Learning Design and Run-Time Resource Binding in a Distributed e-Learning Environment

Nicola Capuano, Matteo Gaeta, Roberto Iannone, Francesco Orciuoli

Learning Design and run-time resource binding in a distributed e-learning environment

N. Capuano[1], M. Gaeta[1], R. Iannone[1], F. Orciuoli[2]
[1]*Centro di Ricerca in Matematica Pura ed Applicata, {capuano, gaeta, iannone}@crmpa.unisa.it*
[2]*MoMA Srl, {orciuoli}@momanet.it*

Abstract

Nowadays, the need for e-learning systems supporting a rich set of pedagogical requirements has been identified as an important issue in the field of distance learning. Several initiatives take place in order to meet this need. Maybe, the most important of these initiatives is IMS Learning Design [6] that provides a framework to depict pedagogies. Furthermore, we are aware that the provision of different learning paths tailored on learner's characteristics and preferences guarantees the learner to reach a cognitive excellence. In this paper, we first illustrate an extension for IMS Learning Design in order to overcome its limitations forcing instructional designers only describing domain-dependent pedagogies. Then, we propose a mechanism to build units of learning by merging domain-independent pedagogies with educational objectives selected by teachers inside a specific didactic domain modelled by a graph structure called ontology. Finally, we illustrate a delivery networked infrastructure able to execute units of learning in a personalized way based on learner preferences.

1. Introduction

Building individualized learning activities to support personalized instruction in an adaptive environment is a big challenge for the future of e-learning. The web offers the perfect technology and environment for individualized learning, since learners can be uniquely identified, content can be specifically personalized, and learners cognitive states progress can be monitored, supported and assessed [7].
An important result in e-learning field is achieved by IMS Learning Design (IMS-LD) that proposes a language for describing learning processes, called learning design scenarios, where users can play learning activities embodying several roles.
Although IMS-LD is very useful for instructional designer to model, in a machine understandable representation, a wide range of pedagogies, nevertheless it does not support, in a native way, personalized formative paths since real contents and services are bound at design-time to the specific learning scenario that models the learning process. IMS-LD early binding thwarts the use of learner profiles that could be used as directives in order to build, at run-time, personalized learning experiences.
Furthermore, nowadays several educational e-content repositories and networked infrastructures are available and accessible in order to delivery learning scenarios where services, users and contents are decentralized and selected at run-time in order to provide effective and reliable learning experiences all the time to final users.
The main goal of this paper is depicting some ideas of extension for IMS-LD in order to model domain independent pedagogies as learning design scenarios. Domain-independent learning design scenarios will be used by teacher to construct domain specific units of learning by merging pedagogies with educational objectives extracted by ontologies used to model specific didactic domains. The units of learning will be bound with real contents and services at learners fruition-time in order to best fit their preferences and networked resources availability.
The paper is structured as follows. In section 2 we present IMS-LD. The section 3 underlines some drawbacks of IMS-LD regarding its lack of support for domain-independence and personalized learning. In section 4 we propose our solution for IMS-LD drawbacks illustrating a scenario in which instructional designers and teachers/authors build units of learning (UoL) embedded in different domains based on a domain-independent learning design scenario. In the section 5 we propose a distributed infrastructure able to run learning experiences. The distributed infrastructure is presented by illustrating a scenario in which

the units of learning constructed before are personalized and delivered to learners. Finally, the section 6 proposes some ideas for further works and conclusions for the paper.

2. IMS Learning Design (LD)

IMS Learning Design is a specification used to describe learning design scenarios. It allows these scenarios to be presented to learners on-line, and enables them to be shared among systems. It can describe a wide variety of pedagogical models, or approaches to learning, including group work and collaborative learning. It does not define individual pedagogical models; instead it provides a high level language, or meta-model, that can describe many different models. The language describes how people perform activities using resources (including contents and services), and how these three things are coordinated into a learning flow.

IMS Learning Design describes how a learning design scenario unfolds through the analogy of a theatrical play. Just as a play can be staged with different actors, in a different theatre with alternative props, so learning design scenarios can be executed again with different learners and tutors, on different systems, with alternative learning resources or tools:

- The play is presented in a series of acts, in which roles are played by those taking part, for example learner, tutor, mentor, and so on.
- People playing the roles undertake a series of activities within an act. For a learner these might include discussing with classmates the relative merit of a piece of source material. A tutor's activity may be to comment on their conclusions.
- Each role is presented with its own learning objects (LO) and services (e.g. communication tools) within an activity.
- An act is completed after all the activities of a specified role, or roles, are finished. Alternatively, a time limit may be set, after which an act completes.
- As soon as an act is completed, the next one starts. The play finishes when all the acts are completed, the learning design scenario finishes when all the plays are completed.

Services offer generic functions such as email, conferencing, searching and announcements. The locations of services are not specified during design, but are made available at run time, after people have been assigned to their roles. Both services and learning objects are referenced by activities. This means that these elements are located separately so that they can be reused and updated easily.

The model revolves around describing units of learning atomic or elemental units providing learning events for learners, satisfying one or more interrelated learning objective.

In a unit of learning, people act in different roles in the teaching-learning process. In these roles, they work towards certain outcomes by performing learning and/or support activities within an environment, consisting of learning objects and services to be used during the performance of the activities.

The approach separates learning objects and services from the educational method used in the unit of learning.

IMS Learning Design specifies a language for describing learning activities, and gives a binding for this language to XML (Extensible Markup Language).

An IMS Learning Design player (the execution engine) is a software tool that interprets, at run-time, the XML notation of a learning design scenario as participants work through it. The player may be a stand-alone tool, or it may be part of a virtual learning environment (VLE).

In conclusion IMS Learning Design orchestrates e-learning scenarios by:

- Describing and implementing learning activities based on different pedagogies, including group work and collaborative learning.
- Coordinating multiple learners and multiple roles within a multi-learner model, or, alternatively, supporting single learner activities.
- Coordinating the use of learning contents with collaborative services.
- Supporting multiple delivery models, including mixed-mode learning.

3. IMS-LD and domain-independent pedagogies

The IMS Learning Design specifications are structured in three level. Level A includes *activities, roles* and *environments*. Activities (learning activities or support activities) can be grouped into *activities structures* and executed into specific *environments*. An *environment* is formed by *learning objects* and *services* provided to users during *activity* execution. Users are classified into *roles* (learners, teachers, tutors, etc…). Nowadays, *learning objects* are educational contents by which learners acquire knowledge and *services* are functionalities invoked during learning process in order to communicate with tutors or other learners. Level B adds properties (storing information about a single person or a group) and conditions (setting constraints upon the flow of activities) to the first level. Level C adds notifications (mechanism to handle messages passing between users) to the framework .

Focusing on Level A and B we observe that though IMS-LD presents a mechanism to personalize the learning experience at run-time using properties and conditions at Level B, however the instructional designer has to provide a fixed set of learning objects and services in which selecting contents and tools for learning experience final users. So we can consider that learning objects and services are statically bound to the learning design scenario.

Design-time resources binding (LOs and services) is affected by the following problems:

- **Learning design scenarios implement domain-dependent pedagogies**: the instructional designer is forced to establish "a priori" the didactic domain, building the environments tied to the scenario's activities.
- **Learning processes cannot be really adaptive (based on learner profiles):** systems for learning material delivery cannot take into account learner preferences and cannot select, at run-time, learning objects that best fit learners characteristics.
- **E-learning scenarios don't exploit some advantages of distributed infrastructures**: the early binding of resources at design-time thwarts the opportunity for dynamic selection of learning objects and services that best fits learning process needs. Of course, these needs can be better supplied if the source of resources is a whole virtual organization (on a networked infrastructure) than the environments linked to the learning design scenario.

Our proposed solution consists in modifying the learning design scenario authoring process specified by IMS-LD in order to provide only requirements for a desired LO rather than a reference to a real LO. Requirements can be expressed using a subset of *IMS Learning Resource Metadata* (IMS-MD) [8] specification. These requirements will be transformed in references to real LOs at learning design scenario execution-time. A learning design scenario with pending references to LOs can be considered a *template learning design scenario* [12]. In the next sections we propose a complete e-learning sample scenario in which an instructional designer describes a pedagogy using a template learning design scenario, an expert models a didactic domain, a teacher prepares a unit of learning based on the constructed template learning design scenario and the provided ontology. At last, a learner runs the unit of learning in a personalized way.

4. A complete e-learning scenario: building a Unit of Learning

A unit of learning (UoL) is a sequence of learning objects (enriched by services provided to final users) assembled by a teacher to explain some concepts.
In this section we show the overall process needed to build a unit of learning (based on a learning design scenario) that can be personalized at run-time with respect to learner preferences.

Pedagogy construction (first step). First of all, the instructional designer builds a pedagogy using IMS-LD language though he doesn't specify real learning objects for each activity (through environments) in the play, on the contrary he/she assigns, for each LO in the learning design scenario, some requirements that learning objects should fulfil in place of setting references to real learning objects. The needed requirements are values for the following properties: *learning resource type* (LRT), *interactivity level* (IL) and *interactivity type* (IT) [8].

IMS-LD, through the mechanism of environments, allows to specify more than one resource (LOs and services) within each activity but, without lack of generality, we will use only one resource for a single

activity and one activity for each act. So, in our example, we will use the word “activity” in the place of the word “act”. The constructed template learning design scenario (or LD document) is illustrated in figure 1.

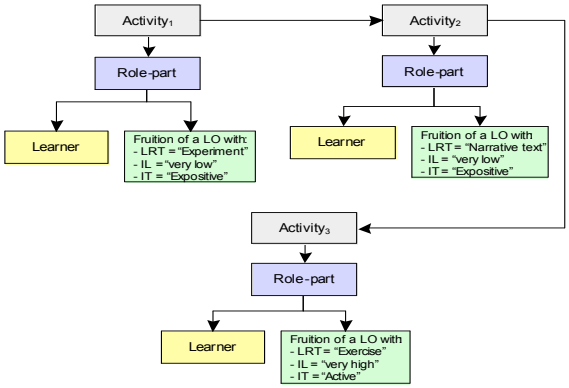

Figure 1.

The previous LD document is based on an inductive method in which learners have to study a “not interactive” LO, a “narrative text” LO and finally they have to test the acquired knowledge interacting with an “exercise”.

Didactic domain modeling (second step). In this step, domain experts model didactic domains through ontologies definition [4]. In our example we show the result of modeling process performed by an expert of “Calculus” domain. An example of ontology is shown in figure 2.

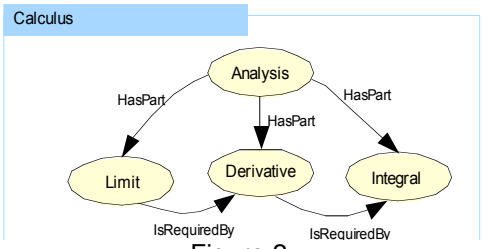

Figure 2.

Here we refer to the ontology-based approach illustrated in [3].

Unit of Learning construction (third step). Last step engages teacher that realizes the UoL overlapping domain-independent pedagogy and domain specific concepts that represent the learning objectives of the UoL.

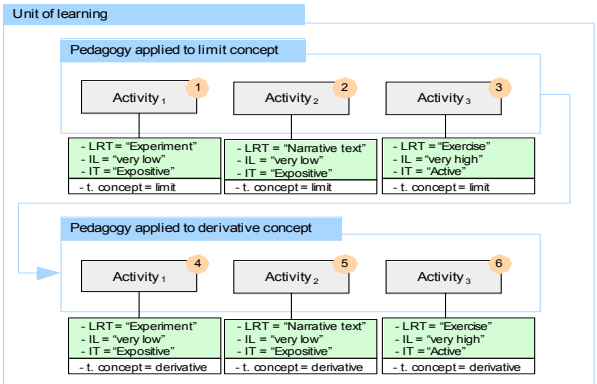

Figure 3.

In our example, teacher chooses the pedagogy (template learning design scenario) of figure 1, the ontology of figure 2 and the concept “derivative” as educational objective of his/her UoL. An automatic process uses inputs from teacher in order to:

1) Extract and organize in a sorted sequence all pre-requisites of the concepts “derivative”, from selected ontology, obtaining the path [Limit, Derivative];
2) Join the selected template learning design scenario with the sorted sequence resulting from the previous point;
3) Obtain the activities flow shown in figure 3 constructed applying the original pedagogy to each concept in the path [Limit, Derivative].

The activities flow is an enriched template learning design scenario with an additional requirement (*concept*) for each desired learning object. In the example of figure 3, the values for property called *concept* will specify that desired learning objects for activity 1, activity 2 and activity 3 have to explain the concept of “limit” in the domain of “Calculus”. Instead, desired learning objects for activity 4, activity 5 and activity 6 have to explain the concept of “derivative” in the domain of “Calculus”. So, the UoL is ready to be published and executed inside the infrastructure defined in section 5.

5. A complete e-learning scenario: a distributed delivery infrastructure

The distributed infrastructure that allows the delivery of the UoL, built in the previous section, is depicted in figure 4. It is based on three types of distributed services:

- **UoL Delivery Service**, that allows the playing of the UoL for authorized users (enrolled to the learning experience as learners, tutors, etc…).
- **Localization Service**, that allows to localize needed services. In particular, in our example, the Localization Service can be invoked in order to find available repositories storing correct (with respect to requirements in template learning design scenario) learning objects.
- **Repositories**, that are services providing two main ports. The first port is invoked to query the learning objects archive in order to obtain a list of metadata representing LOs satisfying the query. The second port (delivery) is invoked to obtain the GUI (Graphical User Interface) that renders a specific learning object delivery.

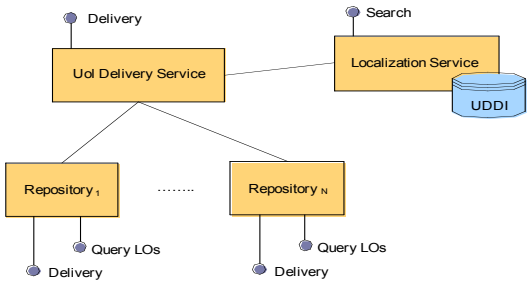

Figure 4.

Delivery ports return frames of mark-up code (typically XHTML) that represent GUIs. These ports can be implemented as WSRP (Web Service for Remote Portlet) producers [1]. *Search* and *Query LOs* ports can be implemented as standard Web Services.
In figure 5 we present the detailed architecture of UoL Delivery Service that is composed by three tiers: 1) Services Connectors for remote services invocations; 2) Enhanced IMS-LD Engine [5] for learning flow resource binding and execution; 3) UoL Player for UoL rendering.

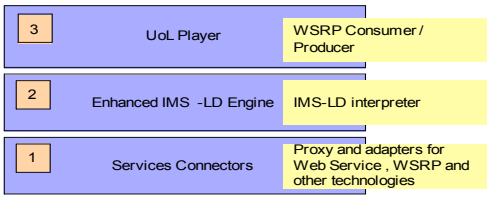

Figure 5.

An UoL delivery will walk across two phase: **startup** and **run**.

The first one is an initialization step where the UoL Delivery Service:

- Retrieves the concepts, within the UoL, interpreting the enriched LD document that describes it.
- Invokes (with respect to the didactic domain covered by UoL) the Localization Service in order to obtain the URL of one or more repositories providing learning objects. Localization Service maintains a registry (possibly UDDI) based on a primary level index that maps associations between didactic domains and repositories storing LOs related to these domains.
- Performs queries over retrieved repositories with respect to concepts obtained by UoL. Queries return lists of metadata representing learning objects related to the required concepts.
- Applies filters over lists, obtained in the previous step, using parameters associated to all activities in the UoL in order to obtain a single LO for each activity. If the applied filter returns more than one LO for the same activity the system can select a single LO using learner preferences (preferred media type) and the *technical.format* field [8] of learning objects metadata.
- Performs binding between activities and real LOs (resource identificator and repository URL) obtaining a personalized unit of learning.

The second one is the an execution step where for each activity the UoL Delivery Service:

- Detects the resource identificator and the repository URL.
- Gets WSRP mark-up code from the right repository delivery port using values obtained in the previous step.
- Composes the obtained WSRP mark-up code with the overall GUI of UoL and returns the complete WSRP mark-up code to the client.

6. Conclusion and future works

In this paper we have presented a solution for the construction and execution of personalized learning experiences based on: 1) an extension of IMS Learning Design to meet the need for domain-independent pedagogies; 2) an ontology-based approach to model didactic domains in which we can plunge the domain-independent pedagogies to obtain units of learning ready to be personalized and executed; 3) a distributed infrastructure for personalized units of learning delivery.
As future works, we are planning to develop the presented delivery architecture extending the IWT (Intelligent Web Teacher) platform implemented by MoMA [11]. Furthermore, we are planning to enrich the IWT platform with a set of authoring tools to cover all phases of units of learning building process. Nowadays, we are making experience with some open source software like Coppercore [9] for learning design scenarios execution and Reload [10] for learning design scenarios authoring.

References

[1]. Schaeck T. and Thompson R. - *Web Services for Remote Portlets (WSRP) Whitepaper*, 28 may 2003.
[2]. N. Capuano, M. Gaeta, A. Micarelli, E. Sangineto - *An Intelligent Web Tutoring System for Learning Personalization and Semantic Web Compatibility*. Proceedings of the 11th International Conference on Powerful ICT Tools for Teaching and Learning. PEG 2003, St. Petersburg, Russia, 2003.
[3]. N. Capuano, M. Gaeta, A. Micarelli, E. Sangineto - *An Integrated Architecture for Automatic Course Generation*. Proceedings of the IEEE International Conference on Advanced Learning Technologies. ICALT 2002, Kazan, Russia, 2002.

[4]. D. Fensel - *Ontologies: a Silver Bullet for Knowledge Management and Electronic Commerce*. Springer, 2001.
[5]. IMS *Learning Design Best Practice and Implementation Guide (Version 1.0 Final Specification)*. Available from: http://www.imsglobal.org/learningdesign/index.html
[6]. *IMS Learning Design Information Model (Version 1.0 Final Specification)*. Available from: http://www.imsglobal.org/learningdesign/index.html
[7]. H. Hummel, J. Manderveld, C. Tattersall and R. Koper - *Educational modelling language and learning design: new opportunities for instructional reusability and personalised learning. Int. J. Learning Technology*, Vol. 1, No. 1, 2004
[8]. *IMS Learning Resource Meta-Data Information Model (Version 1.2.1 Final Specification)* - Available from: http://www.imsglobal.org/metadata/index.html
[9]. *Coppercore project: The IMS Learning Design Engine* - Available from: http://coppercore.sourceforge.net/index.html
[10]. *Reload tool* - Available from: http://www.reload.ac.uk/tools.html
[11]. MoMA srl, *Intelligent Web Teacher v 2.0 White paper.*
[12]. Pythagoras Karampiperis and Demetrios Sampson - *Designing Learning Services for Open Learning Systems Utilizing IMS Learning Design.*