# Software Construction of an Authoring Tool for Adaptive E-learning Platforms

Dessislava Vassileva, Boyan Bontchev, Boryana Chavkova and Vladimir Mitev
*Department of Software Engineering,*
*Faculty of Mathematics and Informatics,*
*Sofia University "St. Kl. Ohridski",*
*5, J. Bourchier Blv., 1164 Sofia, BULGARIA*
*{ddessy, bbontchev, boryana.chavkova, vladimir.mitev}@gmail.com*

## Abstract

*In last decade, more and more platforms for e-learning content delivery provide adaptability towards learners goals, styles and performance. Usually, such platforms rely on own authoring tool or use external one in order to create learning materials. Usually, these tools follow modern e-learning standards but are rather complicated to be used and miss interoperability features. In this paper, we present software construction of an authoring tool, which is a part of a platform for building edutainment (education plus entertainment) services – ADOPTA (ADaptive technOlogy-enhanced Platform for eduTAinment). This authoring tool is designed by using Java EE 5 platform and provides inheritance mechanisms for learning object metadata descriptions, metadata for semantic ontology graphs, and good integration with instructor tool for creation of adaptive courseware.*

## 1. Introduction

Modern learning management platforms are inconceivable without suitable authoring tools for creation and maintenance of e-learning courseware. Hypermedia systems with adaptation towards user character address the same need but pose specific requirements for content organization and metadata description. This is straight following from the chief goal of personalized and adaptive e-learning stated in [1] as assuring of "e-learning content, activities and collaboration, adapted to the specific needs and influenced by specific preferences and context of the student". In order to achieve that goal, Adaptive Hypermedia Systems (AHS) possess abilities for provisioning of various forms of adaptation, such as adaptive navigation, structural adaptation, adaptive presentation and historical adaptation [2]. Dynamic adaptation is used in different instructional scenarios with content package adaptation facilitated by wide usage of Web services [3], or is based on the idea that different forms of learner model can be used to adapt content and links of hypermedia pages to given user [4].

The adaptability to individual is based on clean separation of the learner model from the content model and from the adaptation model, without narrative or pedagogical model to be embedded in the authored content or the adaptation engine [5]. The paper describes software construction of an authoring tool of e-learning courseware specially designed for ADOPTA (ADaptive technOlogy-enhanced Platform for eduTAinment) for building edutainment (education plus entertainment) content for both universities and industry. ADOPTA has been under development at Sofia University, Bulgaria, since 2007 and already provides prototypes of authoring and instructor tools [6] for e-learning courseware design, with intention to be extended for edutainment support. The adaptation engine in still under development - it executes rules controlling the adaptation process toward the learner model.

With ADOPTA, the authoring process is strongly separated from the instructor's learning design and is based on semantic ontology graphs - exported in Ontology Web Language (OWL) [7] and inheritance mechanisms for metadata descriptions of both the learning objects and ontologies. For describing metadata for learning objects (LOs) we use Learning Object Metadata (LOM) [8], while for semantic ontologies we rely on the new coming Ontology Metadata Vocabulary (OMV) [9].

## 2. Conceptual model of system adaptability

We have proposed a new AHS model with main goal to assure strong independence between leaner profile, author content and pedagogical strategy [10]. Table 1 presents its structure together with explanation of the most important characteristics. This is a new hierarchical organizational model which refines the established and widely used model - the AHAM reference model [2].

**Table 1. Tabular presentation of the structure of the conceptual model**

| | |
|---|---|
| **Learner Model** - contains information for the learner profile. Depending on its meaning, it is stored in *Goals and Preferences, Learning Style* or *Knowledge and Performance* sub-models. | **Goals and Preferences** |
| | **Learning Style** |
| | **Knowledge and Performance** |
| **Adaptation Model** - includes description of each course storyboard graph (in *Narrative Storyboard* sub-model), metadata (such as link annotations, exam thresholds, etc.) of each narrative storyboard graph (in *Narrative Metadata* sub-model) and selection logic for passing over particular graph (in *Storyboard Rules* sub-model). | **Narrative Metadata** |
| | **Narrative Storyboard** |
| | **Storyboard Rules** |
| **Domain Model** - is responsible for structuring of learning content. The content is granulized in LOs, which for theirs part are connected among themselves in relevant knowledge domain ontology. LOs and ontology are described by their metadata (in *Content Metadata* sub-model) respectively according IEEE LOM specification and Ontology Metadata Vocabulary OMV standard | **Ontology graph** |
| | **Learning objects** |
| | **Content Metadata** |
| The **Adaptation Engine** communicates with each of the three sub-models at first level in order to generate and delivery to particular learner the most appropriate learning content for her/him | |

Our model is divided into three sub-models, strongly independent one from another. This independence allows each one of the sub-models to be easily changed, without this to affect the others. This hierarchical model consists of two levels. At first level, the model is based on a precise separation between Learner, Domain and Adaptation sub-model, while at second level each of these sub-models is divided into three others sub models. Some of the sub-models may be defined by XML schemas, such as learner characteristics, content – by means of Sharable Content Object Reference Model (SCORM), ontology (OWL), metadata (LOM and OMV), and rules – in Semantic Web Rule Language (SWRL) [11], for a better cross-session interoperability and consistency.

As shown in table 1, in the Learner model we separate goals and preferences from shown knowledge and performance, which misses in other similar models and allows to adapt content according learner's knowledge and performance and to personalize it according learner's goals and preferences. Other difference between our model and similar ones is that we add a new sub-model – the learning style. In this sub-model, for each learner are defined her/his learning style, such as activist, theorist, reflector, pragmatist. This learning style can be polymorphic, which means that it is presented by order quadruple, since usually a particular learner is not fixed to a concrete style but rather to several ones at different level.

The domain model is composed of content itself (granulized in LOs according to the SCORM standard), LO's and ontologies' metadata and semantic ontologies organizing the content (LOs). There are supported various types of LOs – not only narrative content but also tasks, essay, assessment question, game, etc. Thus, the content LOs are developed by the author and next are placed by the course instructor on course pages.

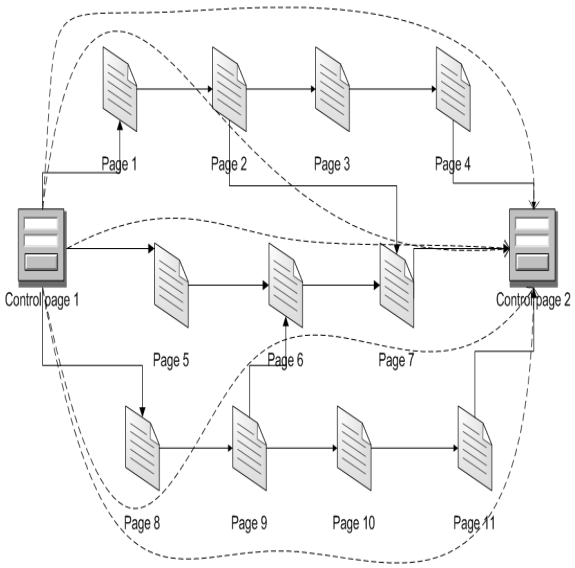


**Figure 1. A sample narrative storyboard graph**

The adaptation model (AM) captures the semantics of the pedagogical strategy employed by a course. It includes support of course storyboard graphs. Fig. 1 presents a sample for narrative storyboard course graph

and it consists of narrative pages (with learning content compound of LOs) such as *Page 1*, *Page 2*, control points (CP) such as *ConrolPage 1* and *ControlPage 2* and so called work paths (WP) between them (CPs). The instructor may define a weight of a WP for each learning style. Therefore a particular working path (WP) may be suitable for one or several learning styles. The control points are used for assessment of current knowledge and performance for a learner, by test generation. This test is composes of questions corresponding to the LOs in the pages, which the learner is visited. The obtained assessment result is used for update of WP weights.

The main benefit of the proposed model is in assuring flexible adaptation of content delivery and possibilities for effectiveness and easy expandability in terms of adaptive content management and support. It can be supported by different system architectures not limiting application of various adaptation techniques, such as adaptive content presentation, navigation support and content selection.

## 3. Principal software platform architecture

### 3.1. General process workflow

The ADOPTA platform for adaptive e-learning includes an authoring tool, an instructor tool, an adaptive engine and a set of administration tools, all communicating through a common repository as shown in fig. 2. The content author is responsible for design of learning materials (objects) by organizing them within ontology with has-a and is-a relationships and, also, for metadata about LOs (by IEEE LOM) and about ontology itself (by OMV). The instructor uses the instructor tool to design a course as a narrative storyboard, by defining course pages and links between them. For a content page, he/she has to drag-and-drop at the page one or more learning objects from a proper ontology defined by an author. The supervisor is responsible for controlling the adaptation engine, e.g. for doing start and stop of adaptation behavior, tracking learner paths, etc. The administrator controls all the users by means of administrative tools.

Finally, the learner follows a course by receiving adaptive content and solving tests at control points. The learner is supposed to start at the first control point by filling an initial test about determining his/her learning style. Next, he/she follows the work path proposed by the adaptation engine but may opt to links to pages not belonging to the path and, thus, to divert to another work path. In such a case, they are always able to return back to the last visited page of the proposed path or, otherwise, to follow the new path until reaching a control point. There, the learner has to solve a test compiled by automatically selected questions about the LOs he/she has passed through.

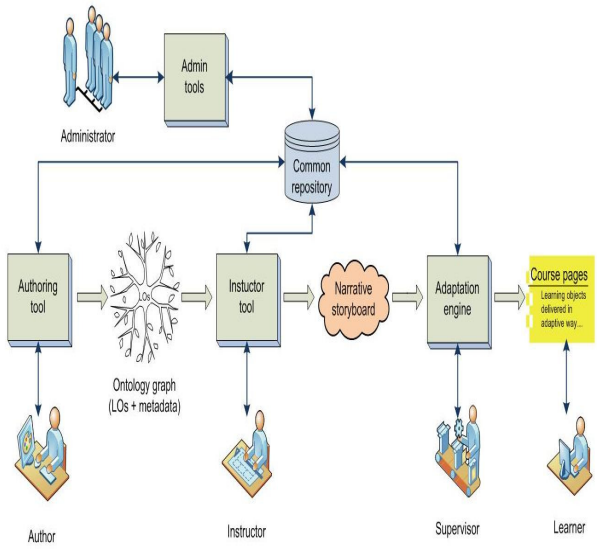


**Figure 2. View of the general workflow**

### 3.2. Overall platform architecture

The ADOPTA software architecture is composed by three main layers – web clients, business layer and persistence layer, as shown in fig. 3. The persistence layer is presented by two sub-layers:

- Adopta Persistence Entities – ungrouped and common for all the platform applications
- Persistence Session Beans – grouped into specific and common, and used for read/store/edit of entities. Within this sub-layer, we have reused functionality for reading the same objects, while the business logic is specific for every module. Even in the case of login, UserEntity is always read but there are checked different roles.

The other layers are as follows:

- Business Session Beans – EJBs [12], which are specific for each of the modules and contain its business logic
- Communication layer (Web services) – provide specific services for each of the modules.
- Web clients – represent web-based service clients. The client layers are build with the constantly growing popularity Flex technology [13]. Among its other benefits, this technology allows to generate easily web service client

classes and method stubs. This is exactly the way the client consumes the published web services. The nature of the Flex applications to be run on the client side (browser/desktop) dispenses the application server with the load of rendering and manipulation the data.


**Figure 3. General platform architecture**

## 4. Software construction issues of the authoring tool

### 4.1. Workflow of authoring e-learning courseware

Authoring process includes definition of LOs, their semantic organization as ontology graph (used for a easy viewing and searching), and metadata about LOs and ontology. The ontology graph allows multiple inheritance and references from one LO to another.

The author may design the ontology using a top-down, a bottom-up or a mixed approach. While node relations of type is-a are defined directly within the ontology graph, reference relations (usually known as has-a relations) are defined while designing the learning object. This constraint is especially dedicated to force the author to allocate a hyperlink to the referenced object within the content of the referring object. LOs may be of various types such as conceptual issue, project task, essay, etc. For anyone of these LOs, the author may design one or several assessment LOs. An assessment object is a QTI question [14] with several answers of type one-of-many or many-of-many. For each answer, the instructor defines a result value. Questions may be only referred by other LOs of type not being question, and cannot refer to other LOs. Moreover, question LOs are not shown at course pages but are used by the adaptation engine to build an assessment tests at next CP.

Besides LOs and their structure, the author is supposed to define metadata for LOs (by IEEE LOM) and for the ontology. He/she may use a mechanism of metadata inheritance from the root LO for the ontology to its successors. If metadata records of each LO are identical, then the author has to specify them only for the root LO. Otherwise, the author has to describe only the differences between metadata records of ancestor LO and its successor (if there are any).

### 4.2. User interface of the authoring tool

One of the main goals of our authoring tool is to provide comfortable, user-friendly and flexible interface for ontology, LOs and its metadata management. For this purpose we use Adobe Flex to design and implement our authoring system interface. It is an open source framework, which assures the creation and maintenance of expressive web applications.

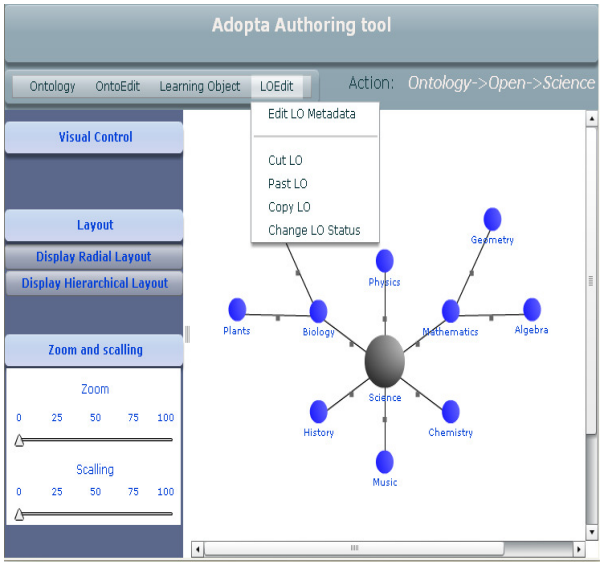


**Figure 4. A sample narrative storyboard graph**

The authoring tool is based on reusing the already existing authoring tool of ARCADE (Architecture for Reusable Courseware Authoring and Delivery) e-learning platform [15]. As far as it may run as a

standalone application, we have integrated its extended version into our system. In this version, the learning content is presented by LOs connected each other within a semantic ontology graph (fig. 4) through links, which can be of type *has_a* or *is_a*. LOs in accordance of theirs structure may be:

- primitive (containing plain text, table, image, audio, animation, video, external resources, or links). This type of LOs has linear structure
- composite (aggregating other LOs). They are with a hierarchical, tree structure.

As well, LOs may have various types in accordance of their purpose – narrative content, task, assessment question, etc.

The authoring tool's interface provides various functionalities for flexible visualization and presentation of a ontology graph as zooming and scaling, different layouts, multiple level view, etc. Adobe Flex enables creating complex data visualization interfaces for social networks, navigation systems, taxonomies, etc. Moreover, the authoring tool facilitates authors of learning content in the filling of metadata for learning objects through assuring of multiple inheritance – LOs lying down inherit LOM from upper objects and may redefine it. Thus, the author should define a full LOM description only for the top LO class within the ontology graph, while for the other LO (subclasses) this description will be inherited with possibility for overriding any field. As the most of contemporary authoring tools, our ones supports export of ontology, LOs, and metadata in appropriate formats.

### 4.3. Architectural view of the authoring tool

Conforming to the general system architecture, each module the authoring tool consists of three layers-persistence, business and web (or client) layer. Each of the layers resolves its own specific problems and relies only on the layer below.

Like the name shows, the persistence layer is responsible for storing and editing of objects. As all modern applications do, the communication with the database is made throughout the Java Persistence API. Sample Java Persistence entities are the LearningObject, Ontology, etc. - all compliant to the ORM standard.

The business layer is build by the latest EJB technology [12]. The business logic itself resides on stateless EJBs. The following EJB have been created LoginBean, LOBean and OntologyBean. The LoginBean is responsible for both the authentication and authorization in the application. The LOBean and the OntologyBean contain the business logic related to a specific set of objects. Each bean exposes an interface so the communication with each bean happens via this interface throughout JNDI injection.

The next layer is build again on the basis of the EJB 3.0 architecture, although part of the business layer can be relatively separated in a newly called communication layer. This layer consists of web services that act as a communication point between the services client and the beans where the business logic resides. The authoring tool publishes several services that can be divided in four groups - login, learning object related, ontology related and learning object links related. All services are published as part of a single WSDL file.

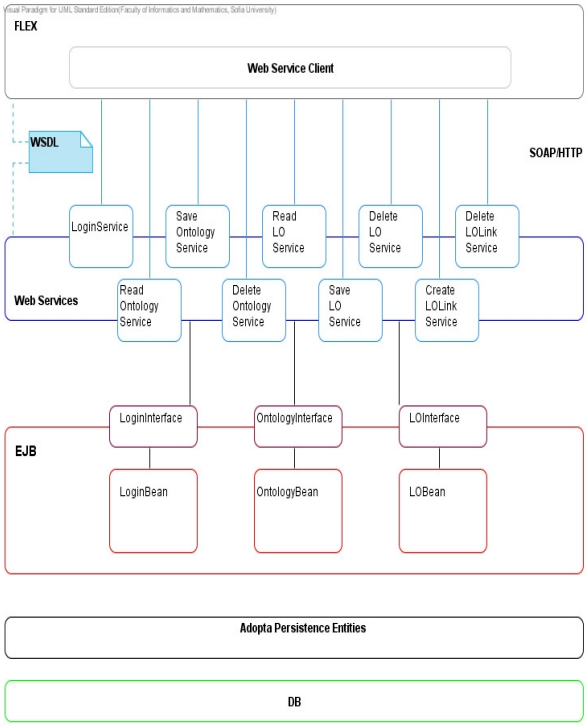


**Figure 5. Architecture of the authoring tool**

The last layer of the application is the web layer. Having in mind the benefits of SOA the web layer may be more precisely called with the more general name-client layer.

## 5. Related works

Most of the older authoring tools such as InterBook and HyperBook [16] are aimed at creating an entire course with pre-defined structure and pedagogical strategy. These applications do not use learning objects

and structuring of learning materials in ontologies. Many contemporary authoring tools have focused their efforts in standardization of content organization and its reuse. Therefore they organized its training materials in learning object. Such systems are WebCT, Learning Object Creator, and ATutor [17]. Unlike these applications, our authoring tool separates the content from the pedagogical strategy, which is not defined by it but by the instructional tool. Moreover, we support inheritance mechanism for defining LOM and very convenient interface created by Flex technology.

## 6. Conclusions

Adaptive e-learning platforms continue needing of appropriate authoring tools facilitating instructional design of adaptive courseware. The paper presented important issues of software construction of the authoring tool of ADOPTA platform for adaptive edutainment. The software architecture of ADOPTA separates the process of course material authoring from instructional design, in order to reuse learning courseware and to facilitate effective construction of narrative storyboards. It makes sense, as the roles of the author and the instructor are different, although they may be played by the same teacher.

Moreover, the system architecture discussed here allows three separated and independent each other applications – the authoring and the instructor tools and the adaptation engine for courseware delivery and assessment – to be deployed and to run on different machines. Each one of the applications contains persistence, business and web layer but only the persistence layer is the same. Thus, even in cases of crash of one of the three applications, others are able to run independently. The EJB remote interface is used in order for deployment of a single persistence layer common for these applications, which allows centralized management and easy version control.

## 7. Acknowledgements

This work was partially supported by the European Social Fund and the Bulgarian Ministry of Education and Science under the Operation programme “Human Resources Development”, Grant BG051PO001/07/3.3-02/7.

## 8. References

[1] Dagger, D., Wade, V. & Conlan, O., Personalization for All: Making Adaptive Course Composition Easy. Special issue of the Educational Technology and Society journal, IEEE IFETS, 2005.

[2] De Bra P. at al. AHAM: A Dexter-based Reference Model for adaptive Hypermedia. Proc. of the ACM Conference on Hypertext and Hypermedia, 1999, pp. 147-156.

[3] Leune, K., W.J. van den Heuvel, Papazoglou M.P. Exploring a Multi-Faceted Framework for SOC: How to develop secure web-service interactions? Proc. of the 14th Int. Workshop on RIDE, USA, 2004, pp. 485-501.

[4] Díaz, P., Sicilia, M.A. and Aedo, I. Evaluation of Hypermedia Educational Systems: Criteria and Imperfect Measures. Proc. of the Int. Conf. on Computers in Education, USA, 2002, pp. 621-626.

[5] Vassileva D., Bontchev B. Self adaptive hypermedia navigation based on learner model characters, Proc. of IADAT-e2006, Barcelona, Spain, 2006, pp.46-52.

[6] Vassileva, D., Bontchev, B. and Grigorov, S., Mastering Adaptive Hypermedia Courseware, Proc. of 6th Int. Conf. on Emerging eLearning Technologies and Applications ICETA'2008.

[7] Moreira D., M. Musen. OBO to OWL: a protege OWL tab to read/save OBO ontologies, In Bioinformatics, Vol. 23, No. 14, 2007, pp. 1868-1870.

[8] Krull, G. An investigation of the development and adoption of educational metadata standards for the widespread use of learning objects, Master Thesis, Rhodes University, November 2004.

[9] Hartmann, J. et al. Ontology Metadata Vocabulary and Applications. Proc. of Int. Conf. on Ontologies, Databases and Applications of Semantics, Workshop on Web Semantics (SWWS), Springer, October 2005, pp.906-915.

[10] Vassileva D., Bontchev B.: Adaptation engine construction based on formal rules, Proc. of CSEDU 2009, ISBN 978-989-8111-82-1, Vol.1, Lisbon, Portual, March 23-26, 2009, pp.327-332.

[11] Mei, J. and Boley, H. Interpreting SWRL Rules in RDF Graphs. In Electr. Notes Theor. Comput. Sci., Vol. 151(2), 2006, pp. 53-69.

[12] J. Wang at al. Constructing an EJB application in a WFMS, Proc. 26th Computer Software and Applications Conference, 2002, ISSN: 0730-3157 , pp. 284- 286.

[13] Coenraets C. Extend AJAX with Adobe Flex, Published by SYS-CON Media, Real-World AJAX Seminar, New York City, USA, June 5-6, 2007, http://www.soaworld2007.com/node/226335.

[14] Radenkovic, S. et al. A QTI Metamodel, Proc. of the Int. Multiconference on Computer Science and Information Technology, ISSN 1896-7094, Volume 2, Wisła, Poland, 2007, pp. 1123 – 1132.

[15] Bontchev B., Vassileva D. “Internet Authoring Tool for E-learning Courseware”, Proc. of the 7th WSEAS Int. Conf. on Computers. Corfu, Greece, 2003, pp. 305-311.

[16] P. Brusilovsky, J. Eklund, InterBook: an Adaptive Tutoring System,. UniServe Science News, Volume 12, March, 1999.

[17] Esmahi, L. at al. An Essay in E-learning Tools Categorization, Proc. of "Computers and Advanced Technology in Education", Cancun, Mexico, 2002, pp.355-218.