Project no. 027087

TENCompetence

Building the European Network for Lifelong Competence Development

Project acronym: Integrated Project TENCompetence

Thematic Priority: 2.4.10

**D6.1 – Report with summary of WP6 outputs of month 1-18**

Due date of deliverable: 31-05-2007  
Actual submission date: 03-01-2008

Start date of project: 01-12-2005  
Duration: 4 years

University of Bolton

Version 1.0

<table>
<thead>
<tr>
<th>Dissemination Level</th>
<th>Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PU Public</td>
<td>X</td>
</tr>
<tr>
<td>PP Restricted to other programme participants (including the Commission Services)</td>
<td></td>
</tr>
<tr>
<td>RE Restricted to a group specified by the consortium (including the Commission Services)</td>
<td></td>
</tr>
<tr>
<td>CO Confidential, only for members of the consortium (including the Commission Services)</td>
<td></td>
</tr>
</tbody>
</table>
Project Deliverable Report

D6.1 – Report with summary of WP6 outputs of month 1-18

Work package  WP6 Learning Activities & Units of Learning
Task  Task 6.1, Task 6.2, Task 6.3 and Task 6.4
Date of delivery  Contractual: 31-05-2007  Actual: 03-01-2008
Code name  D6.1  Version: 1.0  Draft  Final
Type of deliverable  Report  Security (distribution level)  Public
Contributors  David Griffiths, Tom Franklin, Paul Sharples, Phillip Beauvoir, Scott Wilson, Yongwu Miao, Colin Tattersall, Rob Koper, Krassen Stefanov, Milen Petrov, Adelina Aleksiya-Petrova, Toni Navarrete, Helena Batlle, Demetrios Sampson, Pythagoras Karampiperis, Judith Schoonenboom, Marco Luccini, Linda Napoletano
Authors (Partner)  UB, OUNL, SU, FBM-UPF, CERTH/ITI, UvA, ILABS
Contact Person  David Griffiths (UB)
WP/Task responsible  David Griffiths (UB)
EC Project Officer  Mr. Martin Májek

Abstract (for dissemination)  This deliverable consists of four documents:
- Main text summarising work in WP 6 in months 1 - 18
- Annex 1, IMS LD Authoring
- Annex 2, Assessment
- Annex 3, Runtime

Keywords List  IMS Learning Design Authoring, Assessment Model, Run-time IMS LD Services, Pedagogical Templates, Units of Learning

TENCompetence Project Coordination at: Open University of the Netherlands
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
Tel: +31 45 5762624 – Fax: +31 45 5762800
The material reported in this deliverable is extensive.
To ease the burden on the reader it has been broken down into four documents:

- The main text summarising work in WP 6 in months 1 – 18
- Three annexes, each of which constitutes a detailed report on WP 6 activities in that area.
  - Annex 1, IMS LD Authoring
  - Annex 2, Assessment
  - Annex 3, Runtime

Each of these is delivered as a separate file, with its own page numbering.
Project Deliverable Report

D6.1 – Main body of the report

<table>
<thead>
<tr>
<th>Work Package</th>
<th>WP6 Learning Activities &amp; Units of Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Task 6.1, Task 6.2, Task 6.3 and Task 6.4</td>
</tr>
<tr>
<td>Date of delivery</td>
<td>Contractual: 31-05-2007 Actual: 03-01-2008</td>
</tr>
<tr>
<td>Code name</td>
<td>D6.1 Version: 1.0 Draft ☐ Final ☑</td>
</tr>
<tr>
<td>Type of deliverable</td>
<td>Report</td>
</tr>
<tr>
<td>Security (distribution level)</td>
<td>Public</td>
</tr>
<tr>
<td>Contributors</td>
<td>David Griffiths, Tom Franklin, Paul Sharples, Phillip Beauvoir, Scott Wilson, Yongwu Miao, Colin Tattersall, Rob Koper, Krassen Stefanov, Milen Petrov, Adelina Aleksieva-Petrova, Toni Navarrete, Helena Batlle, Demetrios Sampson, Pythagoras Karampiperis, Judith Schoonenboom, Marco Luccini, Linda Napoletano</td>
</tr>
<tr>
<td>Authors (Partner)</td>
<td>UB, OUNL, SU, FBM-UPF, CERTH/ITI, UvA, ILABS</td>
</tr>
<tr>
<td>Contact Person</td>
<td>David Griffiths (UB)</td>
</tr>
<tr>
<td>WP/Task responsible</td>
<td>David Griffiths (UB)</td>
</tr>
<tr>
<td>EC Project Officer</td>
<td>Mr. Martin Májek</td>
</tr>
<tr>
<td>Abstract (for dissemination)</td>
<td>Work package 6 of TENCompetence is developing an infrastructure for learning activities and assessment activities which implements specifications developed by the project, and others which are already established. This report provides a summary of work carried out in the first 18 months of the project in relation to Pedagogic Models, IMS-LD Authoring, IMS-LD Runtime Systems, and the TENCompetence Assessment specification and proof of concept tool.</td>
</tr>
<tr>
<td>Keywords List</td>
<td>IMS Learning Design Authoring, Assessment Model, Run-time IMS LD Services, Pedagogical Templates, Units of Learning</td>
</tr>
</tbody>
</table>

TENCompetence Project Coordination at: Open University of the Netherlands
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
Tel: +31 45 5762624 – Fax: +31 45 5762800
TABLE OF CONTENTS

A. READING GUIDE............................................................................................................................. 5
B. EXECUTIVE SUMMARY................................................................................................................... 6
C. INTRODUCTION TO THE WORK CARRIED OUT .......................................................................... 9
D. TASK 1. IMS LD AUTHORING COMPONENT .............................................................................. 12
   D.1 THE EVOLVING STRATEGY FOR LD AUTHORING IN TENCOMPETENCE ....................... 12
   D.2 SIMPLE AND ELABORATED COMPETENCE DEVELOPMENT PLANS ............................. 14
   D.3 RECourse 1.0, THE LD AUTHORING COMPONENT ............................................................. 14
   D.4 ACHIEVEMENTS IN THE LD AUTHORING COMPONENT .................................................... 18
   D.5 EVALUATION ........................................................................................................................... 19
   D.6 API DEFINITIONS FOR WP6 ................................................................................................. 20
PEDAGOGICAL MODELS .................................................................................................................. 21
ASSESSMENT SPECIFICATION ...................................................................................................... 22
   D.1 AN INTEGRATED COMPETENCE DEVELOPMENT AND COMPETENCE ASSESSMENT PROCESSES MODEL .................................................. 22
   D.2 A TECHNICAL APPROACH TO MODEL AND DELIVER CLASSIC AND NEW FORMS OF ONLINE COMPETENCE ASSESSMENT ........................................ 23
   D.3 THE FIRST VERSION OF TENCOMPETENCE ASSESSMENT SPECIFICATION ............... 23
   D.4 THE PROOF-OF-CONCEPT ASSESSMENT AUTHORING TOOL ......................................... 24
   D.5 REDESIGN OF THE TENCOMPETENCE ASSESSMENT MODEL ......................................... 26
E. IMS LD RUN-TIME COMPONENT .............................................................................................. 27
   E.1 INTRODUCTION ..................................................................................................................... 27
   E.2 IMS QTI RUNTIME DEVELOPMENT ..................................................................................... 27
   E.3 IMS LD – SCORM INTEGRATION ............................................................................................ 30
   E.4 INTEGRATION OF IMS LD AND SCORM ............................................................................. 32
   E.5 THE SOFTWARE DEVELOPED ............................................................................................... 33
   E.6 CONNECTION PROTOCOL .................................................................................................... 34
F. ROADMAP FOR LEARNING ACTIVITIES & UNITS OF LEARNING .................................. 35
   F.1 AUTHORING LEARNING ACTIVITIES AND UNITS OF LEARNING .................................. 35
   F.2 ASSESSMENT ........................................................................................................................... 36
   F.3 RUNTIME ................................................................................................................................ 38
   F.4 RELEASE SCHEDULE .............................................................................................................. 39
G. LINKS TO THE PRINCIPAL OUTCOMES OF WP6 ............................................................. 41
REFERENCES ..................................................................................................................................... 43
A. Reading guide

A great deal of work has been carried out in WP 6 of TENCompetence, and the reader of this document will immediately note that this deliverable is rather extensive. It has, however, been structured so that the body of the report provides a relatively brief overview, while the extensive appendices provide detailed reports on the work carried out in the various areas of activity. It is hoped that in this way the reader will be able to identify the main achievements of the work package, while having access to more detailed information should this be of interest.

Those readers who would like only a summary of the achievements of the work package are directed to the

- Section B: Executive summary (page 4 of this document)
- Section F: Roadmap for learning activities and units of learning (page 32)

Those readers who would like a more detailed picture of the work carried out are invited to carry on to read the core report, i.e. sections D to F of this document. The sections of the report correspond to the three main tasks being undertaken by the work package, each of which is described in a self contained text.

- Task 6.1: IMS LD Authoring Component
- Task 6.2: Assessment Specification
- Task 6.3: IMS LD Run-time Component

The reader can choose to read all three, or only those which are relevant.

It is not intended that any reader should have to read the entire document complete with appendices. Those readers who want access to detailed documents and results will find that appendices and source code are referenced from the three task descriptions.

For quick access to the principal WP 6 outcomes (specifications, software and related publications) an index of links is provided in Section G (page 37)

For detailed reports on the main lines of work carried out, see the four annexes to this deliverable

- Annex 1, IMS LD Authoring
- Annex 2, Assessment
- Annex 3, Runtime
B. Executive Summary

WP 6 is directed at the development of models & tools to realise the creation, storage, search, retrieval and quality rating of learning activities & units of learning. As is stated in the DOW, this is to be achieved with Open Source, standards based technologies. Consequently, as suggested in the DOW, the integrated system to be developed by the WP is unified by the use of IMS specifications.

The first step taken was to review the necessary specifications and the tooling available. As a result of this analysis two principal challenges were identified for WP6. Firstly, the lack of a mature tool set for creating and running activities based on IMS LD and IMS QTI. Secondly, issues relating to the specifications themselves, specifically the non-availability of runtime services for IMS LD (such as forums, chat, etc.), and the limited scope of IMS QTI when applied to the assessment of competences. Consequently work in this first phase has been focused on redressing these problems, to ensure that the standards based integration of the system is a viable strategy for the project. This approach informed the three tasks which make up WP6: IMS LD Authoring, the TENCompetence Assessment Specification, and the Runtime Component. We will now summarise the achievements of the project for each of these tasks.

Task 1. IMS LD Authoring. In this task the main focus of work has been placed on the redesign of the Reload LD Editor using the Eclipse graphic editor frameworks (GEF), the first time this environment has been used for an IMS specification editor. The result was ReCourse v.1.0, which has been publicly released as open source (see This is a fully functioning IMS LD level A editor with a graphic interface (rather than the tree based editor which was used in Reload and similar tools), an integrated resource management and innovative facilities for publishing to a runtime system. Collaboration has been established with a user group who already use Reload to develop IMS LD courses in an authentic context (one of the few groups currently carrying out this work). ReCourse provides all the functionality previously used by teachers in this group in designing courses with Reload. This group provided the development team with initial designs for the LD Editor, and participated in the design, and carried out evaluation of ReCourse. ReCourse also resolves a number of underlying problems in the way in which Reload handled the specification and managed resources. Evaluation of ReCourse has been carried out with this user group, with very positive results. Detailed information about the ReCourse editor, and the evaluation carried out, is available in Annex 1 to this deliverable.

In the first 12 months of the project work was also carried out into the identification of pedagogic models for competence development. The results of this work were submitted in M6.1, at month 12, and included templates for the representation of pedagogic strategies in IMS LD. At this stage it was decided, however, that the appropriate context for developing pedagogic models was WP2, and consequently this work is reported in the context of that work package in deliverable ID2.7.

Task 2. Assessment Specification, is developing a formal assessment specification and a corresponding assessment authoring tool to support practitioners in designing and customizing assessments. The first achievement in this task has been the development of a high level model which integrates competence development and competence assessment processes. This work is summarised in the main text of this
deliverable, and presented in more detail in Annex 2. This model has informed the
development of a technical approach to model and deliver classic and new forms of
online competence assessment through a combined use of IMS LD, IMS QTI, and
assessment-specific services. This work has been extensively published, and links to
the preprints are available in the web links in section G of this report.

The first version of the TENCompetence Assessment Specification was developed on
the basis of the high level model developed in this task. This was achieved by
adapting the OUNL/CITO model, which was simplified in order to facilitate mapping
to IMS LD and IMS QTI. Proof of concept tools were developed for use with the
specification, and evaluated. These are described in the body of the deliverable (with
more detail available in Annex 2) and are publicly available for download. This
practical implementation of the specification has been valuable, as it has revealed that
the document centric nature of the First Assessment Specification means that it is
difficult to map onto IMS LD, which is an activity based specification. As a result an
alternative approach to adapting the OUNL/CITO was established. This is described
in this deliverable, and informs current work which will be reported in D 6.2.

Task 3. Runtime. One of the principal barriers to adoption and use of IMS LD has
been the lack of flexible runtime services (e.g. a forum, or a chat, run external
learning objects and tests, etc.) which can be launched from an LD player (typically a
system based on the CopperCore LD Engine). The principal focus of work on this task
has been to resolve this problem. A first version of the runtime system has been tested
and released. This builds on the most advanced currently available system for
delivering services with IMS LD, which involves integrating services in the CCSI
layer of CopperCore. The first service to be integrated was a SCORM player, making
it possible for the two leading interoperability specifications for learning flows (IMS
LD and SCORM) to work together. This work has been completed, tested and
released as Open Source, and is reported in detail in Annex 3. This experience,
however, showed that the integration of additional services would be a time
consuming process. Given the large number of services which it would be desirable to
use with IMS LD, it was therefore concluded that it was not a generalisable approach
to service integration, as required in TENCompetence. Consequently an entirely new
approach to a LD service integration has been established, making use of Widgets as a
shared service. A specification for this connection protocol has been developed, and is
included in Annex 3. Implementation work is currently underway on a trial server
which will be delivered in D 6.2.

A second line of work provided runtime support for the TENCompetence Assessment
Specification. This is to be expressed in XML by using a combination of IMS LD and
QTI. The CopperCore engine is available as Open Source, and can run the current
version of the IMS LD specification. There is, however, no available runtime engine
for QTI 2.1. Consequently the APIS QTI runtime engine has been updated so that it is
compliant with IMS QTI v. 2.1. This has been published as Open Source.

In summary, the tasks and outputs defined for DIP 1 were all completed successfully,
with the first versions of the Assessment Specification and Connection Protocol
produced, and implementation completed on the first versions of the Authoring,
Assessment and Runtime systems. The outputs from this first phase of work, together
with the insight obtained, mean that the work package is well placed for further
developing a standards based integrated system in the second period of work during DIP 2.
C. Introduction to the work carried out

This deliverable reports on work carried out in TENCompetence WP 6, Learning Activities & Units of Learning. Learning activities are the designed or performed activities of a person that are directed at the attainment of an explicit or implicit learning objective. We refer to designed learning activities as units of learning (UOLs), such as courses, workshops, lessons, etc. Assessment activities are also included within the remit of this work package, and they have a particularly important role within the context of competence assessment.

In the TENCompetence project as a whole a key aspect is the integration of formal and informal competence acquisition. In this work package, however, the focus is on supporting learning and assessment processes which are formalised, in the sense that they have been planned in advance for a particular purpose. Informal learning in TENCompetence is supported by the communities established in the Personal Competence Manager (PCM) and the services provided by the PCM to facilitate and structure their interactions and exchanges. In working with formalised learning and assessment activities in TENCompetence the suite of interoperability specifications published by IMS Global Learning Consortium is a key enabler. More specifically, IMS Learning Design (IMS LD) can be used to define learning activities, while IMS Question and Test Interoperability (IMS QTI) can be used in combination with IMS LD to support assessment activities. These specifications and the infrastructure associated with them provide attractive functionality for our purposes, but there are other systems which could be used. However, the requirement within TENCompetence for Open Source and interoperable solutions means that there is no viable alternative approach for the project. While a number of valuable pilot projects have been carried out using IMS LD and QTI in teaching, their use in other than funded projects remains rather limited. Moreover, their application to the context of Life Long Competence development is new. This means that there is no solid body of existing practice which can simply be adopted by this work package. There is, however, a technical infrastructure available as the result of work carried out within the Valkenburg Group, the UNFOLD project, and a number of publicly and privately funded development projects.

In carrying out this new work the project team faces two principal challenges.

1) The tool set for creating and running activities based on IMS LD and IMS QTI is not yet mature

2) It has been clear for some time that there are outstanding issues to be addressed with both these specifications, specifically
   a) The set of services available for use with IMS LD Units of Learning is very limited
   b) The flexibility and scope of IMS QTI is too restricted for the purposes of competence assessment in TENCompetence
The work carried out in WP6 addresses both these issues for the three principal areas in which the work package is active, as defined in the Detailed Implementation Plan (DIP) 1 and DIP 2.

- Task 6.1: IMS LD Authoring Component
  - A new authoring component has been developed and evaluated, building on the success of the open source reference application Reload, which supports non-technical users in working with the concepts of IMS.
  - The work is based on the existing IMS LD specification
  - Tools design and evaluation has been carried out in close collaboration with the user group. This has been facilitated by the Learning Technologies Research Group at Liverpool Hope University, who are an associate partner of the project. This team is one of the very few groups worldwide who are using IMS LD in producing and running courses with students in an authentic setting.

- Task 6.2: Assessment Specification
  - Work in this area is driven by the development of an assessment specification. This was used to develop proof of concept tools.
  - Tools for authoring and running the new assessment specification have been developed. These have demonstrated the need to revise the specification in order to ensure interoperable units of assessment.
  - A strategy for revising the TENCompetence assessment specification has been developed, which will export to IMS QTI 2 used in combination with IMS LD. Work on this will be submitted, together with exemplars as a component of D 6.2.

- Task 6.3: IMS LD Run-time Component
  - In the work reported here IMS LD was successfully integrated with SCORM, and the results are reported in this deliverable.
  - The QTI runtime engine APIs has been updated so that it can run assessments defined with QTI v 2.1
  - Like task 6.2, Assessment, this task also requires a new specification to deal with services. It is, however, driven primarily by practical problems of implementation, rather than by a theoretical understanding of the domain. A protocol for the integration of IMS LD runtime services has been developed using leveraging the emerging W3 standard for Widgets. The results of implementation work on this new system built to this specification will be submitted in D6.2, in month 30.

Work on the first twelve months of work on Learning Activities and Units of Learning was described in the project milestone M6.1, completed in month 12, and the relevant sections of this milestone are included as components of the Annexes to this report. M6.1 was particularly significant for the work on pedagogic models, as the first phase of work on this aspect was completed at that date, in accordance with the
project plan. Since that date two significant changes have taken place in the work package, which should be borne in mind when reading this deliverable.

Firstly, in DIP 2 work on Pedagogical models for lifelong competence development has been moved to WP2. This is because although these pedagogical models are relevant to WP6, they also inform the vision of the project as a whole, and all activities with the integrated system being produced by WP3. The present deliverable is defined under DIP 1, and so it includes work on pedagogical models within its scope. For a full picture of current work on pedagogical models, however, the reader is directed to WP2.

Secondly, the Work package leader, CERTH/ITI, found that they were unable to fulfil their planned role in the project due to the unavailability of key personnel. While they continue to be involved in the project, this is at a much reduced level which does not enable them to maintain their leadership role in the Work package, nor to carry out the development tasks which were foreseen for them.

Work package leadership has now passed to The University of Bolton, following the approval of DIP-2 in April. Work on the work package has continued despite this disruption to work package coordination and management, although while the reorganization was being carried out there was inevitably some impact on the speed of progress. The principal long term impact has been on task 6.1 (LD Authoring) where it had been foreseen that CERTH/ITI would play a key role. The necessary changes to respond to these developments have been incorporated into the DIP 2 plan.

While these changes caused some delays, these have now been recovered.
Task 1. IMS LD Authoring Component

D.1 The evolving strategy for LD Authoring in TENCompetence

According to the original TENCompetence Detailed Implementation Plan 1 (DIP 1), Task 1 in work package 6 is to Research and Develop new and existing flexible pedagogical models and learning activity models, which can be used in conjunction with the tools developed in the work package to support competence development activities using the infrastructure. The assumption behind this task was that the resulting pedagogical models would be implemented in IMS LD, made available through the TENCompetence infrastructure, and would constitute the means whereby competence development could be achieved by users of the system.

Project Milestone M6.1 accordingly identified a methodology whereby this could be achieved, based on the work of (Draganidis and Mentzas, 2006). This consists of the following key steps: (a) the creation of a competence model through the identification of required job roles and relevant competences, (b) the assessment of existing competences, (c) the gap analysis between existing competences and the required competences for a specific job role, (d) the definition of competence development programmes to minimize the identified gaps and (e) the continuous performance monitoring and assessment to confirm improvement.

This approach, however, became gradually less appropriate as the functionality of the TENCompetence Personal Competence Manager (PCM) and the services on which it draws became clearer. The PCM is a new type of application, and as such its functionality is the result of work carried out in the first phase. In this work (as described in deliverables D3.1 and D3.2) two aspects emerged as being particularly significant:

1. The PCM supports individuals in managing the whole range of their Lifelong Learning activities, including managing their participation in relevant social networks and planning and carrying out their competence development plans. This task is not restricted to a single institution, which is the focus of the Draganidis and Mentzas approach to competency management.

2. The PCM has stronger than anticipated functionality for defining competence development plans, complete with resources and activities. These can be followed by the author, shared, and used by others. Thus a significant proportion of pedagogic planning in TENCompetence can now carried out with the PCM, while WP6 addresses the need for more focused tools and methods for user by teachers and learning designers. Consequently the pedagogic models need to inform work carried out in all various parts of the project, and integrated in the PCM. For this reason Task 1 Research and Develop new and existent flexible pedagogical models and learning activity models was moved to WP2 in the Detailed

---

1 As Draganidis and Mentzas say, the Competency Management which they describe “is a practice that becomes more and more important in private and public organizations, helping them to attract and develop talented employees, identify the right person for a job position, performing succession planning, training analysis and other core human resources functions.” (Draganidis and Mentzas, 2006)
Implementation Plan 2 (DIP2) and is reported in the context of that work package in deliverable ID2.7.

A user of the PCM Rich Client, or another application which uses the same services, can define a Competence Development Plan (CDP), creating a flow of resources and activities. This is in effect a simple learning design, which can be expressed as an IMS LD Unit of Learning (UOL). Editing a CDP and its components, however, can be done without any knowledge of LD whatsoever.

This approach presents a number of advantages in the design of Units of Learning.

- It does not presume any working method or any order of actions.
- It makes no distinction between design and delivery mode making transition between the one and the other seamlessly.
- It allows the active participation of learners in the design process.

In this way, a learning design can become an emergent property of a community. At any point in time a user may decide to capture the outcome of this process. The reasons for doing so can be numerous: reflection on the learning design, ability to provide the same materials for another group of learners, improve the design by adding more sophisticated features, export the design to another e-learning environment, and store the design as a record of the learning experience. At present the PCM does not export CDPs as IMS LD, but it is planned that future releases will be able to represent CDPs as a Level A UOL. The exported UOL can then be edited with any available LD tools. From this point onwards expertise in IMS LD may be required to improve the design.

The consequences of this insight were significant for IMS LD authoring tool development. The Authoring tool no longer had to provide an overarching structure for the development of IMS LD based learning activities and competence development plans, as this is provided by the PCM. Rather the need was for a tool which could enable teachers (whether formally in that role or not) to develop UOLs and CDPs which are more sophisticated than those which can be generated by the PCM. Similarly the research agenda in WP 6 LD authoring has shifted away from the overall architecture of support for competence development activities (which is more of a focus in WP3), and towards issues of representations and methodologies for supporting relatively non-technical users in developing elaborated CDPs.

A potential problem in this approach is that the present Rich Client interface to the PCM enables users to author all aspects of the system (e.g. competence structures, networks...), and so it is inevitably quite complex. This would not suit teachers or others who simply want to define a simple competence development plan. To meet the needs of these users work is underway on a web based tool which enables the user to create and follow a simple CDP using PCM services, and export this to IMS LD level A. We describe this within WP6 as a pre-authoring application because it can be used to develop a basic structure which can then be elaborated using more powerful tools. This work is being carried out in the context of WP6 in DIP2 and will be reported in D6.2.
D.2 Simple and elaborated Competence Development Plans

In TENCompetence a Competence Development Plan (CDP) is a set of competence development activities linked to the attainment of a specified competence. The flow of activities, and the resources which they use, constitute a learning design, that is to say a set of activities directed towards a learning objective.

The PCM can be used alone, without the need for additional editors, such as the LD Author. Some users, however, will find it useful to separate the learning design from the PCM, and perhaps from TENCompetence context as a whole. For this to be useful (and to conform to the requirements of TENCompetence) the separated learning design must be represented in some interoperable format. The most appropriate is IMS LD, Level A.

We distinguish between:

- **Basic** Competence Development Plans. A basic CDP is generated using the features available in PCM, or the equivalent features in the LD Author. A basic CDP cannot be very complex in structure, because the actions available in the PCM are limited (typically associating resources with activities, sequencing them, etc.). It can be represented using a subset of IMS LD Level A.

- **Elaborated** Competence Development Plans. An elaborated CDP includes competence development activities using IMS LD Level A (and optionally B & C) without any constraint.

The main focus of the work described here is on the Elaborated Competence Development plans, where WP6 provides both authoring and runtime support, and this will remain the case in the next period of project work. However, under DIP 2 WP6 is working at the level of Basic Competence Development Plans, (as briefly described in the paragraph on Pre-authoring above).

D.3 ReCourse 1.0, the LD Authoring component

D.3.1 Building on the Reload LD Editor

In month 14, when it became clear that it would not be possible to develop the planned inference engine, a strategy was required to repair the lack of an effective LD authoring environment. This has been done by returning to the core task defined in the TENCompetence Detailed Implementation Plan 2 (DIP 2) which is defined as

“Extend and evaluate existing open source IMS LD Authoring and Run-time tools to support the requirements of lifelong competence development and to function as components and/or services into the overall TENCompetence system and evaluate the usability of the produced software components/tools.”

The Reload LD Editor, mentioned explicitly in the DOW, is widely accepted as the reference implementation of an IMS LD authoring tool, and it has been widely used in the development of UOLs in projects such as UNFOLD and LearningNetworks, Collage, and in the Design for Learning programme of JISC. Consequently it was a
natural choice to focus on this took particularly as the consortium includes people who have previously worked on Reload.

The creation of an effective IMS LD editing component by building on the Reload LD Author has not, however, been a trivial task.

Despite its success, the Reload LD Editor has a number of shortcomings for the purposes of the TENCompetence project. While it enables users to access all parts of the specification, it does so using a tree interface which requires users to have a detailed knowledge of the specification. The interface makes use of a traditional GUI toolset of tables, trees, lists and text fields which is available in Eclipse. This set of interface devices leads to a formalised workflow that reflects the hierarchical structure of the XML binding of the IMS specification rather than a natural workflow reflecting processes of pedagogical design or lesson plan design. This is shown in figure 1 on the following page.

This kind of interface may be satisfactory for technical experts, but the target user group for WP6 is teachers and learning designers. We expect them to have an interest in modelling, and to be willing to engage with complex designs, but we aspire to help them as much as possible in working with the metaphors of IMS LD, and do not expect them to have technical skills or understanding.

A much richer GUI framework can be provided by the Eclipse Graphical Editing Framework (GEF). According to the GEF project website:

The Graphical Editing Framework (GEF) allows developers to create a rich graphical editor from an existing application model. GEF consists of 2 plug-ins. The org.eclipse.draw2d plug-in provides a layout and rendering toolkit for displaying graphics. The developer can then take advantage of the many common operations provided in GEF and/or extend them for the specific domain. GEF employs an MVC (model-view-controller) architecture which enables simple changes to be applied to the model from the view.

In the creation of a graphical interface for Reload these features are highly desirable. Unfortunately, however, the GEF is widely recognised to be poorly documented and to have a steep learning curve. Thus a considerable investment of developer time was required to understand how the GEF could be used in the creation of an IMS LD editor. The outcomes were very positive, and have resulted in version 1.0 of the application, which utilizes 2-dimensional drawing objects made available in the GEF, such as those found in flowcharts, UML design tools and business process tools. This rich toolset means that we can now implement GUIs which are far more intuitive and visual than before (see Figure 2 TENCompetence ReCourse v1.0 graphical interface, developed using the GEF framework). Moreover this can now be done much more rapidly than in the current version of Reload, and so that rapid iterations of interface design and testing with fully functional applications will now be practicable. Figures 1 & 2 on the following page show screen shots of the original Reload application and the new ReCourse application developed by TENCompetence, and give some indication of the possibilities of the new application.
In order to achieve this improved interface a new code base was required. Fortunately not all the code from the current version of Reload needed to be thrown away.
Obviously, new GUI panels required new development time, however this process was eased by the existing work and by the GEF framework itself. Some important code could be re-used from the old LD editor, such as:

- Packaging into a Content Package
- General application framework
- Some editing panels
- XML parsing logic (JDOM)

Nevertheless, in order to implement the new paradigm it was necessary to develop a new framework based upon GEF. This was a straightforward and clear strategy due to the fact that

- The existing Reload Editor was developed using Eclipse
- A GEF interface for an LD Editor has already been developed

Thus the technical challenges presented by the creation of a graphical interface for the Reload LD Editor have now been overcome. The creation of an effective interface for the creation of Units of Learning has, however, been a major stumbling block in the past, with teachers finding it very hard to agree on an appropriate interface or framework of pedagogic components, or to specify how it should be built\(^2\).

In order to meet this challenge the LD Author development team has entered into a strategic alliance with a user group who are willing to work closely with the development team in developing an appropriate interface. This enables us to build to the practical requirements of the teachers, rather than asking them to provide an abstract analysis of their needs (or, worse, trying to imagine what they might be).

**D.3.2 A strategic alliance with an IMS LD user group**

The project team has succeeded in establishing a close working relationship with the IMS LD led by Mark Baxendale at Liverpool Hope University, an Associate Partner of the TENCompetence project. This team is one of the very few groups worldwide who are using IMS LD in producing and running courses with students in an authentic setting. They have also coordinated a number of past and present projects with teachers within and beyond Liverpool Hope, aimed at improving the effectiveness of IMS LD tools. They are therefore in an excellent position to provide the development team with the input they need for designs. The team in Liverpool Hope wrote an evaluation report on the Reload LD Editor, carried as part of the JISC funded LD4P project. In this report they identified interface issues which have been picked up in the development of ReCourse. Liverpool Hope also contributed to the present development effort some initial mock-ups showing their design ideas generated using GUI Design Studio and Toolbook. This collaboration was particularly timely in helping to make fast progress in the wake of changes in work package following

---

Achievements of work with the TENCompetence LD Authoring component may be summarised as follows:

1. There was substantial disruption to the development programme following a change of focus, and the withdrawal of the lead partner and technology provider from the work package. This has led to a delay of some seven months in delivery of this present DIP 1 deliverable, but the work package is on course with work corresponding to DIP 2.

2. A revised technical strategy has been defined, building on the achievements of the Reload LD Editor, the reference implementation of a Learning Design editor. The architecture of the application has been revised, in part to make it possible to use the Eclipse Graphical Editor Framework (GEF), and in part to redress shortcomings in the original Reload implementation.

3. The problem of identifying appropriate interface designs to implement and test has been resolved by establishing a close collaboration with a group of teachers associated with Liverpool Hope University who use IMS LD with learners on a regular basis.

4. Version 1.0 of the ReCourse Learning Design Editor has been developed, released and evaluated. This has the following features
   a. Open, edit and save IMS LD UOLs at level A. This provides the basic functionality equivalent to that used by Reload authors who were not technical experts. This is demonstrated by the fact that evaluation of ReCourse has been able to duplicate the tasks given to users in the evaluation of Reload in trials carried out independently of the project.
   b. Open IMS LD UOLs at levels A, B & C (but not represent the level B & C in this release)
   c. Graphical representation of IMS LD elements. On the one hand this makes the interface easier to understand and use (the user can make structures which make sense to them, and use drag and drop) and also has semantic value (e.g. an arrow between a role and an activity automatically creates a role part).
   d. The problem of exchanging graphical information associated with a Unit of Learning (UOL) is addressed by including this information as XML in the manifest of the UOL. Another application which opens the UOL can be written so that it interprets the graphical information which describes the representation of the UOL in the authoring environment. If this is not done then the graphical information will be ignored.
   e. Unlike other available LD Editors, ReCourse uses the IMS LD XML as the file format for the application’s internal working. This removes the import-export process which is required, for example, in Reload. This also ensures compliance with the specification.
f. A resource manager has been included, resolving a number of shortcomings in the original Reload application.

g. Facilities are provided within the ReCourse for setting up, populating and running Units of Learning. This functionality is not available in any other IMS LD Editor.

D.5 Evaluation

As mentioned above the collaboration with the team at Liverpool Hope has been extremely valuable, in particular with the evaluation of the ReCourse Learning Design Editor. The team evaluated the Reload Learning Design Editor in a series of workshops with practitioners in early 2007, with authentic users from a range of disciplines within HE and FE, including Computing, English, Psychology, Beauty Therapy, Education, Creative Writing and Science. Evaluation of the ReCourse tool repeated the same activities and instruments with a similar user group from Liverpool Hope and St Helens College, using the new editing environment. This has made it possible to make a clear comparison between the two applications. The results of this evaluation are included in Annex 1 to this deliverable, and summarised below:

1) The respondents experienced ReCourse as being easy to use. There were high mean scores for statements related to interface and use of the software. “Easy to use” and “easy to learn” were also high, but slightly lower. This may be interpreted as that the users The following are on a five point scale, with five indicating a positive result.
   a) interface:
      i) future use of the software unaided (4.3)
      ii) screen layout (4.0)
      iii) enjoyment in using the software (4.0)
      iv) the different sections of the software make sense (4.0)
   b) ease of use
      i) easy to use (3.7)
      ii) easy to learn (3.8)

2) The respondents were able to engage successfully with IMS LD using ReCourse. This is shown by very high mean scores for success in creating a UOL, and for expectation that in future the software could be used unaided. This does not demonstrate that ReCourse can be used by any teacher unaided in modelling pedagogy. It should be remembered that very clear tasks were given to respondents in the worksheets, and that in normal circumstances users would have the more demanding task of creating or adapting activities which they decided on themselves. Indeed it seems likely that the relatively low score for “knowing what to do” (when taken in combination with the high scores for usability and future use) is a function of respondents unfamiliarity with the IMS LD concepts and metaphors. The results do, however, indicate that ReCourse is suitable for use by non-technical users as the vehicle through which they develop skills in pedagogic modelling with IMS LD.
i) successfully create a UOL (4.00)
ii) future use unaided (4.3)
iii) know what to do (2.2)

3) **ReCourse is substantially easier to use than Reload.** A comparison of the results for ReCourse compared with those for the same evaluation activities for Reload indicate show improved results for all statements. The greatest improvement is in usability (as might be expected from the strong indicators for ease of use described in (1) above).

**D.6 API definitions for WP6**

According to the project plan API definitions are scheduled for all Work Packages at month 18. The relationship between the Personal Competence Manager (PCM) produced by WP3, and the work being done by WP6 is, however, rather different from the other aspect work packages. This is because WP6 leverages the services provided by CopperCore. Rather than developing a new set of services to be integrated with the PCM, the whole of CopperCore will be integrated with the PCM infrastructure.

This integration at the service layer means that a loosely coupled system is an appropriate solution. In the current release of the PCM the relation between the PCM and IMS-LD runtime is simply a link. Thus, for example, the user of a Unit of Learning sets 'completed' themselves, rather than this being set by an external LD runtime system. The result it has not been necessary to define APIs at this stage in the development process. So no provisioning, authentication or other data goes out, and no results come back.

In the next phase of work, however, it is intended to explore a tighter linkage between the PCM and WP6 (along the lines of the already achieved publication of UOLs from ReCourse to CopperCore). This, however, still does not require extensive changes to APIs, and is foreseen as follows:

- Authoring of Basic Units of Learning using a Web application which accesses and edits competence development plans on the PCM server. This work is currently underway. It does not, however, require changes to the API of the PCM, as it only makes use of existing services.

- It is intended to make it possible for the LD Authoring tool to access competence definitions within the PCM and to align these with learning objectives. Again, this does not require changes to the API of the PCM.

- It would be desirable to enable the PCM to provision and launch UOLs running in CopperCore Learning Design Engine (if it is possible to overcome the identity management issues which this will no doubt raise). This may require some minor changes to the PCM API, to enable it to handle values returned by CopperCore. Discussion with WP3 indicates that this is best addressed when the issue arises in WP6 development work.
Pedagogical Models

As discussed above, the work on Pedagogical Models in TENCompetence has been moved to task 2b of WP 2. This decision was taken because the pedagogic issues identified are related to the use of the whole TENCompetence system, not only that part of the infrastructure which uses IMS LD, IMS QTI and related specifications which is addressed in WP6. Work on that task is consequently reported in two project deliverables

1) The work to month 12 is reported in Milestone M6.1, "Templates of the pedagogical models to be used in authoring environment." This is available on the TENCompetence DSpace repository at http://hdl.handle.net/1820/876. Of particular relevance is section B, which:
   a) analyses the role of IMS LD in Lifelong Competence Development
   b) provides a model of the Competence Development Lifecycle, and relates this to the basic steps to be taken in IMS LD authoring
   c) provides exemplified pedagogical scenarios

2) The work carried out since month 12 on Pedagogic Models is reported in Deliverable D2.2 "Updated Use Case models and underlying vision documents and pedagogical model definitions," submitted in month 24. This defines the pedagogical elements that are needed in supporting life-long competence development, and to show their connections and functioning in context.

The results of this work inform the competence development activities of the project as a whole. They will also feed into the use made of the template functionality in the ReCourse editor, which will be extended in future releases.
Assessment Specification

Competence assessment is an important component of any competence development programme. Although classic forms of assessment (e.g., multiple-choice, filling-in-blank, and ordering) are still used in some situations, competence assessment is usually based upon more advanced forms of assessment (e.g., self- and peer assessment, 360 degree feedback, progress testing, and portfolio assessment). These innovative assessment methods address complex traits of learners.

One of the objectives of the TENCompetence project is to develop a formal assessment specification and a corresponding assessment authoring tool to support ordinary practitioners in designing and customizing assessments. These include new forms of assessment as well as the classic forms of assessment, which are covered by IMS QTI. In addition, in order to support both formative and summative assessment, the assessment authoring tool should integrate with the IMS LD authoring tool, so that an integrated learning and assessment process can be modelled as a specific unit of learning by using the integrated authoring tool. Such a specific Unit of Learning can be executable in integrated IMS LD and IMS QTI compatible run-time environments.

In accordance with the project plan, in the first 18 month period, we developed the first version of TENCompetence Assessment Specification and the first version of the proof-of-concept Assessment Authoring Tool based on the specification. We have also assessed the degree to which the classic and new forms of assessment represented in the classic and new forms defined using the first version of the TENCompetence Assessment Specification can be integrated with IMS Learning Design. This has not proved to be a simple matter, and we also describe the way in which this challenge is being addressed.

In this section of the deliverable we provide an overview of the work undertaken towards these goals, and the results obtained. A detailed description is provided in a detailed report included in Annex 2 to this deliverable. This documents the approach taken and the assessment specification which was developed and also includes sections on requirements and use cases.

D.1 An integrated competence development and competence assessment processes model

We developed a model integrating competence development and competence assessment processes. This model includes the following four stages:
- orientation
- evidence collection
- assessment by others
- performing competence development activities

The model is based on the similar existing models, but is specifically oriented to the needs of the TENCompetence project. The model addresses the roles of various forms of assessment in these four stages of competence development processes. This work has been described in a submitted chapter of a book entitled “The role of competence assessment in the different stages of competence development”. (http://dspace.ou.nl/handle/1820/871) and a conference paper has been included in the proceedings of the TENCompetence Open Workshop in Manchester, January 2007.
D.2 A technical approach to model and deliver classic and new forms of online competence assessment

We have developed a technical approach to model and deliver classic and new forms of online competence assessment through a combined use of IMS LD, IMS QTI, and assessment-specific services. Using these specifications we can model a competence assessment process as a set of units of assessment. Each unit of assessment may be regarded as a specialised unit of learning, which contains one or more assessment-specific components (e.g., IMS QTI documents and/or assessment-specific services). As a consequence, competence assessment, wrapped as a unit of assessment, can be delivered in any integrated IMS LD and IMS QTI compatible run-time environment.

It is important to note that the focus of this work is on modelling in a machine-interpretable manner. This approach has been tested and explained by describing the implementation of two key forms of innovative assessment.

The use of IMS LD and QTI means that the outcomes of this task will be able to leverage the existing implementations of these specifications. This provides the best prospects for practical interoperability and flexibility, and seamless integration with learning activities. The results achieved in this direction are described in four papers:

2. Using open technical e-learning standards and service-orientation to support new forms of e-assessment, in the proceedings of the TENCompetence Open Workshop in Manchester, January 2007 (http://dspace.ou.nl/handle/1820/931)

D.3 The first version of TENCompetence assessment specification

We have developed the first version of TENCompetence assessment specification. This specification is based on OUNL/CITO assessment model. The TENCompetence assessment specification is developed by simplifying OUNL/CITO assessment model. The reasons for simplifying OUNL/CITO assessment model for use in TENCompetence are:

- The OUNL/CITO model is extensive and complex, aiming for completeness in its coverage of all forms of assessment. This complexity has an impact on the usability of the model and a simplification could increase the adoption of its concepts.
- The TENCompetence Domain Model provides a larger framework into which the assessment model must be fitted. Some duplication of concepts is apparent in the two separate models, which can be removed through harmonization.
As a result, the TENCompetence assessment specification describes a conceptual model representing the main concepts and their relationships in a whole assessment process including the following stages:

- Assessment Design
- Item Construction
- Assessment construction
- Assessment Run
- Response Processing

It is important to note that the focus of this work is on modelling in a human-user understandable manner. It is expected that an assessment design modelled in a human-user understandable manner can be mapped to a corresponding assessment design modelled in a machine-interpretable manner.

**D.4 The proof-of-concept Assessment Authoring Tool**

It is necessary to demonstrate the feasibility of modelling both classic and new forms of assessment based on TENCompetence assessment specification, and of mapping the output of the tool into IMS LD and IMS QTI documents automatically. This is being done through the design and implementation a proof-of-concept tool. The tool consists of three components: Assessment Design Service, Assessment Construction Service, and Item Construction Service.

The Assessment Design Service is responsible for the creation and editing of Assessment Plans. The main object is the Assessment Plan. It is defined in terms of units of assessment and their assessment types, as specified from the assessment scenario, determining their sequence and time dependencies. The Assessment policy prescribes which assessment types (methods) can be used and on which conditions.

In context of the Assessment Model all participants in the process are collectively referred to as the Population, and their assessed competences or performance levels are referred to as Traits. Thus a Trait is an abstract object used to measure different personal characteristics. The identification of the most appropriate people to rate the performance of the individual is a key part of the process Ideally the recipient will have full involvement in identifying who they think is in the best position to comment on their performance.

It is also important to consider the provision of briefings for all participants on the objectives of the process and some basic tips for completing the questionnaire. The latter is referred to as the Assessment Policy. This may, for example highlight the importance of marking observed behaviour.

The Assessment Construction Service provides functions for constructing an assessment test, which consists of a set of items in a hierarchical structure. The Item Construction Service provides functions to define the individual items. These two services together are responsible for creation of a questionnaire, which is similar to a QTI test/item definition, but much simpler.
The proof-of-concept tool has been implemented, and Figure 3 above shows a screenshot of the tool. The tool can be used to edit simple form of assessment and to store/load assessment definitions in the forms of XML based on the TENCompetence Assessment model.

The full source code, examples, user-guide and final specification of the Assessment Model are available at:

Formative testing of the proof-of-concept tool showed that:
- the user can define certain types of assessment, such as 360 degree feedback
- the assessment definition can be viewed in the user interface
- the assessment definition can be saved as XML based on TENCompetence assessment model.

However, this internal testing also demonstrated that it was a hard task to transform an assessment definition of this kind into a corresponding executable model represented in IMS LD and IMS QTI. As its name suggests, the function of the proof of concept tool is to identify difficulties in the approach taken. Thus, while the difficulty identified resulted in a delay in the production of the production quality tool, it may be seen as a positive outcome, because it has provided information which can guide ongoing work in this task. There are two ways in which this problem of mapping the assessment definition onto IMS LD and QTI can be resolved, each with their advantages and disadvantages.
a) Further research into improving the mapping algorithms which are used to carry out the transformation. The advantage of this approach is that it will be possible to carry on using the simplified CITO/OUNL model which is currently proposed in the TENCompetence Assessment Specification. The disadvantage is that there is a risk that it will not prove possible to develop a satisfactory algorithm.

b) Revise and extend the TENCompetence Assessment Specification, using elements which are functionally similar to those used in IMS LD. The advantage of this approach is that we can be confident that a mapping between the TENCompetence Assessment Specification and IMS LD/QTI will be possible. The disadvantage is that it means including more elements in the specification, with a corresponding increase in complexity in the editing task (although this may be mitigated by appropriate tool design).

It was decided to follow both lines of work, for two reasons. Firstly, this reinforces risk management efforts within the work package. Secondly, the completed assessment tools based on the first version of the specification can provide comparison with those based on the second version which are currently under development. This will help to establish the degree to which it is possible to use a sophisticated mapping mechanism to reduce the complexity of the modelling task to be carried out by the author of the assessment (and, by extension, other pedagogic models). We discuss the revision and extension of the TENCompetence Assessment Model in the following section.

D.5 Redesign of the TENCompetence Assessment Model

Experience from the development of the proof of concept tool development has indicated that the combination of document-centric and activity-centric models has created much greater practical difficulties than had been foreseen. The first version of the TENCompetence Assessment Model, included in Annex 2, is based on the OUNL/CITO assessment model, which is document-centric. IMS LD, on the other hand, is an activity-centric model. The OUNL/CITO assessment model, like IMS QTI, specifies a data model for the representation of assessment item, assessment test, and results reports. It also contains additional concepts relevant at the assessment design stage, assessment run stage, and decision making stage. The TENCompetence assessment model is a simplified version of OUNL/CITO assessment model. However, while the removal of the decision making stage and several concepts related to other stages has indeed simplified the model, it remains a document-centric model. It is this which makes it difficult to map some concepts and their relations properly to those in IMS LD. We have therefore produced a strategy for production of a second version of the TENCompetence assessment model by modifying the OUNL/CITO assessment model from the perspective of process support. This is described in the Section F Roadmap for learning activities and units of learning.
E. IMS LD Run-time Component

E.1 Introduction

Tasks 1 and 2 described above deal with the authoring process. The outputs of this process are Units of Learning (which are expressed in IMS LD) and Units of Assessment (which are expressed as a combination of IMS LD and IMS QTI). For this approach to be effective there must be run-time systems which interpret these documents and coordinate the appropriate activities for learners. The Run-time Component task sets out to build on existing Open Source runtime engines to ensure that effective and usable systems are available to meet this need. Two principal challenges were identified:

a) There is no runtime environment which is compliant with the latest version of the IMS QTI specification, version 2.1.

b) There is a lack of flexible runtime services (e.g. a forum, or a chat, run external learning objects and tests, etc.) which can be launched from an LD player (typically a system based on the CopperCore LD Engine).

Both these issues have been addressed in this first phase of project work.

Firstly, the APIS runtime environment for IMS QTI which was developed with funding from the UK eFramework has been extended to support IMS QTI 2.1. This was a major undertaking as the structure of the specification has been radically changed in V. 2.1.

Secondly, the first version of the runtime system for integrating services with IMS LD has been tested and released. This builds on the most advanced currently available system for delivering services with IMS LD, and involves integrating services in the CCSI layer of CopperCore. The first service to be integrated was a SCORM player, making it possible for the two leading interoperability specifications for learning flows (IMS LD and SCORM) to work together.

In accordance with TENCompetence policy, all the source code for the IMS LD and SCORM integration is available at http://tencompetence.cvs.sourceforge.net/tencompetence/wp6/ and the APIS extension to cover IMS QTI 2.1 is at http://sourceforge.net/projects/newapis.

We now discuss this work in more detail, while a more exhaustive description is available in Annex 3.

E.2 IMS QTI runtime development

The Assessment Specification being developed by TENCompetence makes extensive use of IMS QTI, but no full runtime implementation of the most recent version of this specification, IMS QTI v2.1, is available. This was identified as an early priority for the project, and WP6 has taken steps to address this situation.
The IMS QTI specification enables the exchange of question and test items between authoring tools, item banks, test composition tools, learning systems, and assessment delivery systems to name a few. It describes a data model for the representation of questions and tests and their corresponding result reports that makes the interchange of information between systems possible. The model is expressed in XML. The basic elements of the QTI specification are as follows:

- **Item**: the question presented to the user along, presentation information, user answers processing mode, hints, and feedback.
- **Section**: a composite part of the assessment test or exam.
- **Test**: is an entire QTI instance divided into sections and subsections, and with a method for combining individual questions scores/marks

QTI v2.0, focused on simplifying the concept of individual question stored in the item QTI element, while aspects such as the grouping of questions in sections or exams were left out. QTI v2.0 introduced changes in interactions and new variables types, which, in turn, induced modifications in the processing of question responses. The formatting of content text was included in the specification and based in XHTML. QTI v2.0 also introduced new kinds of templates along with feedback, hints, and other additions. Another valuable addition is a method that determines how QTI can be integrated in Learning Design documents. This is done by coding assessment materials in QTI format to enable sharing the variables which hold the results in LD.

QTI v2.1, the most recent release, completes the update from version 1.x to 2.x by adding missing QTI elements such as those associated with the grouping of items in sections or tests, alongside a new results report.

To simplify the implementation of this wide and complex specification, we searched for a good implementation to upgrade. But, despite community enthusiasm, no reference implementation of QTI v2 had been done.

We chose to extend APIS the Assessment Provision through Interoperable Segments (APIS) engine (Barr and Sclater, 2006). This was originally created by Strathclyde University, providing a liberally licensed open source implementation of the non-controversial elements of the specification (IMS QTI, 2006). They planned to implement a modular item-rendering engine in line with the version 2.0 of the specification. This engine addresses the operations required by potential tools defined in the Open Knowledge Initiative (OKI, 2006) and IMS Web Services (IMS GWS, 2006).

Our work focused on improving the implementation of the APIS engine to make it compliant with the latest version of the specification (v2.1). Significant new functionality was provided, principally: test context instead of just items (questions), a wide range of new elements test-related, new and more complex response processing and new types of interactions.

The first step carried out was to add elements which were missing from the APIS QTI version 2.0 implementation. APIS did not implement text-based and graphic interactions, new data types and associated cardinalities, logical and other types of expressions, and the upgraded processing of answers and responses. These
functionalities enable the management of tests that require the writing of varying amounts of text or some graphical interaction.

The upgrade resulted in an engine with almost all interactions implemented, except for: the graphical, some of the miscellaneous and few of the text related interactions. All the data types, apart from files, were now included, together with cardinalities and most of the expressions. In addition, a more complex response processing was supported. Furthermore, the newly added data types enabled the formulation of questions whose answers could be composite, for example when test takers are required to select several answers from different pools as a response to a test problem.

The second step was to change the entire engine structure to convert it to QTI v2.1 As this version accepts not just questions but entire tests as well, we have introduced critical changes to the principal components of APIS including the main engine class. Processing of test units thus has become more hierarchical, where tests are considered both as units and groups of divisions. QTI v2.1 appends concrete characteristics for the test's divisions including time constraints, navigation flow style, and others. These paradigms and definitions were implemented and added to the improved engine. This required the adaptation of some of the previously existing processes and elements.

To demonstrate that the resulting engine works correctly, we based our implementation on a set of examples provided by IMS Global Learning Consortium (The IMS QTI specification creator) (IMS Global Learning Consortium, 2006). We adopted these examples as our benchmark for testing as they were published to address and illustrate the specification and tackle it with a wide range of approaches to cover several levels of complexity.

The actual state of the engine, referring to QTI version 2.1, is represented in Figure 4. The elements included in this version of the specification are represented by boxes. The code colour indicates what has been implemented and tested -green-, elements not completed or not started -orange- and that part represented by QTIv2 -yellow-.
Figure 4 QTIv21 scheme of elements

For more information about this implementation work, please see A QTI Management System Implemented for Service Oriented Architectures in Service Oriented Approaches and Lifelong Competence Development Infrastructures: Proceedings of the 2nd TENCompetence Open Workshop, reproduced in Annex 3 Section 4.

E.3 IMS LD – SCORM integration

E.3.1 Services and IMS LD

The motivation which informed the development of Educational Modelling Language (EML) by OUNL in the early years of this decade, and its successor IMS-LD, was to provide a framework which formally describes the usage of learning objects (Koper and Es 2003). As Koper states “The key principle in learning design is that it represents the learning activities and the support activities that are performed by different persons (learners, teachers) in the context of a Unit of Learning. These activities can refer to learning objects … and … services (e.g. forums, chats, wikis) that are used to collaborate and to communicate in the teaching/learning process”. underlines (Koper 2005). The infrastructure available for running Units of Learning (UOLs) provides an effective solution for running activities which refer to learning objects, but the implementation of activities in the context of a range of services remains problematic.
Three factors in the development of the IMS-LD specification contribute to this problem.

- IMS (and other) specifications have been developed to resolve specific interoperability problems, with no integrating framework.
- IMS-LD was developed by adapting EML. In this process a number of EML services were removed because they were already covered by other IMS specifications (e.g. assessment).
- only a small number of simple services were included in IMS-LD (send mail, conference, index search, monitor), in order to make implementation (and hence adoption) simpler.

**E.3.2 IMS LD Runtime systems**

These historical factors mean that it is extremely hard to integrate IMS-LD with the services such as forums, chats and wikis mentioned by Koper. This lack of services at runtime for IMS LD is a serious barrier to the use of the specification, both within TENCompetence and beyond.

Moreover, since the publication of the IMS-LD specification there has been a trend towards distributed service oriented architectures, which is manifested in eLearning in initiatives such as the eFramework (Olivier, Roberts et al. 2005), the Personal Learning Environment approach (Wilson, Liber et al. 2006). These tendencies are undermining the dominance of the monolithic Virtual Learning Environments which were assumed as a given when EML and IMS-LD were developed, and make the resolution of the services problem still more urgent. The TENCompetence project may be considered part of this trend.

Within IMS there is related work underway towards the integration of services. The IMS Tools Interoperability Specification (IMS Global Learning 2006) seeks to facilitate the integration of third party tools with core LMS platforms, and IMS Shareable State Persistence describes a way of storing and sharing state information in runtime systems (IMS Global Learning 2006). In many ways Business Process Execution Language for Web Services version 1.1 (BPEL) (IBM, BEA Systems et al. 2005) is also addressing similar problems in a different domain, and might be a source of future solutions. None of these initiatives, however, provides a usable solution to the specific problems of orchestrating services in IMS-LD.

TENCompetence is taking a dual approach to integrating services into IMS-LD. Desk research is being carried out on the requirements and possible approaches to developing a connector protocol, but in parallel practical implementations are being developed, not only to provide much needed functionality, but also to ensure that the connector protocol which is produced will be rooted in real examples of integrated services. We now discuss the implementation work completed in the first 12 months of the project and reported in M6.1, and will then move on to outline the proposed connector protocol which has been developed since then and the development work currently underway.
E.4 Integration of IMS LD and SCORM

As described by Olivier Chapter 2 in the Springer Learning Design book by Bill Olivier (Koper, R; Tattersall, C Eds., 2005) “Learning services are likely to come in two varieties: those … which are set up as part of a local environment; and those that are set up as remote web services”. These varieties are also termed closely and loosely linked scenarios. In the work reported here we have focused on the closely linked scenario, in order to avoid the additional complications of SSO or account replication (see the presentation by (Wilson 2005) for an outline of these and other relevant issues).

The particular service which was chosen for implementation was a SCORM 1.2 Sharable Content Object player. There are two principal reasons for this. Firstly, there are many SCORM objects available, and it would be valuable to learning designers to be able to use them in UOLs which can provide a much wider range of pedagogic contexts. Secondly, although IMS-LD redresses the pedagogic limitations of SCORM (see, for example (Hummel, Koper et al. 2005)) there is a mistaken perception that use of the two specifications together is not useful. We hope to correct this.

The analysis carried out in (Tattersall, Burgos et al.) provides a starting point for our implementation. This paper proposes that SCORM Sharable Content Objects (SCOs) could appear within the Environment section of an IMS-LD activity, and describes two approaches to implementation.

The first involves the SCO being physically located within a separate SCORM aware VLE. The Learning Object within the UOL simply references the web address of the SCO running on the remote VLE. The advantage is that the IMS-LD Runtime Environment does not have to know how to handle the SCOs runtime calls to the API, but the Learning Design Environment does not have access to the data model that the SCO interacted with, so there is little or no interaction between the SCO and the executing Learning Design that referenced it.

The second proposed method involves the SCO being physically part of the Learning Design package, and directly imported into the IMS-LD runtime environment. When the SCO needs to be launched within the IMS-LD Player, it is passed to a dispatcher. This acts as an interface to the Learning Design runtime environment and in this case, the SCORM runtime environment. The dispatcher acts as the SCORM aware LMS, provides the environment for the SCO to execute in, and has access to the data model which the SCO interacts with. Subsequently the changes in the SCOs data model can be used to update properties and conditions within the Learning Design, and so provide true interaction between the SCO and the rest of the Learning Design. This was considered to be the required functionality, and so this second method was adopted.
E.5 The software developed

Two existing open source systems were used, the CopperCore Learning Design Engine (CopperCore), and the Reload SCORM player (Reload). A dispatcher framework was available in the CopperCore Runtime Environment, the CopperCore Service Integration framework, (CCSI). Some related work had already been carried out to integrate CopperCore and IMS-QTI, using the dispatcher method to integrate the APIS QTI service. However, whereas IMS-QTI content essentially consists of an XML file which needs to be processed and rendered, a SCO presents a different set of problems. Typically a SCO will consist of a single HTML page with embedded javascript. This page needs to be able to access an APIAdapter object within the page (or frameset hierarchy) as it is being taken. It is the responsibility of the LMS to provide the APIAdapter and it allows the SCO to set and get values from a defined data model. The data model holds various values that would allow an LMS to track details on how the user interacted with the SCO. Without an APIAdapter, the SCO cannot set and get these values and so is really no different to ordinary web content.

For details of the implementation work carried out, please see Annex 3 Section 3.

In order to test the system a UOL was developed in which the learners interactions with a SCO determine the learning flow. The UOL contains six acts, and the SCO includes a test with five questions. At the start of the Unit of Learning the user is only able to access the first Learning Activity. From here the user has access to the SCO which is located within the Environment section of the Learning Activity. The user clicks on the link which loads the SCO and then s/he posts answers to the questions. Depending on which questions the user answered correctly, the structure of the Unit of Learning changes (although a browser refresh may be needed to reflect this). New learning activities then appear in the following acts which are designed to help the learner answer the questions previously answered incorrectly.

The process of designing, implementation and testing of the functionality, clarified the problems and complexity posed by connecting a new service to an IMSLD runtime system. The CopperCore Service Interface layer provides the necessary framework, but actually implementing a new service requires knowledge of this specific API, and not a defined and agreed standard. The framework also is quite open, allowing the developer to write his/her own calls between LD Engine and new service. While this allows a large amount of freedom to the developer, it is also very abstract. The functionality provided by the runtime system developed is valuable, and it meets the need to integrate two of the most significant specifications used to represent learning activities. The effort involved, however, suggests that it is not a solution which can meet the need for the agile integration of a large number of services into IMS LD, which is required for the TENCompetence Communication Protocol. This has led to the development of a new line of research and development, based on the use of multi-user widgets, which we will now outline.
E.6 Connection Protocol

This deliverable contains the first draft of the Connection Protocol, which is the focus of current implementation work in Task 3. In this section of the deliverable we provide only an overview, and more detail is available in Annex 3 Section 5. The current document describes the technical requirements, non-functional requirements, and the widget service demonstrator which will be built in the coming period. This will be adapted and extended in the light of practical experience before proposed as a public specification. In developing the draft protocol the following approaches were considered as providing a potential basis for the solution architecture:

1. The IMS Tools Interoperability Guidelines, v1.0
2. The IMS Learning Tools Interoperability Specification, v 2.0 (under development)
3. The LAMS Tool Contract
4. The W3C Widget Protocol, v1.0 (under development)

Initial exploration and experiments indicated that:

- The IMS Tools Interoperability Guidelines specify an architecture that is much more like a conventional LMS setup, and offered no capabilities for monitoring, or shared state. It also requires considerable effort on the part of tool authors to implement a range of SOAP Web Services.

- The LAMS tool contract is a mature technology, but many of the underlying assumptions of the API are dependent on the LAMS system, particularly the tight coupling of user information within a single application server, and would be more difficult to make generic, component based solution.

- The general approach of the W3C Widget Protocol is the most appropriate for TENCompetence. In particular it supports a much simpler and more lightweight set of requirements for collaboration services to meet, and enables existing widgets developed for the Apple, Microsoft, and Yahoo platforms to be fairly easily ported to operate within an LD environment.

Accordingly it was decided to create a draft Connection Protocol building on the W3C Widget Protocol, and to create a prototype implementation based on this, which is currently under development. This is further described in the next section, Roadmap for learning activities and units of learning.
### F. Roadmap for learning activities & units of learning

A solid start has been made on providing an open source and standards based infrastructure for learning activities and units of learning. Valuable infrastructure been created which will be used in the coming period of project activity, such as the integration of SCORM and IMS LD, the provision of up-to-date QTI runtime environment, significant advances in IMS LD authoring. The first version of the Connector Protocol has been defined, and the Assessment Specification has been completed and used to create proof of concept tools.

This work has also provided a number of new insights, which will inform future work, as we describe in this section.

#### F.1 Authoring learning activities and units of learning

The development of ReCourse v.1.0 provides a dramatically improved tool for authoring IMS LD with a graphical user interface, but substantially more can be achieved.

Firstly, in order to be truly usable it is not enough to create an effective modelling environment compliant with interoperability specifications. Teachers need to use Learning Design in their daily work, and at the moment users are faced with many technical issues in publishing and provisioning Units of Learning at runtime. A principal thrust of work in the coming period will be to investigate and implement a means of publishing transparently from the authoring environment, so that teachers will only be required to provide details of the learners who are to take part.

Secondly there is a need for further integration with other essential parts of the Learning Design infrastructure. These include:

- An editor for questionnaires and tests, which will generate IMS QTI 2.0. This will resolve one of the longstanding problems for IMS LD authors, as it has always been difficult to include such tests in a UOL, because of the need to create matching properties in both LD and QTI. The new component will make this easy to achieve.

- Integration with a Learning Design repository. ReCourse will be integrated with a repository, probably the opendocument.net open source repository, which provides information about the UOLs which it stores, and which makes it possible to retrieve resources from inside stored UOLs.

- Integrated editing for runtime services provided by the TENCompetence Widget Server, which is under development in the WP6 Runtime task. This will enable users to define chats, forums and other services easily from within the ReCourse tool.

It is planned to provide these features in the second release of the ReCourse LD Editor, in D6.2, scheduled for May 2008.

Beyond this date the focus will be on two areas.
Firstly, IMS Learning Design at level B provides what is effectively a programming language, with the capability to set properties and conditions which affect the learning flow. At the moment this work has to be carried out by technical experts using the Reload application. Research is needed to design and implement a workflow editor interface for ReCourse which extend the ease of use achieved for Level A to this more complex domain.

Secondly, in order to fully implement the IMS LD specification ReCourse will have to support the definition of metadata throughout the components of a Unit of Learning. This task is made more complex by the fact that there are a number of metadata specifications which could be used, and more may appear in future. The team has investigated the Eclipse Modelling Framework (EMF) and intends to research the potential of this technology as an innovative solution to this implementation challenge. EMF would enable the application to leverage a domain model which reflects IMS specifications, and map this against the GUI model. The EMF can take care of reading, writing and modelling XML Schemata and XML document instances, making unnecessary the development of a hand-crafted domain model, such as that used for the IMS LD specification in ReCourse. This greatly facilitates implementation of new and changed specifications, as may well be required for metadata support. The IMS LD specification, on the other hand, has remained stable since its release in 2003, and there are no current plans to change it. In view of this the advantages of a hand-crafted model outweigh those of a domain modelling approach for the IMS LD specification which is at the core of the application.

Thirdly, the target users of ReCourse are teachers and learning designers who are not technical experts but who are willing to spend some time understanding the powerful modelling concepts of IMS LD. This is a powerful language, and while its use does not require technical knowledge, it is by nature complex. Many teachers do not have time, or are not motivated, to engage with IMS LD in this way. In part this can be addressed by improved support for templates, which is on the development roadmap for ReCourse. However there will be many teachers who are not able or willing to engage with an application such as ReCourse, and who need a much simpler application. With these users in mind, the next phase of work will research and implement a simple “pre-authoring” Web application which can sequence learning resources and activities. This system will also explore the potential of the TENCompetence Personal Competence Manager as a provider of services which can support such an authoring system, and will produce Basic Competence Development Plans, as defined in the section on IMS LD authoring above. It is foreseen that the new Web tool will generate IMS LD Level A, and so integrate with the ReCourse IMS LD Editor, enabling the Basic Plans to be elaborated further.

**F.2 Assessment**

In view of the difficulties experienced in representing the assessment model using the activity centric structures of IMS LD the team has developed an alternative strategy in parallel with the completion of the proof of concept tools.
The observations which have guided the new approach to the OUNL/CITO are:
1) an activity-centric assessment process model can be integrated with IMS LD model
2) an authoring tool based on this assessment process model can be integrated with
the IMS LD authoring tool being developed in this work package
3) an assessment process definition created by using our target assessment process
authoring tool can be transformed into an executable model represented in IMS LD
and IMS QTI.

The principal ways in which the OUNL/CITO assessment model needs to be adapted
in order to comply with this new approach may be summarized as follows:

1) Explicitly use the concept of *stage*. This is used only as an underlying concept in
the model to organize related concepts and constitute a process model.

2) Introduce the concept of a *role* and reorganize related concepts such as *candidate*,
*group*, and *person*

3) Introduce the concept of a document representing all artefacts produced and used
in assessment including *assessment test* and *assessment item*, which will be
represented in IMS QTI directly

4) Introduce the concept of a *tool* to represent all applications/services used in
assessment including assessment test/item authoring tool and response tool

5) Introduce the concept of an *activity*, which is treated as the central concept in the
model in a way that other concepts such as *role*, *document*, *tool*, and *stage* are
connected to it. Thus, the central position of assessment test/item in the model is
replaced by activity

6) Remove some concepts related to the run-time stage (e.g., *population*, *assessment-
take*, and *assessment session*) and some of the abstract concepts (like *construction
item* and *assessment function*

7) Introduce more assessment-specific and practitioner-familiar concepts (e.g.,
*portfolio*, *concept-mapping tool*, *responding activity*, *assessor*, and so on). This is
treated as an open set in order to explicitly support a wide variety of forms of
assessment

Using this approach the team will develop an alternative version of TENCompetence
assessment specification in the period starting month 19 of the project. In order to
distinguish this model from the simplified OUNL/CITO model, this model will be
called the TENCompetence Assessment Process Model. This work will be reported on
in deliverable D6.2.

In order to demonstrate the Assessment Process Model units of assessment will be
developed and expressed as a combination of IMS LD and QTI. Authoring support
will be provided within the ReCourse editor, in order to maximise integration of the
final system.

In order to support this strategy it is proposed that

a) a QTI editor will be integrated into the ReCourse editor

b) templates will be created which will facilitate authoring of units of assessment
It is planned that the QTI editor will be directly integrated with the Personal Competence Manager (PCM), which will enable standards based tests and questionnaires to be created within the PCM client. More complex units of assessment produced with the TENCompetence Assessment Model will leverage the integrated runtime environment being developed by WP 6.

F.3 Runtime

As described above, the established approach to integration of services with IMS LD was used in the integration of SCORM and IMS LD. However, this proved to be too time consuming, and insufficiently flexible and generalisable. In response to this the first draft of the TENCompetence Connection Protocol has been developed which defines an innovative new approach to IMS LD services. This will provide the main theme for research and development into runtime systems in the coming period. The elements of a system using the TENCompetence Connection Protocol may be described as follows:

A Widget is a self-contained application for displaying and updating remote data that can be deployed as part of a broader platform. Widgets may operate singly or may act collaboratively using shared states. Widgets are deployed on, and instantiated by, a Widget Server. Each Widget Instance can communicate with the Widget Server through the Widget API. (The term "widget" is introduced in this document to replace the ambiguous term "service" used in Learning Design). The following components are required.

The Widget Server is the platform which provides a Widget Service, and deploys one or more Widget Instances and their corresponding Widget API, typically through serving an iFrame containing a page with JavaScript references to the Widget JavaScript and the Widget API JavaScript. The Widget Server can offer a range of Widgets that have been imported and registered using standard Widget Packages (such as the W3C, Apple, Windows and Yahoo Widget Package formats).

The Widget Service is the service provided by the Widget Server that responds to calls made to the Widget API and instantiates and manages widgets in response to requests from external applications, such as CCSI.

The Widget API is the local JavaScript API accessible to each Widget Instance that enables it to communicate with the Widget Server.

A Widget Instance is a Widget instantiated within a particular user's browser.

A Widget Proxy Service is a service that supports Widgets to make calls to remote services, avoiding the Same Origin Policy of the user's browser.

A preliminary architecture for this system has been developed (see below) and this will form the basis for the implementation work to be carried out. We also note that if this work proves successful then there will be implications for the design of the LD Authoring applications, which will need to be able to define the services to be used. This will be assessed when the Widget Server has been tested and its use evaluated.
F.4 Release schedule

The next release of the WP6 infrastructure will be in month 30 of the project. In this release the focus will be on integration of the components developed for authoring, assessment, and runtime to produce a system which makes it substantially easier for teachers to create and publish units of learning. In the following year’s work this system will be further developed in response to evaluation work, and with the planned inclusion of extended functionality.

The components for the core and extended integrated functionalities are shown in the tables on the following page.
## Components of the WP 6 integrated core system, scheduled month 30

<table>
<thead>
<tr>
<th>1. Pre-authoring</th>
<th>Exemplars of Units of Assessment, written as UOL/QTI combinations and commented in the LD authoring tool. Pre-authoring does not form part of core functionality, but a first version of Pre-authoring component, covering Create Course and Run Course will be delivered in month 30.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Authoring</td>
<td>Full read / write / import of IMS LD Level A QTI authoring plug-in with, (1) multiple choice (2) multiple response (3) fill in the blank (4) Lickert scale. Creates QTI 2.1 Integration of QTI plug-in properties with LD authoring Button to publish UOL, with interface to add users Integration with repository Inclusion of Widget services in UOLs, selected from list made available to the author by the Widget server.</td>
</tr>
<tr>
<td>3. Runtime</td>
<td>Integration of extended APIS into SLeD Automatically set up and populate UOL with eMail addresses provided by LD authoring tool Widget server integrated into SLeD, with at least chat and messaging services available Maintenance and bug fixing of SLeD server</td>
</tr>
</tbody>
</table>

## Components of the WP 6 integrated extended system, scheduled month 42

<table>
<thead>
<tr>
<th>1. Pre-authoring</th>
<th>Development of a pre-authoring tool for Units of Assessment, in accordance with the TENCompetence assessment specification. Further development and extension of the pre-authoring component for Units of Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Authoring</td>
<td>Full support for IMS LD levels B &amp; C Integration of UOL objectives with competence development objectives in the PCM Extended support for setting up and populating UOLs from within the authoring tool and PCM</td>
</tr>
<tr>
<td>3. Runtime</td>
<td>Development of additional widget services and a production quality server Extended integration with Learn eXact Upgrading of CopperCore and APIS servers are required</td>
</tr>
</tbody>
</table>
G. Links to the principal outcomes of WP6

The following links provide direct access to the principal software and publications produced by WP6 in the first 18 months of the project.

- Authoring
  - Links to the code and user guide for the ReCourse Learning Design Authoring tool v.1.0: http://www.tencompetence.org/ldauthor/
  - Paper at the TENCompetence Open Workshop In Barcelona “Development and evaluation of the Reload Learning Design Editor” http://hdl.handle.net/1820/1135
  - A paper delivered to IEEE 2006 “Adaptive Learning Objects Sequencing for Competence-Based Learning” http://hdl.handle.net/1820/682

- Assessment:
  - TENCompetence Assessment Specification, v. 1.0. http://hdl.handle.net/1820/1107
  - TENCompetence Assessment Model (TAM), v.1.0 http://hdl.handle.net/1820/1108
  - Portfolio assessment player tool, v. 1.0 http://hdl.handle.net/1820/1109
  - 360 degree feedback assessment tool, v.1.01 http://hdl.handle.net/1820/1111
  - User Guides to Assessment tools http://hdl.handle.net/1820/1110
  - A book chapter “The role of competence assessment in the different stages of competence development” http://dspace.ou.nl/handle/1820/871
  - Conference paper “An Efficient and Flexible Technical Approach to Develop and Deliver Online Peer Assessment”. In proceedings of CSCL conference, July 2007 http://dspace.ou.nl/handle/1820/853
  - Conference paper “Using open technical e-learning standards and service-orientation to support new forms of e-assessment” http://dspace.ou.nl/handle/1820/931
• Runtime
  o IMS LD
    ▪ IMS LD and SCORM runtime integration source code
    ▪ Documentation for the IMS LD and SCORM runtime integration
      http://www.tencompetence.org/ldruntime/
  o Conference paper “Integrating IMS Learning Design and ADL SCORM using CopperCore Service Integration” http://hdl.handle.net/1820/1023
  o The draft TENCompetence connection protocol included in Appendix 3 of this deliverable
  o Conference paper “Extending IMS Learning Design services using Widgets: Initial findings and proposed architecture” http://hdl.handle.net/1820/963.
References


Garrett, Jesse James (2005) "Ajax: A New Approach to Web Applications"


