Using the Personal Competence Manager as a complementary approach to IMS Learning Design authoring

Hubert Vogten[a*], Rob Koper[a], Harrie Martens[a], Jan van Bruggen[a]

[a]*Open University of the Netherlands, The Netherlands*

Abstract: In this article TENCompetence will be presented as a framework for lifelong competence development. More specifically, the relationship between the TENCompetence framework and the IMS Learning Design (LD) specification is explored. LD authoring has proven to be challenging and the toolset currently available is targeting expert users mostly working for institutions of higher educations. Furthermore these tools re-enforce a fairly rigid top-down workflow approach towards design and delivery. This approach it is not always the most suitable model in all circumstances for all practitioners. TENCompetence provides an alternative bottom-up approach to LD authoring via its first implementation: the Personal Competence Manager (PCM). Constructs such as competence profiles and competence development programmes, let users define, modify, and acquire competences they need for achieving their personal goals. We will show how the PCM provides support for these constructs and stimulates the bottom-up development of learning materials. We will also show how these concepts can be mapped towards LD. This allows the ad hoc designs of the PCM to be captured in a Unit of Learning (UOL). These UOLs can be enhanced and eventually fed back into the PCM, therewith closing the edit cycle. This editing cycle allows for a gradual integration of bottom-up ad hoc designs with more formal top-down designs introducing LD in a gentle fashion.

Keywords: personal competence manager; competence development; learning design; authoring.

Introduction

Emerging e-learning standardization initiatives have led to a number of interesting new specifications and standards. One of those initiatives is IMS Learning Design (IMS, 2003; Koper & Olivier, 2004; Olivier & Tattersall, 2005). LD is a formal language for the specification of learning designs using semantically meaningful concepts from the pedagogical domain. The most relevant objectives achieved by applying LD are formalization, reproducibility and reusability of the learning designs. LD is a very expressive specification capable of describing a wide variety of learning designs. However it is also a very complex and complicated specification. The current toolset supporting LD is still very closely and directly informed by the specification itself and requires a profound understanding of the specification. Therefore LD is used mainly in institutions for higher education where sufficient expertise is available to work with the current toolset.

*Corresponding author: Hubert Vogten. Open University of the Netherlands, PO Box 2960, NL-6401DL, Heerlen, The Netherlands. Email: hubert.vogten@ou.nl.

The recently launched TENCompetence initiative targets the development of an infrastructure for lifelong competence development. TENCompetence has the ambition to support formal and informal learning during the lifetime of an individual. TENCompetence ambitions reach beyond the scope of the educational institutions.

The first release of the TENCompetence software is called the Personal Competence Manager (PCM). The PCM provides an integrated environment for both learning and authoring without making a clear distinction between the two modes. This article will show how this aspect can be beneficial for the easy creation of simple UOLs. A UOL is the collection of files including the learning design expressed in LD that is ready to be deployed in a suitable runtime environment. We will also see how UOLs can be enhanced and in turn be reused in the PCM closing the editing cycle. In this way a gentle introduction to LD authoring can be achieved using the PCM as an initial more loosely authoring environment. In later stage LD can be used to capture, enhance and redeploy the created learning experience when needed.

IMS Learning Design tools

LD is targeted at the educational designer allowing ‘learning designs’ to be explicitly modeled using semantically meaningful concepts from the pedagogical domain. Although expressive, the specification is also very complex due to the numerous language constructs, its declarative nature, and its fairly generic vocabulary (Griffiths & Blat, 2007; Olivier, 2004). However LD was developed with a toolset in mind that would help the educational designer in using LD (Griffiths, Blat, García, Vogten, & Kwong, 2005). Three years after the release of LD, a user community is established working on the development and enhancements of these tools. So far this has resulted in a toolset dealing with LD editing and authoring aspects on the one hand and run-time delivery aspects on the other hand (Griffiths et al., 2005). These authors categorize the tools on two dimensions:
Higher vs. lower level tools: This dimension is related to the level of expertise in LD required from the user of the tool.
General purpose versus specific purpose tools: This dimension deals with the pedagogical scope of the tools. Specific purpose tools will hide complexity by translating generics into the specific context and filling in and leaving out optional elements where appropriate. Generic purpose tools however, will allow authoring and delivery of LD in all its glory.

Although efforts have been made to create or adapt specialized authoring tools with some success such as COLLAGE (Hernández-Leo et al., 2006), HyCo-ALD (Berlanga & García, 2007) and MOT+ (Paquette, De la Teja, Léonard, Lundgren-Cayrol, & Marino, 2005), in general most of the available tools that are LD compliant on levels A, B and C must be categorized as generic and still rather low level. They allow the editing of the complete LD specification and keep very close to the specification. A typical example in this category is Reload (Miligan, Beauvoir, & Sharples, 2005; Reload Learning Design Editor, 2007) which is by far the most popular LD editor at the moment. However, as a consequence, an ample understanding of LD is required to use these tools. An even more profound understanding of LD is required when advanced concepts as described by levels B and C of LD, are required by the design. In general, this level of understanding is limited to expert educational designers and is rarely found in practitioners such as teachers. This leaves many practitioners out of the direct loop of designing and adapting UOLs. Some LD tools available allow limited post design runtime adaptations through code introspection (Zarraonandia, Dodero, & Fernández, 2006). However, these post design runtime adaptations will not be reflected in the UOL and therefore will be lost in the next run (Tattersall et al., 2005a) of the same UOL.

Furthermore, the current toolset imposes a, be it an implicit, top-down approach of the overall design and delivery process. This is further encouraged by the separation of the authoring environments and runtime delivery environments (Tattersall, Vogten, & Koper, 2005b). Typically, elicitation and selection of the type of educational scenarios is the first step in the design process followed by the coding of the scenario into a UOL using the authoring environment. Next this UOL is published so it can be delivered to teachers and learners via a runtime environment such as CopperCore (Martens, Vogten, Van Rosmalen, & Koper, 2004). This UOL can be adapted, refined and improved in following design cycles repeating the whole process again. This workflow resembles the waterfall approach of traditional software development and has advantages especially in cases where the same UOL is offered to different groups for lengthy periods of time (Tattersall et al., 2005b). This approach can help enhance the quality of the learning experience because educational scenarios are made elicit in a very explicit and formal manner allowing reflection on the quality and effectiveness of the designs. This quality control can be further enhanced by collecting runtime data as is demonstrated in aLFanet (Van Rosmalen et al., 2007). Concluding it can be said that with the current toolset practitioners must adopt this top-down approach and need to have ample knowledge of LD. Therefore, LD has been taken up mainly by institutions for higher education where the required expertise can be found.

In the following sections we will present the TENCompetence domain model (Koper, 2006) followed by the first implementation based on this domain model called the Personal Competence Manager. We will discuss how the PCM can complement the current toolset available for LD. We will discuss how the PCM empowers individual users to create basic UOLs using a bottom-up approach without the need for any specific LD expertise. Furthermore we will discuss how these UOLs can be fed back into the PCM allowing a more controlled and reproducible provisioning of the learning process.

TENCompetence Domain Model

The aims of TENCompetence has been defined on the web site (TENCompetence consortium, 2007) as:
"A competence-based approach to lifelong learning aims to take account of all the informal and experiential learning that an individual acquires during the course of his or her lifetime rather than focusing solely on academic or theoretical achievement. This way an individual can make the most of his or her achievements, be they scholastic, work-based or the result of a leisure pursuit. The concept of competence development bridges

the worlds of education, training, knowledge management, human resource management & informal learning in all domains which, hitherto, operated in relative isolation in respect of one another. A competence approach to lifelong learning ensures that the pursuit of a learning goal does not happen in a vacuum, but instead is bound to a precisely defined purpose such as an occupation, a profession, a market or a particular life or work situation."

TENCompetence is finding solutions for seven major problem areas (Koper & Specht, 2007) currently preventing an infrastructure for lifelong competence development to become a reality. TENCompetence is focusing at the needs of the individual lifelong learner that want to maintain their autonomy and control as much as possible. This aspect of user empowerment is typical for initiatives in the area of Personal Learning Environments (CETIS, 2007). Users are expected to develop their own competences, not merely by taking up competence development courses, but also by actively contributing to these courses.

Before discussing the TENCompetence domain model we have to give our definition of a competence. We define a competence as the estimated ability of an actor to deal with certain critical events, problems or tasks that can occur in a certain situation. This estimation can be based on: self assessment, informal assessments by others, formal assessments by others or automated assessments. Competences can be attributed to an individual person, but also to a team or to an organization. We will use the term actor as a container for individuals, teams or organizations. Dealing with these critical events, problems or tasks requires a number of different competences. This set of required competences is called the competence profile (CP). Actors will develop and maintain many competences during their lifetime and these competences can be considered dispositions of these actors. A competence is a highly situational concept meaning that the definition and understanding of a competence is attributed to the relationship between actor and environment. Some of these competences are highly specific and others are transferable to more general situations. The specific labels we give to competences and CPs are determined by a community of practice that consists of all participants who are regular actors in that situation. Therefore, the competences for the same profession, job or function may vary from community to community even though the required behaviors are exactly the same. Finally a competence is a latent characteristic of an actor: it is neither directly visible nor measurable. Only the concrete performances of actors are visible. From these performances we infer these latent characteristics and get an idea of the competences these actors have acquired.

The TENCompetence domain model is the conceptual model for lifelong competence development and it describes the various entities and their relationships that play a role. The domain model is informed by our definition of competences, by the principles of LD and finally by the concepts of learning networks (Koper, 2005).

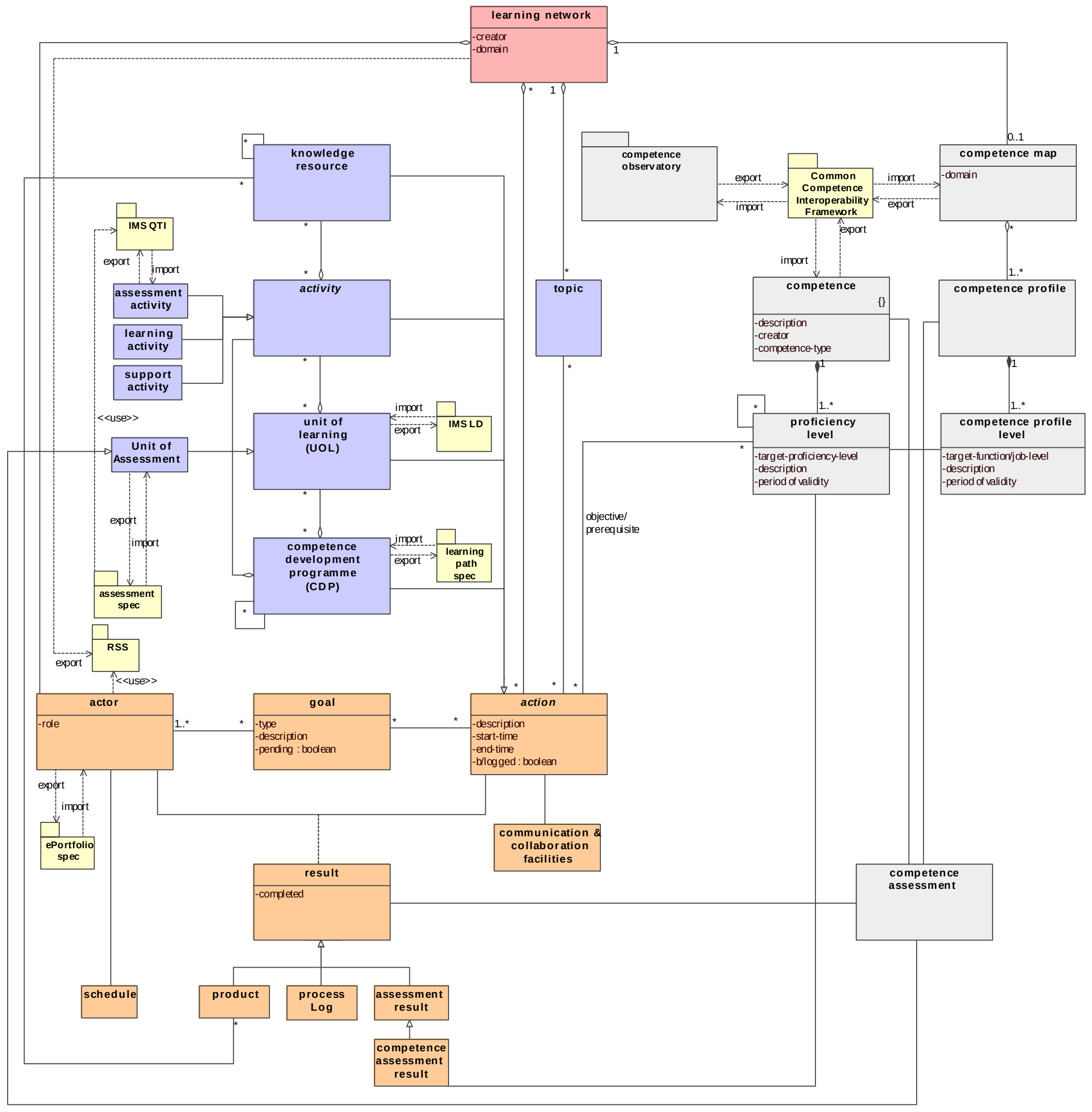


Figure 1 UML Class Diagram of the TENCompetence Domain Model.

Figure 1 depicts the UML (OMG, 2003) class diagram of the TENCompetence domain model. The model is divided into four separate modeling areas: learning materials, actor performance, competence model and finally the learning network, or community of practice, as a container for all these concepts. The domain model will now be elaborated in more detail through these concepts and their relationships.

Actors will perform actions in order to achieve their goals. Typical goals are: keeping up-to-date with a profession; improving particular competences; comparing competences with peers. These actions are always performed in the context of a community of practice which is represented by a learning network in the domain model. While performing actions, actors have the possibility and are stimulated to provide support to each other by means of communication and collaboration facilities.

By performing the actions actors leave traces of their performance behind. These traces can take many forms ranging from mere activity logs to learning outcomes. These traces will be used to infer the measure in which an actor has acquired certain competences. Because competences are highly situational concepts their definitions are specific to the learning network. Competences can be acquired at different levels. These levels are modeled via proficiency levels, each representing a discrete ordinal measure to which a competence has been acquired. Competences can alternatively and/or additionally be assessed through specific competence assessments.

A CP is a collection of competences, targeted at specific proficiency levels which are required to be able to deal with certain critical events, problems, or tasks in a certain situation. CPs can be further split up into competence profile levels representing the levels of a profession, e.g. like trainer, master and trainee.

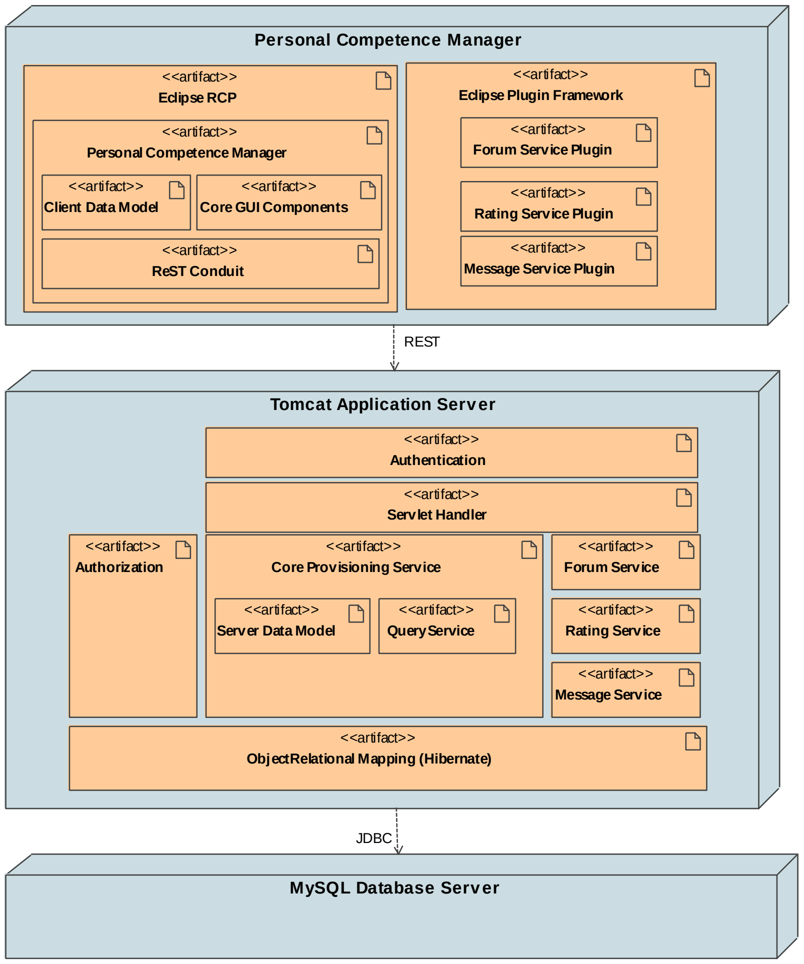
Each learning network will define and describe its own set of competences and CPs. This set makes up the competence map of that community of practice. Some of these competences are generic and/or common to a domain but merely described differently for a particular community. A competence observatory will maintain the common and more formal definitions and descriptions for these generic competences ensuring transferability between communities of practice. Communities may contribute their competences and CPs to this observatory and thereby share their definitions with other communities. Equally communities may decide to reuse competences and CPs present in the competence observatory.

Finally, the model for actions is informed by the concepts of learning design. Actions can be divided into: knowledge resources, activities, units of learning, and competence development programmes (CDP). A CDP is an ordered set of activities and units of learning that have to be mastered to attain a certain competence or CP. CDPs can be exported to a learning path specification. We will see how the PCM, besides using LD as formalism for learning design which is quite natural, also uses LD as formalism for this learning path specification.

The Personal Competence Manager

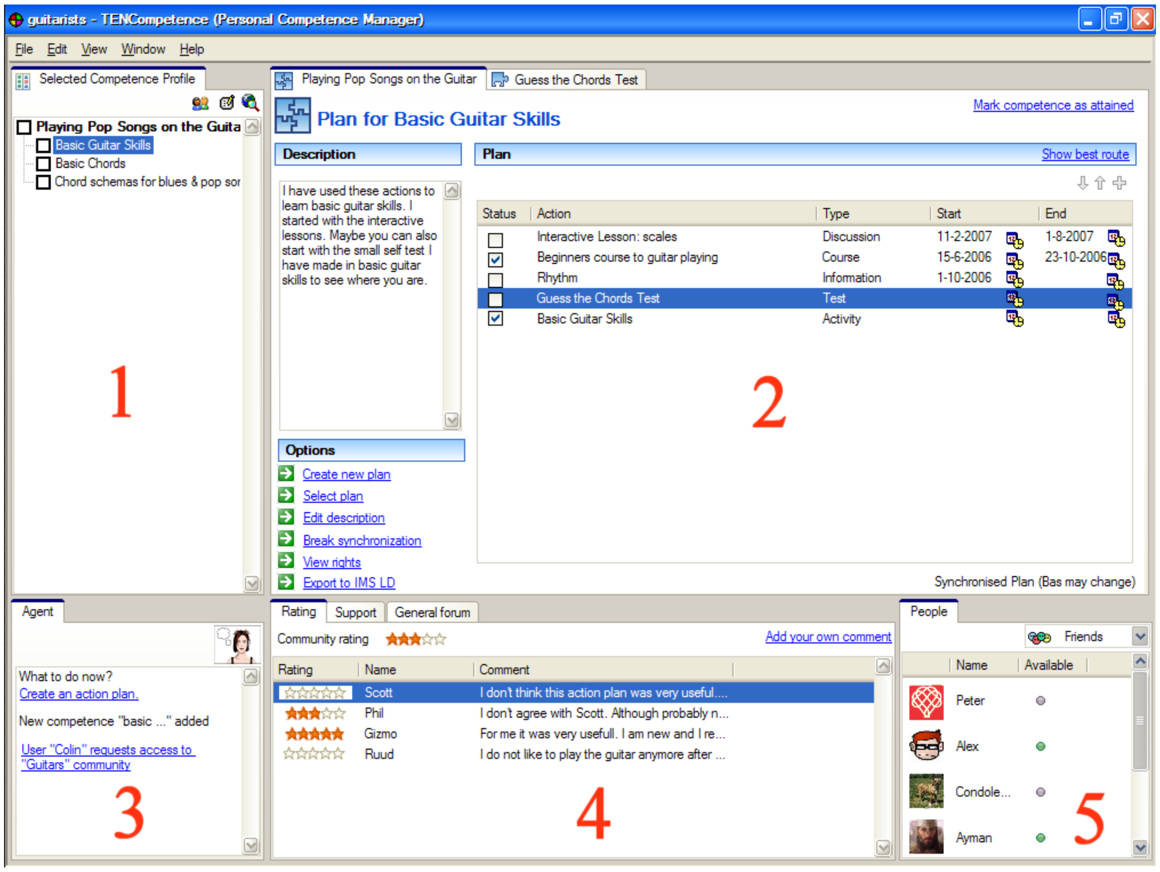
The PCM is a client server application implementing a simplified version of the TENCompetence domain model. The PCM lets users manage their own competence profiles in the context of learning networks for which they are registered. These competence profiles can be used to reflect on their personal competences with respect to this profile. The PCM helps users find most suitable learning materials and learning opportunities for acquiring these competences. Furthermore the PCM encourages users to create and share their personal contributions with the rest of the community. For this purpose design and runtime are closely integrated in the PCM. The PCM does not work with concepts like releases or versions and the learning opportunities are continuously changing and hopefully thereby improving. This is very much in contrast with the top-down approach supported by the current LD toolset.

At the time of writing of this article the design stages have been concluded and coding of the PCM has started. The software is available as open source on SourceForge at: http://sourceforge.net/projects/tencompetence/. Figure 2 depicts the overall architecture for the PCM. The PCM is developed as a desktop client application using the Eclipse Rich Client Platform (Eclipse, 2007) allowing it to run on a range of platforms. The client is extensible via the Eclipse plug-in framework. The client communicates with the server using REST (Fielding, 2000) providing an easy to use interface for other clients in the future. The PCM server is deployed on a Tomcat application server. It provides several services which are governed by a servlet handler which in effect is acting as a simple service bus. The server core provides basic provisioning and query services for the data model objects we already encountered in the TENCompetence domain model.


Personal Competence Manager
<<artifact>> Eclipse RCP
<<artifact>> Personal Competence Manager
<<artifact>> Client Data Model
<<artifact>> Core GUI Components
<<artifact>> ReST Conduit
<<artifact>> Eclipse Plugin Framework
<<artifact>> Forum Service Plugin
<<artifact>> Rating Service Plugin
<<artifact>> Message Service Plugin
REST
Tomcat Application Server
<<artifact>> Authentication
<<artifact>> Servlet Handler
<<artifact>> Authorization
<<artifact>> Core Provisioning Service
<<artifact>> Server Data Model
<<artifact>> Query Service
<<artifact>> Forum Service
<<artifact>> Rating Service
<<artifact>> Message Service
<<artifact>> ObjectRelational Mapping (Hibernate)
JDBC
MySQL Database Server
Figure 2 Personal Competence Manager Architecture.

Besides the core service, a number of additional, more autonomous services are provided by the server such as the forum, rating and message services. The idea is that these services will be extensible in future releases. Access to these services is governed by an authorization module. Finally, data persistence is managed through an object relational mapping using Hibernate.

The core functionality of the PCM will be discussed using detailed screen designs that were available at the time of writing of this article.


guitarists - TENCompetence (Personal Competence Manager)
File Edit View Window Help
Selected Competence Profile
Playing Pop Songs on the Guita
Basic Guitar Skills
Basic Chords
Chord schemas for blues & pop sor
1
Playing Pop Songs on the Guitar
Guess the Chords Test
Plan for Basic Guitar Skills
Mark competence as attained
Description
Plan
Show best route
2
Options
Create new plan
Select plan
Edit description
Break synchronization
View rights
Export to IMS LD
Synchronised Plan (Bas may change)
Agent
3
Rating
Support
General forum
Community rating
Add your own comment
4
People
Friends
5

I have used these actions to learn basic guitar skills. I started with the interactive lessons. Maybe you can also start with the small self test I have made in basic guitar skills to see where you are.

Status	Action	Type	Start	End
☐	Interactive Lesson: scales	Discussion	11-2-2007	1-8-2007
☑	Beginners course to guitar playing	Course	15-6-2006	23-10-2006
☐	Rhythm	Information	1-10-2006	
☐	Guess the Chords Test	Test		
☑	Basic Guitar Skills	Activity		

What to do now?
Create an action plan.

New competence "basic ..." added

User "Colin" requests access to "Guitars" community

Rating	Name	Comment
☆☆☆☆☆	Scott	I don't think this action plan was very useful....
★★★☆☆	Phil	I don't agree with Scott. Although probably n...
★★★★★	Gizmo	For me it was very usefull. I am new and I re...
☆☆☆☆☆	Ruud	I do not like to play the guitar anymore after ...

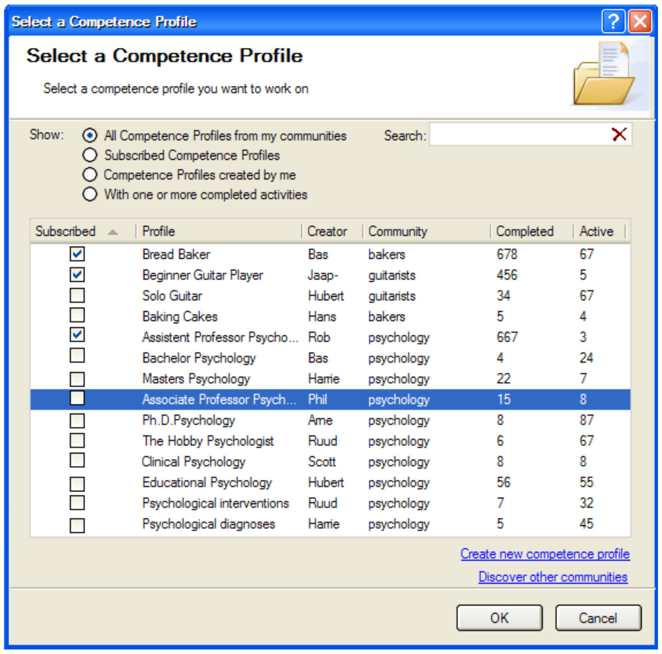
Name	Available
Peter	●
Alex	●
Condole...	●
Ayman	●

Figure 3 Screenshot of the Personal Competence Manager User Interface Design.

Figure 3 depicts the main application window of the PCM. The PCM user interface can be roughly divided into two areas. The top half area (1 & 2) contains views and editors intended for viewing and editing CPs, competences, and actions. The lower half of the main window (3, 4 & 5) contains views that help and support the users in their task performed in the upper half. In Figure 3 the ‘Plan for Basic Guitar Skills’ is the active editor (2) and therefore provides the context for all views in the lower part of the screen.

Figure 3 represents a snapshot of a situation where a learning network, in the PCM represented by it synonymous term community, already has been created and some content has been added to this network. Furthermore any user may decide to start a new learning network at any moment in time. Learning networks are not governed by any central authority and can be set up by anyone. The creator of a learning network is also the owner of the community and determines policies for the learning network access. This principle of an entity owner controlling its access rights applies for almost all entities. The general idea is that the PCM should tend to openness whenever possible in order to stimulate active participation and contributions of all community members. The PCM relies on the principles of self-organization to regulate this process (Hadeli, Zamifirescu, van Brussel, Holvoet, & Steegmans, 2003).

View 1 of Figure 3 shows the CP selected by the user. A user can select CPs via the competence selection dialog shown in Figure 4.


Select a Competence Profile

Select a Competence Profile

Select a competence profile you want to work on

Show: [✓ All Competence Profiles from my communities] Subscribed Competence Profiles Competence Profiles created by me With one or more completed activities

Search:

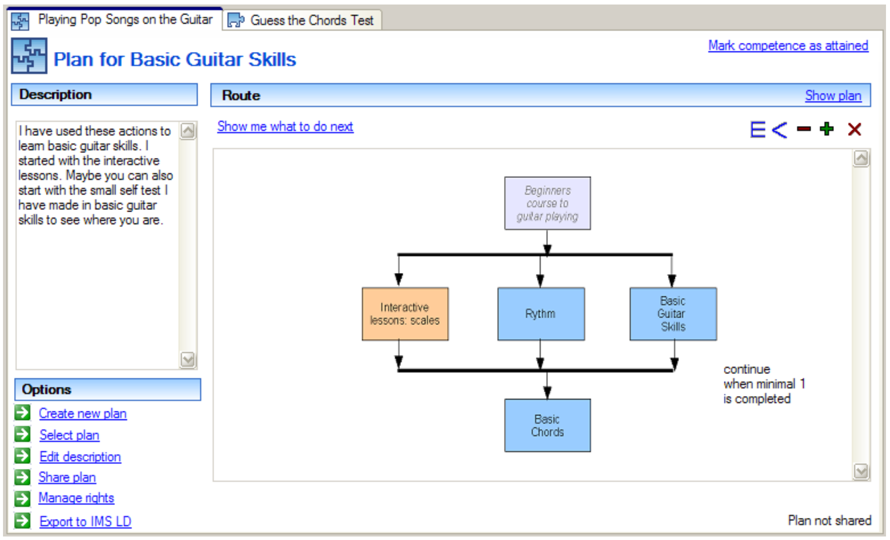
Subscribed	Profile	Creator	Community	Completed	Active
✓	Bread Baker	Bas	bakers	678	67
✓	Beginner Guitar Player	Jaap-	guitarists	456	5
☐	Solo Guitar	Hubert	guitarists	34	67
☐	Baking Cakes	Hans	bakers	5	4
✓	Assistent Professor Psycho...	Rob	psychology	667	3
☐	Bachelor Psychology	Bas	psychology	4	24
☐	Masters Psychology	Harrie	psychology	22	7
☐	Associate Professor Psych...	Phil	psychology	15	8
☐	Ph.D.Psychology	Arne	psychology	8	87
☐	The Hobby Psychologist	Ruud	psychology	6	67
☐	Clinical Psychology	Scott	psychology	8	8
☐	Educational Psychology	Hubert	psychology	56	55
☐	Psychological interventions	Ruud	psychology	7	32
☐	Psychological diagnoses	Harrie	psychology	5	45

Create new competence profile

Discover other communities

Figure 4 Competence Profile Selection Dialog.

Once the profile has been selected, the user may access the competence development plans for these competences. These will be opened in the CDP editor depicted in Figure 3 (2). For any competence many CDPs may exist. The CDP is a container for a number of actions that represent a learning design targeted at the associated competence. A user may decide to simply start performing one of these actions by selecting them from the list, but can alternatively also decide to get some advice about the best next actions to take by clicking the ‘Show best route’ link. The PCM will now show a flow chart like navigational view of the CDP revealing the relations between the actions of the CDP.



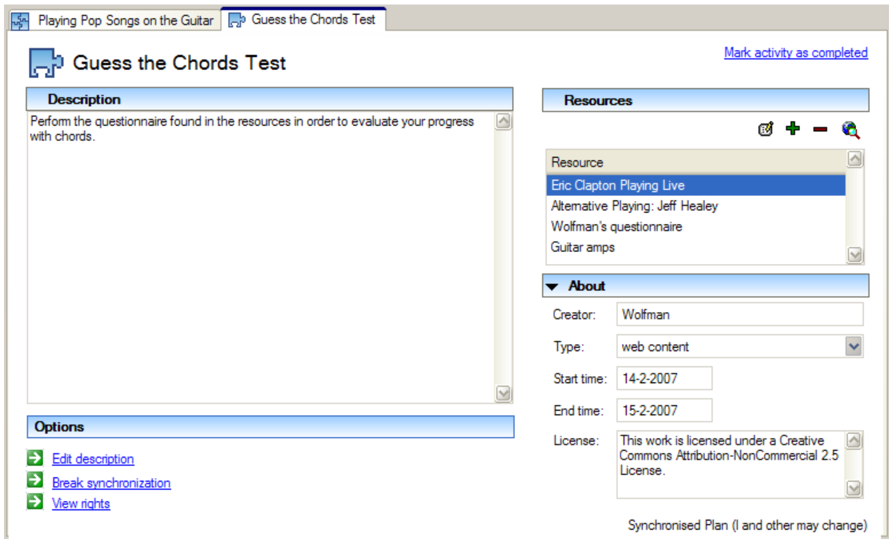
I have used these actions to learn basic guitar skills. I started with the interactive lessons. Maybe you can also start with the small self test I have made in basic guitar skills to see where you are.

Figure 5 Navigation View.

Figure 5 depicts this navigation view of the CDP. Actions in the CDP can be structured into sequences and selections. These concepts are very much informed by LD. By clicking ‘Show me what to do next’ the user activates the navigation service to receive help in selecting the best next action. In the first release of the PCM this navigation service will be implemented using the simplest of algorithms possible: suggest the next action which is not yet completed, but needs completing. In the future advanced navigation services will be available that also take personal preferences, learning styles and past performances of others into account.

Users can actively contribute to a CDP by adding new actions or modifying the detailed learning path as shown in Figure 5. By applying these changes to a CDP the user is sharing the changes with others. A shared CDP is behaving like a Wiki with regard to this sharing aspect. Alternatively, a user has may decide to create a different CDP for the selected competence all together. This CDP will show up as alternative when another user is selecting a CDP in order to acquire this competence.

When a user decides to perform an action from the CDP the action editor depicted in Figure 6 is opened.


Playing Pop Songs on the Guitar | Guess the Chords Test

Guess the Chords Test

Mark activity as completed

Description

Perform the questionnaire found in the resources in order to evaluate your progress with chords.

Options

- Edit description
- Break synchronization
- View rights

Resources

Resource
Eric Clapton Playing Live
Alternative Playing: Jeff Healey
Wolfman's questionnaire
Guitar amps

About

Creator:	Wolfman
Type:	web content
Start time:	14-2-2007
End time:	15-2-2007
License:	This work is licensed under a Creative Commons Attribution-NonCommercial 2.5 License.

Synchronised Plan (I and other may change)

Figure 6 Action Editor.

The concept of an action was also informed by LD. Actions can take two forms: a link to an external implementation like for instance a link to a run of a UOL, or an action that is managed by the PCM itself. An action has a description instructing the learner what he is expected to do. Furthermore there are resources available helping the learner to perform this action. An action can be modified by changing the description and/or by modifying the resource associated with it.

The bottom half of Figure 3 that is composed out of 3, 4 & 5 contain services that will help the users in performing their tasks. The agent view (3), informs the user about events occurring in the community. Next (4) there is a group of services that are helping the user to perform the selected action (2) which consists of a rating service, a support forum, and a general discussion forum. Finally, there is a member services showing all the members of the community. The PCM will support FOAF (FOAF project, 2007) to support the creation of ad hoc user communities. The PCM may be extended with additional services via the standard plug-in mechanism provided by Eclipse.

Capturing the Competence Development Plan using LD

We have seen that the TENCompetence domain model and therefore also the PCM are informed by LD, especially the part dealing with learning materials. Concepts such as learning activities, support activities, learning resources and units of learning can be directly mapped onto concepts defined in LD. Furthermore, competences themselves can be mapped through LD prerequisites and objectives. Although this mapping may seem not that obvious at first, LD started out as a specification for modeling competence based learning (Tattersall, Vogten, & Hermans, 2005). In the LD specification references are made to the 'IMS Reusable Definition of Competency or Educational Objective' specification (IMS, 2007) for both the prerequisites and objectives sections. Finally, the CDP brings all these components together and can be mapped onto the method section of LD. The CDP consists of a simple list of actions that may be performed by the user. This list can be mapped directly to a selection in LD. In more advanced designs of the learning path within the CDP there can be a mix of selections and sequences of actions. These constructs map directly onto the selections and sequences as defined in LD. So all CDPs main constructs can be mapped to equivalent LD constructs. Table 1 depicts the global mapping of the main entities found within the TENCompetence domain model onto the LD elements. Note that most elements have a direct one-to-one mapping with the exception of the CDP which requires a more elaborate mapping because it provides the container for all other elements.

Table 1 Translation of main TENCompetence Domain Model entities onto IMS Learning Design elements.

TENCompetence domain model entity	IMS Learning Design element(s)
knowledge resource	learning-object
learning activity	learning-activity
support-activity	support-activity
assessment-activity	learning-activity with IMS Question and Test Interoperability content.
unit-of-learning	No mapping required because this is a place holder for the UOL itself. This allows a UOL to be fed back into the PCM.
Competence	prerequisite or learning-objective
CDP	unit of learning containing one learner role, the competences addressed by the associated CP expressed as objectives, selections and sequences as defined by the learning path of the CDP and a play for wrapping the activities.

The user can initiate the transformation by clicking the 'Export to LD' option. The resulting UOL can be stored for publication or if needed, for further refinements and enhancements.

Just as important as the data model entities themselves, is the way how they are created. We have shown via the wire frames that editing a CDP and its components can be done without any knowledge or awareness of LD whatsoever. The PCM does not presume any particular workflow and allows a bottom-up approach because no distinction is made between design time and runtime. Via the principle of "what you see is what you get", the PCM allows the active participation of learners in the creation of educational materials and scenarios. A learning design can become an emergent property of the work of a whole community. At any point in time a user may decide to capture the outcomes of this process in the form of a UOL by performing an export. The reasons for doing so can be numerous like being able to:

- Reflect on the *quality* of the learning design which can be achieved more easily now because the design is made explicit and formal;
- Reuse the same materials for another group of learners making the learning experience *reproducible*;

- Improve the design by adding more sophisticated features adding a great deal of *extensibility* and *flexibility* to the PCM;
- Share the design with other practitioners who could be using other e-learning environments. LD provides this *interoperability*;
- Capture a design as a permanent record for the learning experience provided. This record in the form of a UOL can provide *accountability* independent of a particular version of particular software.

The PCM uses LD as an export format for its CDPs. The exported UOL only captures parts of the functionality offered by the PCM because it is merely a snapshot of the design modeled through the CDP, not of the process that has lead to it. The context in which the CDP has been created, like the groups discussion, ratings of alternatives CDPs, creation of ad hoc communities working together on the topic, building of reputations of users within the community etc. is not captured by the resulting UOL. Also personalized data such as the planned start and end dates for activities are not captured in the UOL because LD specifies a learning design at the level of user roles rather than at the level of an individual. This is also the reason that a UOL needs to be populated through the run mechanism before it can be deployed: the personal information has to be added by the runtime engine in order to deliver the design.

The example depicted by Figure 2 and Figure 5 would result in the following LD fragment which has been greatly simplified for readability purposes.

```
<learning-design>
  <title>Plan for Basic Guitar Skills</title>
  <learning-objectives>
    <item identifierref=”basic_guitar_skills"/>
  </learning-objectives>
  <components>
    <roles>
      <learner identifier="learner"><title>Learner</title></learner>
    </roles>
    <activities>
      <learning-activity identifier="a_beginners_course_guitar_playing">
        <title>Beginners course guitar playing</title>
      </learning-activity>
      <learning-activity identifier="a_interactive_lessons:_scales">
        <title>Interactive lessons: scales </title>
      </learning-activity>
      <learning-activity identifier="a_rhythm">
        <title>Rhythm</title>
      </learning-activity>
      <learning-activity identifier="a_basic_guitar_skills">
        <title>Basic Guitar Skills</title>
      </learning-activity>
      <learning-activity identifier="a_basic_chords">
```

```
        <title>Beginners course guitar playing</title>
      </learning-activity>
      <activity-structure identifier="seq_1" structure-type="sequence">
        <learning-activity-ref ref=" a_beginners_course_guitar_playing " />
        <activity-structure-ref ref="sel_1"/>
        <learning-activity-ref ref=" a_basic_chords "/>
      </activity-structure>
      <activity-structure identifier="sel_1" structure-type="selection">
        <learning-activity-ref ref="a_interactive_lessons:_scales" />
        <learning-activity-ref ref="a_rhythm"/>
        <learning-activity-ref ref="a_basic_guitar_skills"/>
      </activity-structure>
    </activities>
..</components>
  <method>
    <play>
      <act>
        <role-part>
          <role-ref ref="learner"/><activity-structure-ref ref="seq_1"/>
        </role-part>
      </act>
    </play>
  </method>
<learning-design>
```

The translation of the constructs in the PCM has been fairly straightforward according to the rules described in Table 1. All exported CDPs have such a fairly basic learning design because the possibilities to vary this design are relatively limited compared to the modeling possibilities and freedom offered by LD.

The exported UOL can be edited with all available LD authoring tools, enhancing the design where needed. These tools allow more sophisticated editing of the UOL because they make all constructs of LD available to the user. However, this also implies that from this point onwards ample LD expertise is required to maintain the UOL. An enhanced design can be fed back into PCM by creating a new action that wraps this UOL. The PCM integrates the CopperCore (Martens et al., 2004) LD runtime environment in order to deploy the modified UOL. Without this integration reuse of the enhanced UOL with in the PCM would not be impossible because the PCM would not be capable of interpreting the enhanced design itself. This also implies that once a UOL has been enhanced it can only be re-edited via the regular LD tools.

This action that wraps the exported UOL, can replace the original CDP because its learning objectives are targeted towards the same competence as the CDP it was derived from. The action containing referring to the UOL could also be included into a bigger CDP which in turn could be exported to another UOL resembling the Russian dolls model. This way the bottom-up authoring approach provided by the PCM can be integrated with the more formal top-down design approach associated with current LD authoring environments, providing the best of both worlds. Figure 7 depicts this editing cycle.

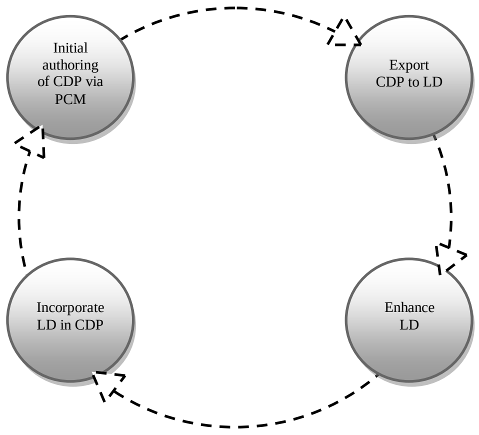


Figure 7 The Editing Cycle.

In order for the round-trip editing cycle to succeed, a specific deployment approach for the exported UOL has to be chosen. Because the PCM relies on the ad hoc formation of communities per CDP, the resulting runtime delivery of the UOL should adhere to these communities as well. The proper integration of the PCM and the CopperCore runtime engine is crucial because the CDP membership and the UOL run subscriptions have to be kept synchronized at all time. Therefore exactly one run will be created of a UOL for every CDP containing that UOL. Users are added and removed from a run in accordance to their registration for the containing CDP. So de-facto, the CDP population and the run population are kept in sync. For this first release of the PCM it is assumed that the UOL will allow users to be “rolled on” and “rolled off”. It is however possible to use LD constructs that forbid this type of continues registration by forcing users to be added in cohorts. These restrictions will simply be ignored in the first release of the PCM and need further investigation in the future.

Because the exported UOL is wrapped with its own action when it is imported in the PCM, all regular support tools such as ratings and forums and self assessments are available when executing the UOL. Therefore there is no need to synchronize outcomes of the CopperCore runtime engine with the PCM. However in future releases, this could be the case. The CopperCore Service Integration framework (Vogten & Martens, 2006) provides a first direction towards a closer integration when the need should arise in future release of the PCM.

The assignment of roles is another issue that needs to be resolved for the editing cycle of Figure 7 to work. In LD, users can fulfill multiple roles in one design. A user needs to be assigned to one or more of such roles before the user can actively participate in a run. In those cases where a UOL is merely exported and not modified, this assignment is simple and can be done without any additional actions because there will be only one role defined in the exported UOL as we have seen a few paragraphs ago in the simplified example. However when the generated UOL is enhanced it is perfectly reasonable to have a more complex role structure. When the role mappings are the same for every user this is no real problem because the role assignments can still be handled automatically. However when the design assumes users to take on different roles, the mapping is not that straightforward anymore. Intervention by a user or intelligent role mapping services may be required in those cases. For the first release of the software, simple mappings are assumed by default and user interaction is required for these more complex situations. For future releases this is an issue that needs further exploration.

Conclusion and future work

In this paper we have argued that LD is a very generic, complete and therefore also a complex specification. For a non specialist the use of LD in the daily teaching practice is only feasible with the help of sophisticated and probably specialized tools. The current state-of-the-art LD tools can be categorized as generic and LD aware, requiring a specialist's expertise. Furthermore, an external data representation such as LD, leads to a natural separation of design time and runtime tooling. This in turn introduces a top-down workflow approach to provisioning of learning through consecutive stages of design, authoring, publication, user management, and finally delivery.

Although this is a perfectly sound approach, it can also be problematic in cases where practitioners prefer a more bottom-up approach without having a very elicit view on the design. These practitioners will probably prefer an environment where there is no strict separation between design time and runtime. This approach is often more appealing, intuitive, and suitable for the initial stages of a design. The PCM provides this type of editing. Especially the CDP editor provides an easy means for creating a learning design that is build up from actions which in turn can be organized into sequences and selections. In a later stage, especially when a design has matured and proven to be particularly successful, there may be a need to redeploy the same design for a different group of learners. The ad hoc design can be exported to a UOL making the design formal. Other reasons for exporting the design could be the need to reuse the same design with other resources. It could also be the case that it would be worthwhile to redeploy the same design in a totally different e-learning environment. Quality assurance could be another reason for formalizing an ad hoc design into a UOL. The exported basic designs can be improved upon with the normal LD tools and then be reused in the context of the PCM itself or by any other LD compliant environment. The PCM integrates LD tools such as CopperCore for this purpose.

The approach presented in this paper allows for an easy introduction of users to LD in cases when there are clear benefits for the user to do so. The generated LD can be used as it is, but can also be improved upon. Whatever is the case, the user will need ample LD knowledge from that point onwards. Nevertheless, the user has a clear motivation, one of the aforementioned reasons, to make the additional effort needed to become familiar with LD. Although the user can feed back the altered UOL into the PCM, once exported and modified, the point of no return has been passed. The PCM will not be able to help the user maintain the UOL. The reason for this is that advanced LD concepts have no equivalent in the PCM such as e.g. support for advanced personalization, support for different pedagogies, support for multiple roles and support for advanced role based workflow. Therefore, the PCM will never be able to really replace the existing LD tools but must rather be considered to offer a gentle introduction to LD for those practitioners who are new to the tools and concepts and do not have or see a need to invest in them right from the start.

When exporting the CDP to a UOL two distinct approaches can be defined. First, the one discussed so far, where the produced UOL is reused in the context of the CDP. This export may assume that the services offered by the CDP will be available to the UOL as well because the UOL is reused in the same context. However, if the UOL is reused in a totally different context from the CDP, another type of export may be required because referenced and implicit services have to be defined and bundled in some form into the UOL. Although initial steps have been taken in this direction with e.g. the integration of assessment services through IMS Question and Test Interoperability (IMS, 2006; Vogten et al., 2007), there is still further work to be done in this area especially regarding the standardization of service interfaces. For now the PCM will only support the first type of LD export requiring the PCM to run the constructed UOL.

At the moment of writing several initiatives are improving on the available LD tools. In fact some of these initiatives have been bundled in the TENCompetence integrated project (TENCompetence consortium, 2007). It will be interesting to see how these tools develop and what this means for the integration in the PCM. A first step towards this integration is the harmonization of the look and feel of both the CDP editor and the Reload based LD editor. Work towards this direction has recently started and although at the time of writing this development is still very much in its early stages it looks like a promising step towards a more seamless integration of the PCM and LD.

Until that time, the approach presented in this article combining the implicit bottom-up design method provided by the PCM and the more formal elicit top-down design favored by the current LD toolset offers a practical alternative.

Acknowledgements

The authors wish to thank the management and staff of the Schloss Dagstuhl International Conference and Research Center for Computer Science for providing a pleasant, stimulating and well organized environment for the writing of this article. Authors' efforts were (partly) funded by the European Commission in TENCompetence (IST-2004-02787) (http://www.tencompetence.org).

References

Berlanga, A., & García, F. (2007, August 17). IMS LD reusable elements for adaptive learning designs. *Journal of Interactive Media in Education*,from http://jime.open.ac.uk/2005/11.

CETIS (2007). *PLE Report.* Retrieved August 01, 2007, from the website of CETIS: http://wiki.cetis.ac.uk/Ple.

Eclipse (2007, May 1). *Eclipse*. Retrieved May 07, 2007, from Website of Eclipse Consortium: http://www.eclipse.org.

Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures.

FOAF project (2007). *FOAF*. Retrieved from Website of the Friend of a Friend project: http://www.foaf-project.org/.

Griffiths, D., & Blat, J. (2007, October 1). The Role of Teachers in Editing and Authoring Units of Learning Using IMS Learning Design. *International Journal on Advanced Technology for Learning, 2*(4).

Griffiths, D., Blat, J., García, R., Vogten, H., & Kwong, K.-L. (2005). Learning Design Tools. In R. Koper & C. Tattersall (Eds.). *Learning Design, a Handbook on Modelling and Delivering Networked Education and Training* (pp. 109-135) (chap. 7). Heidelberg: Springer.

Hernández-Leo, D., Villasclaras-Fernández, E., Asensio-Pérez, J., Dimitriadis, Y., Jorrín-Abellán, I., Ruiz-Requies, I., et al. (2006). COLLAGE: A collaborative Learning Design editor based on patterns. *Educational Technology & Society, 9*(1), 58-71.

IMS (2003). *IMS Learning Design Specification.* Retrieved July 03, 2003, from Website of IMS Global Learning Consortium: http://www.imsglobal.org/learningdesign/index.cfm.

IMS (2006). *IMS Question and Test Interoperability*. Retrieved January 12, 2006, from Website of IMS Global Learning Consortium: http://www.imsglobal.org/question/index.html.

IMS (2007). *IMS Reusable Definition of Competency or Educational Objective.* Retrieved from Website of IMS Global Learning Consortium: http://www.imsglobal.org/competencies/index.html.

Koper, R. (2005). Designing Learning Network for Lifelong Learners. In R. Koper & C. Tattersall (Eds.). *A Handbook on Modelling and Delivering Networked Education and Training* (pp. 239-252) (chap. 14).Springer.

Koper, R. (2006). *TENCompetence Domain Model*. Retrieved May 12, 2007, from http://hdl.handle.net/1820/649.

Koper, R., & Olivier, B. (2004). Representing the Learning Design of Units of learning. *Educational Technology and Society, 7*(3), 97-111.

Koper, R., & Specht, M. (2007). TenCompetence: Lifelong Competence Development and Learning. In M. Sicilia (Ed.). *Competencies in Organizational E-Learning: Concepts and Tools* (pp. 230-247) (chap. 11). Idea Group Inc.

Martens, H., Vogten, H., Van Rosmalen, P., & Koper, E. J. R. (2004). *CopperCore.* Retrieved January 14, 2005, from SourceForge: http://coppercore.org.

Miligan, C., Beauvoir, P., & Sharples, P. (2005, July 1). The Reload Learning Design Tools. *Journal of Interactive Media in Education,*from http://jime.open.ac.uk/2005/06.

Olivier, B. (2004). *Learning Design Update*. Retrieved August 4, 2006, from http://www.jisc.ac.uk/uploaded_documents/Learning_Design_State_of_Play.pdf.

Olivier, B., & Tattersall, C. (2005). The Learning Design Specification. In R. Koper & C. Tattersall (Eds.). *Learning Design. A Handbook on Modelling and Delivering Networked Education and Training* (pp. 21-40) (chap. 2).

OMG (2003). *Unified Modeling Language (UML).* Retrieved November 06, 2003, from http://www.omg.org.

Paquette, G., De la Teja, I., Léonard, M., Lundgren-Cayrol, K., & Marino, O. (2005). An Instructional Engineering Method and Tool for the Design of Units of Learning. In R. Koper & C. Tattersall (Eds.). *Learning Design. A Handbook on Modelling and Delivering Networked Education and Training* (pp. 161-184) (chap. 9).Springer.

Reload Learning Design Editor (2007). *Reload Learning Design Editor*. Retrieved June 12, 2006, from the website of the Reload Project: http://www.reload.ac.uk/.

Tattersall, C., Vogten, H., Brouns, F., Koper, R., Rosmalen, P. v., Sloep, P., et al. (2005a). How to create flexible runtime delivery of distance learning courses. *Educational Technology & Society, 8*(3), 226-236.

Tattersall, C., Vogten, H., & Hermans, H. (2005). The Edubox Learning Design Player. In R. Koper & C. Tattersall (Eds.). *Learning Design, a Handbook on Modelling and Delivering Networked Education and Training* (pp. 303-310) (chap. 19). Heidelberg: Springer.

Tattersall, C., Vogten, H., & Koper, R. (2005b). An Architecture for the Delivery of E-learning Courses. In R. Koper & C. Tattersall (Eds.). *Learning Design. A Handbook* *on Modelling and Delivering Networked Education and Training* (pp. 63-73) (chap. 4).Springer.

TENCompetence consortium (2007). *TENCompetence Project.* Retrieved February 01, 2006, from Website of the TENCompetence project: http://www.tencompetence.org

Van Rosmalen, P., Vogten, H., van Es, R., Passier, H., Poelmans, P., & Koper, R. (2007). Authoring a full life cycle model in standards-based, adaptive e-learning. *Educational Technology & Society*, *9*(1), 72-83.from http://www.ifets.info/journals/9_1/7.pdf

Vogten, H., & Martens, H. (2006). *CopperCore Service Integration.* Retrieved February 02, 2006, from Website of the CopperCore Service Integration framework: **http://sf.net/projects/ccsi**

Vogten, H., Martens, H., Nadolski, R., Tattersall, C., Rosmalen, P. v., & Koper, E. J. R. (2007, in press). CopperCore Service Integration. *Journal on Interactive Learning Environments*,(special issue)

Zarraonandia, T., Dodero, J. M., & Fernández, C. (2006). Crosscutting runtime adaptations of LD execution. *Educational Technology & Society*, *9*(1), 123-137.