An Architecture for Defining Re-usable Adaptive Educational Content

Charalampos Karagiannidis and Demetrios Sampson
Informatics and Telematics Institute (I.T.I.)
Centre for Research and Technology – Hellas (CE.R.T.H.)
1, Kyvernidou Street, Thessaloniki, GR-54639 Greece
Tel: +30-31-868324, 868785, 868580, internal 105
Fax: +30-31-868324, 868785, 868580, internal 213
E-mail: karagian@iti.gr, sampson@iti.gr
www.iti.gr

Fabrizio Cardinali
GIUNTI Interactive Labs S.r.l.
Via al Ponte Calvi 3/15
16124 Genova, Italy
Tel.:+39-010-2465178
Fax: +39-010-2465179
E-mail: f.cardinali@giuntilabs.it
www.giuntilabs.it

Abstract

This paper addresses re-usability in personalized learning environments. The paper presents the work of the European Project KOD "Knowledge on Demand", towards the development of an architecture for defining re-usable adaptive educational content which can be easily interchanged and re-used across different personalized learning applications and services.

1. Introduction and Background – The Need and the current Practice

The KOD project aims to address the needs of the different categories of users (market players) involved in the *e*-Learning arena, including: *e-Learning assets publishers*, aiming to version their learning assets for different online learning solutions in a re-usable and interoperable way; *e-Learning platform providers*, mainly interested in providing architectural solutions for *e*-Learning at different levels, e.g. learning management systems, assessment systems, performance support systems, etc; and *e-Learning service providers*, mainly using existing *e*-Learning platforms in conjunction with their own, or third-party learning content, to support service provision in the *e*-Learning arena.

In general, these users need to be able to publish to (or access from) public knowledge repositories learning material (either "single" learning assets, or "learning packages"), so that it can be easily interchanged across different applications and services. This, in turn, requires that learning material is described and published in a *common format* [1].

This need has resulted in a number of international standardization activities aiming to define common learning technologies specifications and standards that can ensure interoperability in the *e*-Learning arena. The main initiatives in the area are the IEEE LTSC (Learning Technologies Standards Committee, ltsc.ieee.org), the European CEN/ISSS Learning Technologies Workshop (www.cenorm.be/isss/Workshop/lt/), the IMS (Instructional Management Systems) Global Learning Consortium Inc (www.imsproject.org) and the US ADLnet (Advanced Distributed Learning Network, www. adlnet.org) [2].

These efforts have already resulted in a number of specifications for *e*-Learning applications and services. In fact, current specifications already encompass (or will, in the next future) almost all aspects of a standard *e*-Learning architecture (e.g. the IEEE LTSA, Learning Technologies Standard Architecture) from the description of learning objects meta-data based on shareable XML-based data structures (e.g. through the IEEE LOM, Learning Objects Metadata Schemas) to the assessment of user performances (e.g. through the IMS QTI, Question and Testing Interoperability Schemas). That is, existing specifications enable the common description of learning units, questions and tests, learner profiles, etc, so that they can be easily interchanged between different applications.

Moreover, existing specifications enable users (publishers, platform providers, service providers, etc) to "package" and publish content structures which are built on content building blocks in a standard way, and in particular through the IMS Content Packaging (CP) Specification. The CP XML schema reports a meta-data header describing the packaged resource itself, an *organization* field describing content structure and content packages included in the proposed learning path, and the list of meta-data of all referenced resources together with links to the encapsulated resources themselves. The schema allows for iterative descriptions, i.e. one can build onion like structures encapsulating resources within resources; every new resource has a new general meta-data describing it and an organization field describing its navigational structure. Using the CP specification, a publisher may describe a course as a "packaged pathway" through existing modular resources own by either the

0-7695-1013-2/01 $10.00 © 2001 IEEE

publisher or third parties. These can be described and chained in the standard description of the flat file based on a XML notation implemented in the CP format.

However, existing standards do not adequately support the definition and interchange of reusable adaptive and flexible learning methods which are beyond the "rigid" approach of directive, curricular-based, linear learning, as enabled by the envisaged hierarchical structure description in the Content Packaging standard. In particular, the organization field in the XML schema enabling the IMS content packaging manifest, although open to any notational description of navigation, mainly considers "rigid" hierarchical tree-based content structures description; no standard declarative notation, toolkit and viewer is available for conditional branching navigation or for path redirection.

In this context, the KOD project has identified a procedure for defining adaptive educational content in a way that is can be easily interchanged and re-used across different *e*-Learning applications and services [3]. The steps of this procedure include the definition of:

1. the concept ontology of the learning material to be presented,
2. the learning resources, i.e. the learning units that are to be communicated to the learner,
3. the competencies which are related to each node of the concept ontology,
4. the questions and tests that define when a learner has acquired a specific competency,
5. the different user profiles of the user groups that are expected to be interacting with the system, and finally
6. the navigation rules which define how different learning objects are selected for different learners; these "rules" specify the "matching" between the learner profiles and the learning content, that is the "learning path" that is appropriate for each different learner profile.

This paper focuses on the architecture that has been developed in the KOD project for defining adaptive educational content in a common format, following the above procedure. The architecture is shown in Fig. 1, and described in the following section.

2. The Proposed Architecture

The architecture includes a *Learner Interface*, which facilitates access to the learners' functionality. The Learner Interface supports all the features that are available in existing state-of-the-art *e*-Learning, *e*-Publishing and *e*-Knowledge tools. In addition, it facilitates access to PL services, i.e. enabling the user to authenticate himself, and subsequently to define, review and modify his/her user profile, so that the PL environment is adapted to the user's requirements, preferences, interests, goals, etc.

In the following paragraphs, the "KOD Factory" is described, i.e. the components of the architecture which enable the definition of adaptive educational material. The description is given in terms of a scenario of use, describing the steps that need to be performed by an "editor", i.e. a person responsible for defining adaptive content so that is can be interchanged across different PL environments.

2.1 Ontology toolkit

First of all, editors need to define the content to be presented to the learners. To this end, the first step in the definition of a "knowledge route" (i.e. the output of the PL environment) is the definition of the ***ontology*** that describes the concepts to be communicated. It should be noted that this ontology contains a factual description of the learning concepts (e.g. computer science is composed of operating systems, programming languages and databases), which is *independent* from the learning process. This ontology can be provided by a content expert, who is being interviewed by an editor. In this context, the Editor Factory includes an ***Ontology Toolkit***, which assists editors in this process. The Ontology Toolkit employs a specific *knowledge representation* technique for storing the ontology, and provides as output XML files (***Ontology Profiles***), which are maintained in the ***XML database*** of the PL environment.

2.2 Meta-Data Toolkit

Following the above process, editors need to define the learning material / resources that are available for each "atom" of the ontology (i.e. the leafs in the ontology structure). In this context, the Editor Factory includes a ***Meta-Data Toolkit***, which assist editors in defining the meta-data of the learning resources that are available. The Meta-Data Toolkit stores the meta-data as XML files (***Meta-Data Profiles***), which are also included in the ***XML database*** of the PL environment.

2.3 Questions & Tests Toolkit

In addition, the ontology should be "enriched" with questions and tests, which can specify when the learner has mastered a concept in the ontology hierarchy, and can therefore proceed with the next concept. The process of defining these tests is assisted by a ***Question & Test Toolkit***, which enables the editor (with the help of the content expert) to define the questions and tests that are related for each concept in the ontology hierarchy. The

Question & Test Toolkit stores ***Question & Test Information*** this information into XML files, maintained into the ***XML database***.

2.4 Competencies Toolkit

Based on the ontology, editors then need to define the competencies that are related to each node of the ontology. In this context, the Editor factory includes a ***Competency Toolkit***, assisting editors interviewing content experts, and storing this information into XML files (***Competency Profiles***), maintained in the ***XML Database***.

2.5 User Profiles Toolkit

Then editors need to define which are different profiles of the users (learners) who are expected, foreseen, etc, to interact with the system. The definition of the user profiles (which is assisted by a content expert) is assisted by a ***User Profiles Toolkit***, with similar functionality with the above tools. The User Profiles Toolkit stores ***User Profiles*** in XML format, maintained within the ***XML database***.

2.6 Rule Toolkit

Finally, editors need to define how the learners will navigate the concepts of the ontology (i.e. viewing the learning resources, questions and tests, etc) that have been specified for each concept in the ontology, based on the user profile of the learner, as well as the competency level. Therefore, a ***Navigation Rules Toolkit*** is included in the Editor Factory, which assists editors to define the rules that determine the learning paths in the ontology that should be followed, and their matching to user profiles and competencies. It should be noted that these rules (***Navigation Rules***) are *dependent on the specific learning content* to be communicated, and are again stored in XML files within the ***XML database***.

The Rule Toolkit also enables the editor to define ***general rules***, which are applicable to every learning context, and stored in the SQL database of the PL environment.

All the above information form the basis for the definition of the different *knowledge routes* of the PL environment. Knowledge routes contain the ontology of the learning material, the learning objects and questions and tests that are related to each node in the ontology, the different user profiles and competency profiles, as well as the *navigation rules* which determine how the ontology is navigated for different learner profiles.

2.7 Agents Toolkit

The proposed architecture is based on software agents, which are the knowledge analyzing, monitoring, generating, adapting and delivering "pulse" of the PL environment. In this context, the Editor Factory includes also an ***Agents Toolkit***, which enables the editor to construct new agents, destruct existing agents, modify agent parameters, etc.

Agents are capable of presenting knowledge routes to learners, i.e. processing knowledge routes files. In particular, agents extract the information contained in the knowledge routes, and present learning material according to the learner profile. At each node of the ontology, agents are capable of identifying whether the user can understand the respective concept, or whether there is a need for presenting some "pre-requisite" concepts before. In case that the user cannot understand these concepts based on the learning material that is available (i.e. the physical learning resources indicated in the knowledge routes file), the agent automatically searches for additional information, both from specific repositories (defined by the editor), and from the Internet.

Also, agents provide assistance for the "verification" of the information that is encapsulated in the knowledge routes. For example, agents can notify the editor that the users that have accessed a specific knowledge route can be classified into more (or less) user profiles (employing data mining techniques).

3. Discussion and Conclusions

The different types of information that are defined through the procedure described in the previous section are necessary for the provision of adaptive content. In order for this information to be transferred across different applications, it needs to be maintained in *a common format*, i.e. following the existing and emerging *e*-Learning standards and specifications. For example, the description of the learning objects needs to be based on the Learning Objects Metadata (LOM) standard of the IEEE LTSC P1484.12 Working Group; the competencies need to be represented following the specification of the IEEE LTSC P1484.20 Competency Definitions Working Group; etc.

On the other hand, *all* the above types of information need to be represented *together* following a single common specification. To this end, the KOD project is currently working on the proposition of an extension of the IMS Content Packaging Specification [4], to include the above information. In particular, as it has been described in the introductory section, the current version of the specification enables (through the "organization"

element) the description of "rigid" hierarchical tree-based content structures description (e.g. simple table of content).

The KOD project aims to propose an extension of this specification, where all the information presented in the previous section is also included in the content packaging description. This will facilitate the description of adaptive educational content in a common format, thus enabling users and publishers to share not only content and content routes, but also navigation algorithms (i.e. conditional branching based on user performances). As a result, adaptive educational content can be interchanged and re-used across different personalized learning applications and services, and thus re-usability in personalized learning can be promoted.

References

[1] Sampson, D., & Karagiannidis, C. (eds.) (2000). *Report on User Needs Analysis and the KOD Model Definition.* KOD Project Deliverable D1.1 (available from the authors).

[2] Bacsich, P., Heath, A., Lefrere, P., Miller, P., & Riley K. (1999). *The Standards Fora for Online Education.* D-Lib Magazine, 5(12).

[3] Karagiannidis, C., Sampson, D. & Cardinali, F. (2001). *Integrating Adaptive Educational Content into Different Courses and Curricula.* Educational Society and Technology, Special Issue on Learning, Instruction, Curriculum and the Internet, 4(3), In Press.

[4] IMS Project (2000). *Content Packaging Information Model.* Version 1.0 – Final Specification.

Acknowledgements

The KOD "Knowledge on Demand" Project (www. kodweb.org, kod.iti.gr) is partially funded by the European Commission Information Society Technologies (IST) Programme.

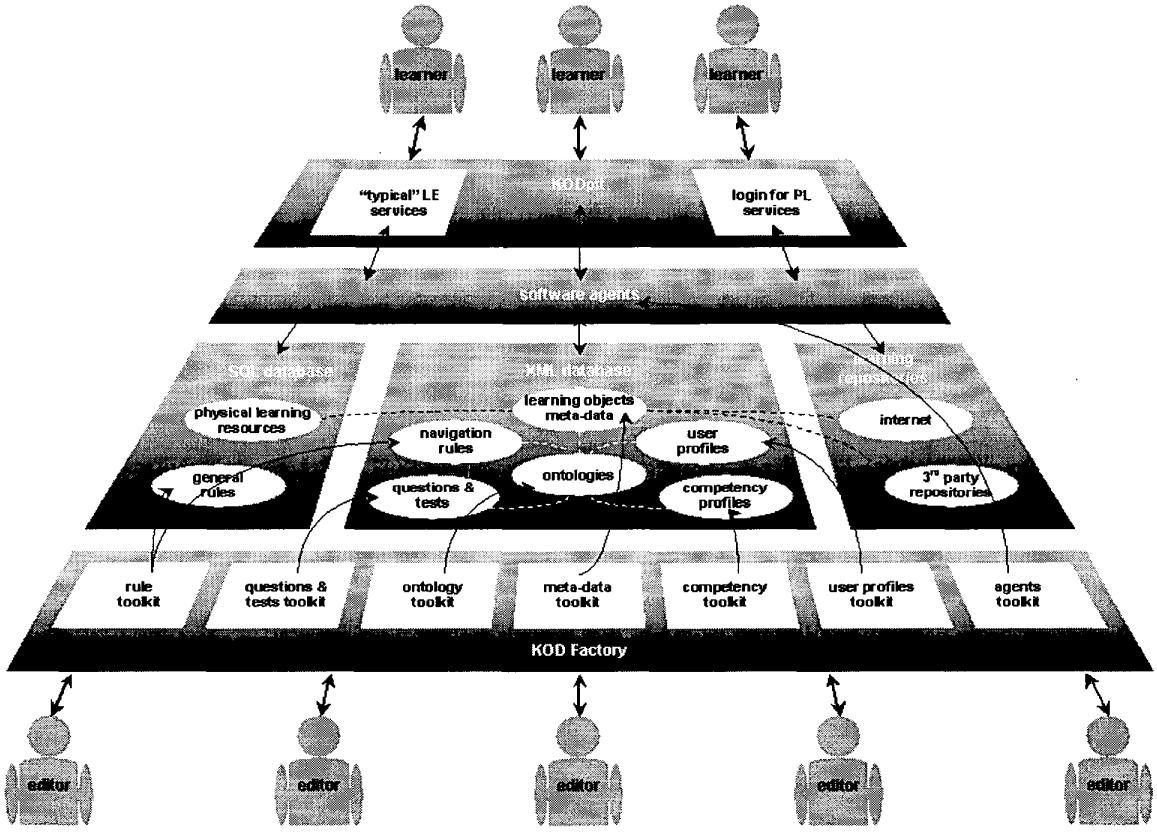

Fig. 1. The KOD Architecture for the Definition of Re-usable Adaptive Educational Content