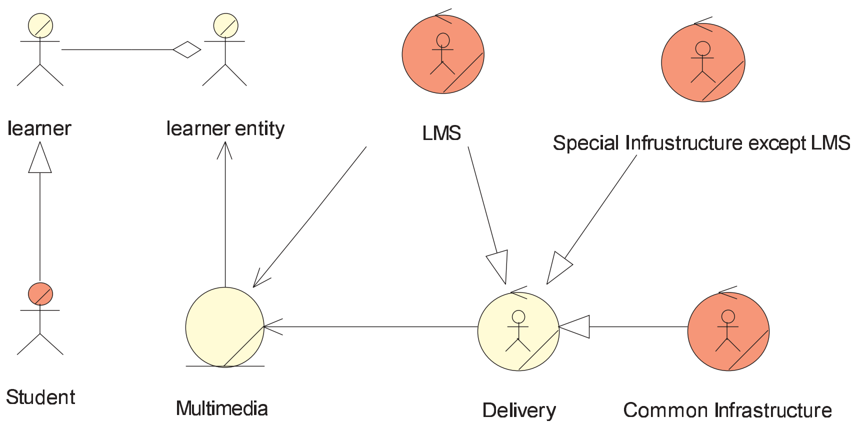

Fig. 12. The realization of the " require sensory input and/or physical interaction" use-case in the Web-based Learning System

5 Conclusions and future work

Modeling in software engineering has been an active research field for decades and never ceases to be timely and all-important. It is actually rather surprising that the modeling technique that was inaugurated in software engineering has revolutionized business engineering [14] with the concepts of Business Process Improvement or more radically with Business Process Re-engineering [29–32]. Business modeling assists in the comprehension and management of a business by modeling the actual business and its vision, objectives, processes and resources (human, material and technological). A business model is essentially a model, i.e. a simplified view of a complex reality, the business, and offers the following advantages [5, 16]:

- It is a means of creating an abstraction of the structure and the dynamics of a business, irrelevantly of its inherent complexity.
- It establishes a common understanding that can be communicated to the business stakeholders, facilitating discussion among them, helping them to reach agreement of the key fundamentals and to work toward common goals.
- It can act as a basis for business improvement or even business re-engineering, since existing problems and shortcomings are identified.
- It can be the basis for the development process of the various software systems that support the business.

This paper has presented an approach in modeling LTS and WLS as business systems. The innovative aspects of this idea are:

- The application of the business modeling discipline to the domain of Learning Technology Systems and Web-based Learning Systems, in consideration of international Learning Technology standards and with the aid of the Rational Unified Process and the Unified Modeling Language.
- The employment of these business models in designing the supporting software systems of LTS and WLS, as described in [33].

The advantages derived from the proposed approach are both in understanding and improving the business systems, and in building the software systems that support them. More specifically:

- We have documented in a solid and unambiguous way the structure and dynamics of WLS, a rather complex business system. This documentation can be communicated to all the stakeholders of the system, and act as a basis for understanding it per se, as well as its strengths and weaknesses and thus improving it. It also makes the business system maintainable and change-tolerant.
- By modeling WLS, we have created the essential business model to lead to the development of several supporting software systems: Learning Management Systems, Learning Portals, People and Institute Resources Management Systems, Collaborative Learning Support Systems, Assessment Management Systems and so on. Except for defining the requirements for these systems and providing them with a lot of vital information that increases their quality, the use of the same modeling language for both the business and the software model, increases the traceability between them. This means that a specific function in the software system can be traced back to a specific requirement in the business. Subsequently a change in the business model can more easily be propagated to the software model.
- We have achieved design reuse at two levels:
  - The business logic of LTS, as codified in the corresponding business model, can be reused in producing business models for specializations of LTS by extending the LTS business model. For example, having the LTS business model as a starting point we can extend it to design the business model for a computer-based training system or an intelligent tutoring systems or another Web-based Learning System. The added value is that we do not have to 're-invent the wheel' in specifying the human activities, computer processes, and system components involved in Learning Technology Systems.
  - The WLS business logic is reused in producing different supporting software systems. In this way, the software systems become an integrated part of the overall business supporting the business and enhancing the work and the results. They also integrate easily with each other and can share and exchange information.

The work presented in this paper is part of a research effort, concerning the development of a *Reference Architecture* for Learning Management Systems [33]. In particular, the business models described above, constitute a valuable input to the design of software architectures for Learning Management Systems. Future work in this area initially includes the identification of shortcomings in the WLS model in order to make improvements to the Web Engineering course. We also want to have more case studies of WLS business models, and in that sense we will design the business models for two more undergraduate courses of the subject areas of "Software Engineering", "Introduction to Compilers" and a post-graduate course entitled "Object-oriented Software Engineering Development". Finally a Learning Management System has already been developed and is being integrated into the aforementioned WLS, and will verify the correctness and efficiency of the "Web Engineering" course business model.

Acknowledgements. This paper is based on work supported in part by the National Technical University of Athens, under the Programme for the Support of Basic Research "Thalis". Project title "FL-ArchDesign: Formal Languages for Describing the Architectural Design of Software Systems".

References

1. IEEE Learning Technology Standards Committee: Draft standard for Learning Technology Systems Architecture (LTSA), Draft 9, November 2001
2. Moore, D., Kearsley, G.: Distance Education: A Systems View. Wadsworth Publishing Company, Belmont, CA, 1996
3. Carlson P.: Advanced educational technologies-promise and puzzlement. Journal of Universal Computer Science, 4(3): 210–216, 1998
4. Ford, P., Goodyear, P., Heseltine, R., Lewis, R., Darby, J., Graves, J., Sartorius, P., Harwood, D., King, T.: Managing

Change in Higher Education: A Learning Environment Architecture. Open University Press, London, 1996
5. Eriksson, H., Penker, M.: Business Modeling with UML, Business Patterns at work. Wiley Computer Publishing, New York, 2000
6. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice. Addison-Wesley, Reading, MA, 1998
7. Sun Microsystems: Understanding Distance Learning Architectures. A White Paper, 1999
8. Cisco Systems: Blueprint for Enterprise E-learning. A white paper, 2002
9. Kruchten, P.: The Rational Unified Process, An introduction. Addison-Wesley, Reading, MA, 1999
10. Bubenko, J.A., Wangler, B.: Objectives driven capture of business rules and of information systems requirements. Proceedings of International Conference on Systems, Man and Cybernetics. 'Systems Engineering in the Service of Humans', vol. 1, 1993, pp. 670–677
11. Krallmann, H., Derszteler, G.: Workflow Management Cycle. In: Scholz-Reiter, B., Stickel, E.: Business Process Modeling. Springer, Berlin, 1996
12. Avison, D.E., Fitzgerald, G.: Information Systems Development: Methodologies, Techniques and Tools. Blackwell Scientific Publications, Oxford, 1988
13. Valiris, G., Glykas, M.: Critical review of existing BPR methodologies. Business Process Management Journal 5(1): 65–86, 1999
14. Jacobson, I., Ericsson, M., Jacobson, A.: The object advantage: Business Process Re-engineering with object Technology. Addison-Wesley, Reading, MA, 1995
15. Nillson, A., Tollis, C., Nellborn, C.: Perspectives of Business Modeling. Springer-Verlag, New York, 1999
16. The Rational Unified Process, v. 2001.03.00.23, Rational Software Corporation, part of the *Rational Solutions for Windows* suite, 2000
17. Jacobson, I., Booch, G., Rumbaugh, J.: The Unified Software Development Process, Addison-Wesley, Reading, MA, 1999
18. Object Management Group: "UML Profile for Business Modeling". OMG Unified Modeling Language Specification, version 1.4, September 2001, pp. 4.9–4.17
19. Scheer, A.: Business Process Engineering, Reference Models for Industrial Enterprises. Springer-Verlag, Berlin, 1994
20. Scheer, A.: ARIS-Business Process Modeling. Springer-Verlag, Berlin, 2000
21. Scheer, A.: ARIS-Business Process Frameworks. Springer-Verlag, Berlin, 2000
22. Kettinger, W.J., Teng, J.T.C., Guha, S.: Business process change: a study of methodologies, techniques, and tools. MISQ Quarterly March 1997, 55–80
23. Wastell, G.W., White, P., Kawalek, P.: A methodology for business redesign: experience and issues. Journal of Strategic Information Systems 3(1): 5–22, 1994
24. Harrison, B.D., Pratt, M.D.: A methodology for reengineering businesses. Planning Review 21(2): 6–11, 1993
25. Booch, G., Rumbaugh, J., Jacobson, I.: The UML User Guide. Addison-Wesley, Reading, MA, 1999
26. Rumbaugh, J., Jacobson, I., Booch, G.: The UML Reference Manual. Addison-Wesley, Reading, MA, 1999
27. Retalis, S., Avgeriou, P.: Modelling Web-based Instructional Systems. Journal of Information Technology Education 1(1): 25–41, 2002
28. Lindner, R. (techn. ed.): Expertise and Role Identification for Learning Environments (ERILE). Proposed Standard Draft for German DIN, 2001
29. Hammer, M., Champy, J.: Reengineering the corporation. Harper Collins Books, New York, 1993
30. Champy, J.: Reengineering Management: The Mandate for New Leadership. Harper Collins Books, New York, 1995
31. Davenport, T.: Process Innovation – Reengineering Work through Information Technology. Harvard Business School Press, Boston, MA, 1992
32. Prasad, B.: Hybrid re-engineering strategies for process improvement. Business Process Management Journal 5(2): 178–197, 1999
33. Avgeriou, P.: A Reference Architecture For Open Learning Management Systems. PhD thesis, January 2003, National Technical University of Athens

Dr. Paris Avgeriou is a Lecturer at the Department of Computer Science, University of Cyprus. He received a diploma (MSc) in Electrical and Computer Engineering (1999), as well as a PhD in Software Engineering (2003) from the National Technical University of Athens (NTUA), Greece. He has worked as a research assistant and tutor at NTUA, has participated in a number of European R&D projects and has published several papers in international journals and conferences. He acts as a reviewer in journals and conferences and takes part in the international standardization activities of CEN/ISSS Learning Technology Workshop and ISO/IEC JTC1. His research interests concern the area of software engineering and particularly software and business modeling, with emphasis on software architecture. He also conducts research in quality assurance, web engineering and mobile clients. His research is mainly applied in the domain of e-learning.

Dr. Symeon Retalis is Lecturer at the Department of Computer Science, University of Cyprus. He holds a diploma of Electrical and Computer Engineer from the Department of Electrical and Computer Engineering, National Technical University of Athens, Greece. His MSc degree is on the Information Technology-Knowledge Based Systems from the Department of Artificial Intelligence, University of Edinburgh, Scotland. In 1998 he was awarded with a PhD from the Department of Electrical and Computer Engineering, National Technical University of Athens, Greece. His research interests lie on the web-based courseware development, hypermedia systems design, and web engineering. He serves in the editorial board of international journals and participates to the ACM Web Engineering special interest group and to the IEEE Learning Technologies Standardisation Committee.

Nikolaos S. Papaspyrou is a lecturer at the School of Electrical and Computer Engineering of the National Technical University of Athens (NTUA), Greece. He was born in Athens in 1971. He received a B.Sc. in Electrical and Computer Engineering from the NTUA (1993), a M.Sc. in Computer Science from Cornell University, Ithaca, NY (1995), and a Ph.D. in Com-

puter Science from NTUA (1998). His current research focuses on the theory and implementation of programming languages (semantics, type systems, compiler construction, functional, logic and intensional programming). Other interests include man-machine interface, educational software and learning environments. Dr. Papaspyrou has worked as a primary researcher in several R&D projects, as a short-term lecturer at the Department of Electronic and Computer Engineering of the Technical University of Crete and as a visiting postdoctoral researcher at the Department of Computer Science of Yale University, New Haven, CT.