Engineering Modular Web-based Education Systems to Support EML Units of Learning

Manuel Caeiro-Rodríguez, Luis Anido-Rifón, Martín Lamas-Nistal
University of Vigo, Telematic Engineering Department,
Campus Universitario, S/N Vigo, E-36310 Spain
{Manuel.Caeiro; Luis.Anido; Martin.Llamas}@det.uvigo.es
WWW home page: http://www-gist.det.uvigo.es/~mcaeiro

Abstract. Educational Modeling Languages (EMLs) have been proposed to support the modeling of units of learning enabling the description of different pedagogical approaches. Eventually, such models are intended to support the operation of appropriate computer systems, controlling and managing the corresponding units of learning. This paper considers EMLs involving a set of independent perspectives. Our purpose is to use this set of perspectives to facilitate the development of Web-based education systems that are able to support the execution of EML models. As a result, we obtain a modular architecture where different pedagogical approaches can be supported.

1 Introduction

Educational Modeling languages (EMLs) [3] were proposed to support the learning design (modeling) of units of learning. The main idea underlying these specifications is not to constrain the educational design by focusing on a certain class of instruction or pedagogical approach. Following this principle, the IMS *Learning Design* (LD) Specification [2] was proposed as a new e-learning standard. Eventually, the models of units of learning will be processed by appropriate computational systems to provide the intended educational experiences. As a result, EMLs offer a promising view for the future development of web-based educational systems, as they enable the separation of pedagogical and technological concerns: educational authors (e.g. teachers) could focus on the modeling of units of learning, while technology developers (e.g. software engineers, programmers) could centre on the provision of computer applications.

This paper proposes a component-based architecture for web-based education systems that support the execution of a particular kind of EML model. Our proposal

involves a meta-model based on a separation of concerns approach, to support the design of units of learning in a set of more or less independent perspectives.

2   Perspectives in educational processes

To deal with the complexity of the whole modeling of units of learning, we propose to decompose this problem into a number of partial models, each one focusing on a certain issue, namely perspectives. A perspective is defined as a feature that involves a certain purpose and that can be analyzed independently. The first perspectives identified in educational processes are provided by the answers to the questions *who?*, *what?*, and *when?*: (i) *organizational perspective* (who are the participants (learners and staff) involved in each activity? What roles do they play? How are they organized?); (ii) *functional perspective* (What are the goals to be achieved in each activity? What are the participants intended to do? Do they create documents? Do they transfer information to other participants?); (iii) *process perspective* (When are activities intended to be attained? In which order are tasks sequenced?). Each perspective is concerned with a concrete feature. Following this approach we have identified twelve perspectives. Six of them have been directly taken from the workflow domain: *functional, process, information*, *operational*, *organizational* and *resource*. In addition, six more perspectives have been considered to support collaborative and educational process modeling: *causal, temporal, authorization, interaction, awareness* and *manual intervention*. To obtain a more complete description of the perspectives see [1].

3   The architecture

In this section we present the main features of a component-based architecture for the development of adaptive web-based education systems that support such a meta-model. The proposed architecture (c.f. Figure 1) is composed of three main blocks following a typical three-tier scheme: a *Web Engine*, an *EML Execution Engine* and a *Web Client*. In addition, there is a *Setup and Administration* function devoted to the management of the educational resources and the instantiation of the elements.

3.1   The web engine

The *Web Engine* plays the role of the back-end tier, providing backup and supporting functionalities. More specifically:
− The *Data Base* provides a persistence service maintaining administrative information such as the name, password and profile of each participant (e.g. preferences, portfolio data), the documents and contents managed by the participants, information about the active units of learning and their models, properties and variables, etc.
− The *Security Manager* provides temporal authentication certificates to the participants to enable their access to the available applications. This service is

required to control the access of participants only to the applications and resources assigned to the instance of the unit of learning they are performing, and not to the applications of other instances.

− The *Rule Manager* provides Event-Condition-Action (ECA) rules management to support dependencies in the perspectives. This provides basic event functionality enabling event propagation, filtering, processing, etc.
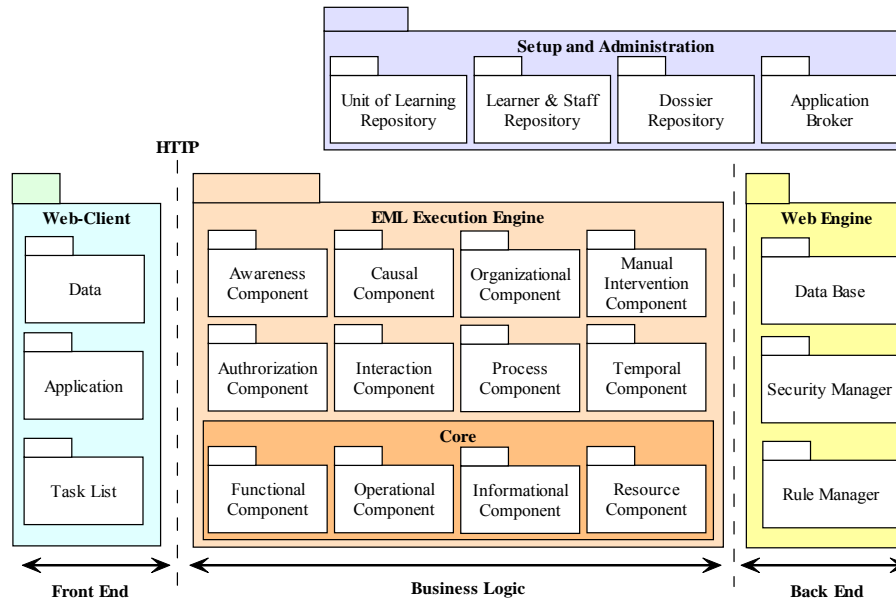

Figure 1. Main components of the proposed architecture

3.2   The EML execution engine

The *EML execution engine* is the logic-tier responsible for managing the educational practices accordingly to the learning designs (education models). This engine determines the activities that can be performed by each participant, the tools and resources available in each activity environment, the conditions to use such tools and interact with other participants, etc. All this management and control functionality is considered in separate components in accordance with the perspectives identified.

The proposed architecture considers a core set of components and several extension components. At the core of the architecture several components provide basic educational functionality in accordance with a set of perspectives required to enable a minimum support. Additional components implement features concerned with specific perspectives that are not strictly required for a basic functionality. This design localizes the changes required to tailor a web-based educational system to the component that implements it. Additionally, the same perspective may be supported by different components providing different degrees of support. For example, the

Process Perspective involves several categories of control flow, ranging from basic to advanced control structures. A system may be upgraded by replacing a component at a basic level by another component with more comprehensive support.

The core. The engine architecture adopts a minimalist approach. At the core of the system, four separate components allow instructional designers to define and execute educational practices modeled in accordance with a basic set of perspectives, required to provide minimum support:

− The *Functional Component* provides the basic mechanism to support and control the execution of activities. This is the main component of the architecture, as it maintains the execution state of educational practices and interacts with other components to support functionalities managed in the corresponding perspectives.
− The *Resource Component* is concerned with the assignment and management of the participant or participants responsible for their achievement of the activity.
− *The Informational Component* is devoted to provide and manage the data and artifacts to be used in the activity.
− The *Operational Component* is intended to manage the integration of external applications and services. It enables participants to access the applications and services that may be used in educational activities.

The extensions. These components provide the functionalities identified in the other perspectives that are not strictly required in some educational practices:

− The *Authorization Component* is concerned with the permissions (e.g. authorizations, prohibitions, obligations, dispensations) that are assigned to participants to use or access resources, applications, etc.
− The *Interaction Component* is devoted to the management of the interaction between participants and services in communications and collaborations.
− The *Process Component* is used to control the order in which the activities may be accessed by participants. Activities are initiated by the *Functional Component* but the order in which they are proposed may be controlled by this component.
− The *Temporal Component* manages the time dependencies that affect the enabling of activities. For example, controlling that a group of activities are initiated at the same time.
− The *Awareness Component* provides information about the status of running educational processes. This component separates *how* the system tracks the execution from *what* it does with this information. In this way, educational designers can add awareness specifications only when their educational practice needs this functionality.
− The *Causal Component* provides information about the purpose and features of the educational practice. It is descriptive information such as meta-data, learning goals, pre-requisites, etc.
− The *Organizational Component* enables the support of groups, the arrangement of participants, the maintenance of data about participants, etc. Its main purpose is to maintain information that may be used by other components.
− The *Manual Intervention Component* allows authorized users to make decisions on the modeling of the unit of learning while it is being executed. This component provides the mechanisms for stopping, rewinding and resuming running processes, etc. In effect, it is devoted to support dynamic and emergent behavior.

3.3   The web client

The *Web Client* provides the front-tier where the participant interacts with the system. This interface is controlled by the *EML Execution Engine* (through the *Functional Component*) to present the available activities, tools and information. We have identified the following sections in the *Web Client*:

− A *Task List* is required to present the activities that are proposed to the participant. Task lists may present different representations (e.g. a dynamic tree, a list), purpose (e.g. pending tasks, performed tasks), etc.
− The *Data* section provides an appropriate environment to support data access and operation by participants. During an educational practice different kinds of data or information (e.g. personal properties, variables) may be used by participants that gain access to them through this section
− The *Application* section is concerned with the rendering and control of the applications used during educational practices in the activity environment.

4   Conclusions

This paper presents a component-based architecture for the development of web-based education systems. The architecture presents a component-based approach based in the separation of modeling concerns or perspectives. This approach has several advantages over traditional web-based educational systems. First, its simplicity makes it easier to understand the system. Second, each component encapsulates a design decision. This facilitates the customizing or replacing of individual components. For example, developers can extend the process component. Thirdly, it is possible to add advanced features by adding new components.

Acknowledgement

References

1. M. Caeiro-Rodríguez, M. Llamas-Nistal, L. Anido-Rifón, Modeling Group-based Education, *Conceptual Modeling – ER* (Springer-Verlag LNCS, Klagenfurt, Austria, 2005).

2. R. Koper, B. Olivier and T. Anderson (Editors.) *IMS Learning Design Information Model*, IMS Global Consortium (2003).

3. A. Rawlings , P. Van Rosmalen, R. Koper, M. Rodríguez-Artacho and P. Lefrere, CEN/ISSS WS/LT Learning Technologies Workshop: Survey of Educational Modelling Languages (EMLs) (2002).