A MDA-based framework for building interoperable e-learning platforms

Nathalie Moreno[1] and **José Raúl Romero**[1]

[1] Universidad de Málaga, E.T.S. Ingeniería Informática, Campus de Teatinos, 29071 Málaga, Spain

The increasing interest in new technology trends has promoted the diffusion of e-learning systems in the academic and enterprise fields. Different technological and organizational proposals have emerged for the development of these systems. However, this implies great effort and business investments which should be preserved. With this purpose, new standard specifications have appeared, such as SCORM or IMS, but the current lack of integration among already existing e-learning platforms is sometimes critical despite their providing similar functionalities. To address this issue, we propose a framework that allows us to separate the system specification into different complementary perspectives. Moreover, this framework isolates the final development process (technology) from the system modelling by using the MDA (Model Driven Architecture) concepts provided to capture the system functionality and organize the specification of complex e-learning systems.

Keywords MDA; separation of concerns; framework; interoperability; reusability;

1. Introduction

The increasing interest in new technology trends has promoted the development of numerous e-learning systems. These interoperate at hardware and software level constrained by development agreements that allow users to operate different components and concepts defined on their corresponding technologies. However, the need to achieve a more efficient interoperability mechanism, which preserves business investments, has motivated the appearance of new standards like SCORM[1], IMS[2], AICC[3], etc. Nevertheless, many e-learning systems are still developed following the current technology trends without considering which interoperability facilities they should offer. In consequence, each solution provides different functionality according to its specific architecture and development approach.

One of the common ways of dealing with the inherent complexity of specifying large systems is by dividing the design activity into a number of areas of concern (*separation of concerns*), each one dealing with a specific aspect of the system [4]. These areas of concerns enable different abstraction viewpoints, allowing participants to observe a system from different suitable perspectives. Recently, OMG promoted MDA [5, 6] as a new approach for the development of systems with a great initial effort to specify functionalities and behaviour. MDA is mainly based on *models* consisting of a set of elements that describe some physical, abstract or hypothetical reality. MDA is originally based on the *separation of concerns* and establishes a clear distinction between modelling and implementation details. In this way, MDA tries to reach three goals: portability, interoperability and reusability.

Following MDA, each model should be based on a meta-model, which defines the specific language for an application domain. The most extended meta-model is UML. MDA provides three different types of models, according to the viewpoint from which the system is observed:

- *Computation Independent Model* (CIM), which is focused on the domain modelling by hiding structure details.
- *Platform Independent Model* (PIM), which provides the proper functionalities, structure and behaviour of the system.

- *Platform Specific Model* (PSM), which combines the PIM specifications with specific details concerning the way in which the system uses a particular kind of platform..

MDA defines *transformation models* as the process of converting from one model to another of the same system. Therefore, there is a *transformation engine* that applies a series of *transformation rules* on a *source model* to generate a *target model*.

Our proposal consists of a framework (called e-MDA) that tries to achieve the advantages provided by the separation of concerns into two fields: *(a)* dividing the system into different perspectives that allow us to isolate specific design aspects; and *(b)* making use of the mechanisms provided by MDA to generate e-learning applications from models, independently of the platform in which they will be implemented later.

After this brief introduction, this document is divided as follows. Section 2 describes the fundamentals of the e-MDA framework and introduces the prescribed perspectives. Section 3 details how MDA may influence the development process of e-learning systems. Finally, Section 4 describes some concluding remarks and future work.

2. The e-MDA framework

Actually, the benefits of considering different areas of concern for the specification of complex systems is well-known. In fact, current software architectural practices define several distinct viewpoints of systems in order to accomplish such specification decomposition. Examples include the viewpoints described in IEEE Std.1471 [7], the "4+1" view model [8], the Zachman's framework [9] or RM-ODP [10]. Based on the *separation of concerns principle* [4], our framework tries to help to organize the specification of complex e-learning systems by separating the different issues that matter for model-driven e-learning application development. These issues are addressed following different *perspectives,* which define different points of view of the same system as well as their own specification language and rules.

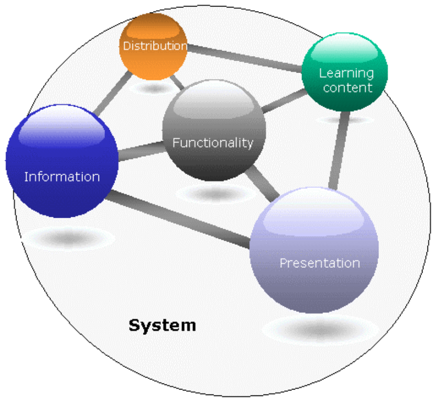

Figure 1. e-MDA framework perspectives

In fact, every perspective should be sufficiently independent to simplify reasoning about the complete specification of the system. The mutual consistency among the framework perspectives is guaranteed in our approach by means of restrictions. In this sense, OCL is used for constraining perspective models and for specifying invariants and pre & post-conditions.

As shown in Figure 1, for the development of interoperable e-learning platforms, e-MDA provides five different viewpoints, each one corresponding to a perspective: *functional*, *presentation*, *distribution*, *learning content* and *information*. Each perspective addresses one concern and comprise a set of models defined in terms of the entities that are relevant to that concern, and the relationships between them. Consequently, an e-learning application in our approach distinguishes five main PIMs, one for each perspective.

Functional Perspective. This perspective encapsulates the application's business logic, i.e., how the learning contents are processed, and how the application interacts with other computerized systems. It describes the major classes or component types representing services in the e-learning system. The design of this perspective is driven by the needs of the processes that implement the business logic of the system, taking into account the tasks that end-users (administrators, teachers, students, etc) can perform. This model is completed with a precise description of the behaviour of every component as well as the set of activities that are executed in order to achieve an e-learning objective. Consequently, the functional perspective defines the system functional requirements, objectives and constraints as well as the functional decomposition – in terms of subsystems, components, connectors, etc. – and behaviour. This specification is independent of distribution.

Presentation Perspective. This perspective is focused on the facilities provided to the end user for accessing and navigating through both the information managed by the application and the e-learning contents, which are just views of the data described by the Information perspective. The presentation perspective considers how this information should be structured, accessed and presented to facilitate end-user processes depending on the context and the user profile. Since each user looks at the same information and navigates through the application in a different way, all possible navigational structures for starting or for reaching available information have to be defined. We can also add constraints to those paths describing which events trigger the navigation.

Distribution Perspective. Current software systems are highly open and distributed, which implies that platforms should provide the mechanisms needed for allowing the dislocation, autonomy, mobility, evolution and heterogeneity of their functional elements. Thus, the distribution perspective tries to manage the physical decomposition of these elements, since both information and processes can be fragmented in nodes or replicated in different locations. In that way, we have the ability to reuse distributed content repositories and/or services.

Learning Content Perspective. This perspective defines the structure of the specific information concerning the learning process. In this way, the specification of reusable learning objects is possible (not only text but also images, video and links to other related content or documents), content models (e.g., XML and XSL templates), course structures, etc. From our framework point of view, each course consists of a collection of independent modules with well-defined entrance and exit levels of knowledge. In this sense, one of the most important aspects for academic e-learning is to assist teachers in selecting quality contents that will depend on user's needs and domains. To define the order of the appearance of the contents, teacher should establish a certain criteria and model a graph of dependencies between modules. Finally, we want to point out that appropriate utilities for the development and the design of new contents are not considered here. On the contrary, they should appear as services in the functional perspective.

Information Perspective. The information perspective describes the structure of the persistent information managed by the e-learning applications without considering those contents specifically designed for the learning process (e.g., the course organization). The information perspective is represented in terms of the data elements that comprise the application-related information and the semantic relationships between them. Moreover, it also contains the invariants imposed on data elements.

3. Developing platforms in the MDA context

Following the MDA approach [5, 6], a system's perspective is actually specified by a set of PIMs, which do not include features of a platform of any particular type. These PIMs relate to each other to specify how different subsystems (content management system, content access system, evaluation system, etc.)

can be combined and finally incorporated in the e-learning platform. Moreover, PIMs of different perspectives are also related by means of correspondences, so the complete view of the system is obtained independently of how it will be implemented in the future.

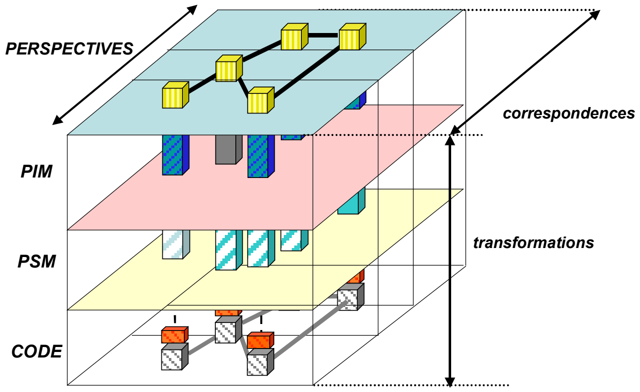

Figure 2. Applying the MDA chain to the development process of e-learning systems

By adopting the MDA strategy, our framework can be used both to build e-learning systems from scratch, and to build e-learning systems based on existing models defined in previous projects. In this paper we will address the first option. Though the description of the detailed methodology supported by the framework is outside the scope of this paper, its basic steps (see Figure 2) can be summarized as follows:

- From the system requirements, we need to identify which framework perspectives need to be instantiated. In this step, both the framework models and the constraints on them can be adapted to particular application sub-domains, implying in many cases extensions or simplifications of the original model notation/concepts, in order to accommodate to the particular system requirements.
- Because perspective is implemented using a different technology, we need to define at least three PIMs, one for each perspective (one PIM for the Functional Perspective, one PIM for the Presentation Perspective, one PIM for the Distribution Perspective, one PIM for the Learning Content Perspective, and one PIM for the Information Perspective).
- Once the PIMs of our system are described, we need to provide some sort of support for their deployment, configuration and execution in a particular platform, i.e., we need to generate their respective PSMs. This is a known process, well documented for instance in [a, b,c, d]. The process of generating a set of PSMs from the system's PIM is guided by a set of model transformations. The only special attention that needs to be paid when defining the transformations is to preserve the bridges between the PIMs, represented by the dependency relationships in our framework.
- Finally, the process of transforming the PSMs to code is also addressed in [a,b,c,d]. As a matter of fact, the models produced do not usually contain all the information required to produce an implementation (e.g., normally some behavioral information is missing). Of course, the more information the PIMs and the transformations contain, the better. However, in our experience we have seen that additional information needs to be provided later.

4. Conclusion remaks

At the beginning of this paper we mentioned some of the specific problems that e-learning systems exhibit. This paper presents a scalable perspective-based architectural framework for e-learning system design that helps solve some of these drawbacks. The framework follows the separation of concerns and the MDA principles, and can serve to guide the modular development of e-learning systems. The use of different perspectives (distribution, information, e-learning content, etc.) guarantees flexibility in the configuration of functions to match industrial objectives and priorities and pedagogic strategies. Furthermore, the framework has been designed to be neutral, generic and extensible, and can also be used as a common framework where current e-learning development platforms could be compared. In consequence, one of the benefits of designing the framework using the MDA principles is that we can easily provide a smooth integration with the rest of the e-learning platform proposals using either their models, their compilers, or both. Besides, generated models (PIMs) are not tied to any architectural style and are independent from many aspects and concerns. Finally, our approach is extensible in the sense that new concerns can be easily defined and integrated into the framework.

Ongoing research on the framework is focused on three main areas. In the first place, we plan to validate it with some medium to large applications, beyond the simple examples that have served to initially validate it. Secondly, we are finishing the complete definition of the methodology for the systematic construction of systems that can be defined using our framework. Finally, the integration of existing models for the new systems – following a top-down approach- is another issue that we think our framework can help address, and on which we are actively working.

Acknowledgements This work is partially supported by the Spanish Project TIC2002-04309-C02-02.

References

[1] S. Forth, B. McPhee, NQ. Chang. *An introduction to SCORM*. Recombo: Canada (www.recombo.com), 2002.

[2] IMS Global Consortium. *IMS Guidelines for developing accessible learning applications*. IMS Global Learning Consortium, Inc. 2001

[3] The Aviation Industry CBT (Computer-Based Training) Committee. http://www.aicc.org

[4] E.W. Dijkstra. *A Displicine of Programming*. Prentice Hall, Englewood Cliffs, NJ. 1976.

[5] J. Miller, J. Mukerji. *MDA Guide*. Object Management Group, January 2003. OMG document ab/2003-06-01.

[6] OMG. *Model Driven Architecture. A Technical Perspective*. Object Management Group, January 2001. OMG document ab/2001-01-01.

[7] IEEE. *Recommened Practice for Architectural Description of Software-Intensive Systems*. IEEE Standard 1471, 2000.

[8] Philippe Kruchten. *Architectural blueprints. The "4+1" view model of software architecture*. IEEE Software, 12(6):42{50, November 1995.

[9] John A. Zachman. *The Zachman Framework: A Primer for Enterprise Engineering and Manufacturing*. Zachman International, 1997. http://www.zifa.com.

[10] ISO/IEC. *RM-ODP. Reference Model for Open Distributed Processing*. Switzerland, 1997. International Standard ISO/IEC 10746-1 to 10746-4, ITU-T Rec. X.901 to X.904.

[11] J. Bézivin, E. Breton, P. Valduriez, G. Dupé. *The ATL transformation-based model management framework*. Research Report 03.08, IRIN, University of Nantes, 2003.

[12] K. Czarnecki, S. Helsen. *Classification of Model Transformation Approaches*. In Online Proceedings of the 2nd OOPSLA 2003 Workshop on Generative Techniques in the Context of MDA, Anaheim, 2003.