

Developing an approach for Learning Design Players

Patrick McAndrew, Rob Nadolski,
Alex Little

Abstract:

The EduBox system as described in Tattersall, Vogten and Hermans (2005) was developed at the Open University of the Netherlands in order first to support the use of EML as a formal way to describe its courses, and then redeveloped to support later versions of EML that evolved into Learning Design. Edubox is located as a solution for the OUNL and this allows it to offer a viable platform for student use and to use particular implementations for aspects that are not fully determined in the Learning Design specification. Examples of these are the ways to use question and test and the format for content. Edubox is built around EML 1.1 which is very similar to Learning Design but not identical. This means that EduBox is not designed as the basis for players outside the OUNL. This paper looks at developments on players that can be used more widely. It focuses on the use of the CopperCore Learning Design Engine [1] as a basis for how a player can work and two related implementations that use that engine: the Copper Core LD player and the SLeD player [2]. In all cases the intention is to produce open source software for free reuse and to assist others working in the same area.

Keywords: learning design; web services; java; open source.

Interactive Demonstration: A demonstrator service for the SLeD system is under development and can be accessed at <http://sled.open.ac.uk/>.

Commentaries:

All JIME articles are published with links to a commentaries area, which includes part of the article's original review debate. Readers are invited to make use of this resource, and to add their own commentaries. The authors, reviewers, and anyone else who has 'subscribed' to this article via the website will receive e-mail copies of your postings.

1 CopperCore as a basis for Learning Design Players

CopperCore is provided as an open source reference implementation of a Learning Design Engine (see Chapter 6 of the book) but in addition, starting from its first release it provided as an example a LD compliant player based on the Engine. This was intended to give an important source of background information for implementing other LD compliant player rather than for adoption. In practice it has been seen as a Reference Player and an important test-bed for running designs and validating other LD tools such as editors and authoring systems.

1.1 CopperCore Reference LD Player

The Roadmap for the development of CopperCore consisted of successively implementing the LD levels (A, B and C) for the CopperCore LD Engine. In parallel to this the development of the CopperCore LD Player took place to illustrate each facility developed in the engine. The CopperCore LD player was designed to meet the following needs:

1. a good test-bed for the developers of the CopperCore Engine enabling them to quickly check if the engine is performing properly
2. a way to prove the API design allowed a thin client to be built on top of the engine with the minimum amount of additional server side code. Most player code should consist of simple XML transformations.
3. a good demonstrator for using the engine. This implies that LD constructs/issues that are not covered by the engine are not part of the player. Typical example is the integration with other specifications and the connection with services. These aspects were left for other software and other engines.
4. web technology/XML based allowing the player to be used on all platforms on which the engine can be deployed,
5. able to play all LD constructs during runtime of a certain run,
6. able to show all LD constructs during runtime of a certain run, so that the constructs are apparent to the users,
7. able to show the runtime behaviour for all users assigned to a certain run, and for their roles, one being the active role which has to be specified in advance,

8. able to show the LD content (mainly (X)HTML) in a legible format,
9. pedagogically neutral,
10. able to meet as much as possible conformance to the common GUI guidelines as were also used for the development of the Edubox player (Tabbers, Kester, Hummel, & Nadolski, 2004), although this might conflict with points (6) and (7).

In this list it is important to note that there are no performance requirements or scalability criteria, and there are also no direct user requirements. The player is intended to show how LD works rather than support learners using LD.

Remaining issues, to be resolved in the future, were:

- a) For the CopperCore LD player and for the CopperCore Engine: How to deal with the integration of services? This applies both during design time and during runtime. For example: Ideally a learning design should take into account the way a student performs in a test (using IMS Question and Test Interoperability, QTI[3]), but there is no clear infrastructure that can decide when and what properties are exchanged. This hinders a clear cut communication between the CopperCore LD Engine and an existing QTI-engine in both directions. Moreover, the CopperCore LD Player itself is not able to render QTI content in a legible format, so currently a QTI-player is needed in addition to the LD player. As a result users are very likely to be confronted with a different GUI for both players which could result in confusion among users.
- b) Better integration of management of units of learning. This is used when deploying runs, assigning users to roles etc. A key question is whether this is within the educational process or if it should this be considered an administrative task. If the latter is true, the player and the administrative module could be seen as separate entities that may be integrated through a learning management system. However if the creation of roles and role assignment is part of the educational process, the tooling should be part of the player. The LD specification does not favour the one view over the other and the CopperCore Engine is neutral in this respect. While the API have been divided into two blocks, those with an administrative nature and one with a more delivery oriented nature, nothing prevents a player combining these aspects.
- c) For the CopperCore LD Player: Ways to make it more responsive, to provide a more intuitive user interface and to improve security.

1.2 Initial release of CopperCore LD Player

The initial release of CopperCore included the level A CopperCore LD Engine and the level A CopperCore LD Player.

In figure 1 there is a screenshot of the initial CopperCore LD Player (level A), playing the unit of learning (UOL) 'IMS Learning Design Level 0' (Burgos et al., 2004). In this example, the selected role is 'learner' and the 'activity description' for the activity 'Introduction' is shown in the right plane. Notice that in the top left plane titles such as 'IMS Learning Design Level 0', 'Act 0' might be disturbing for a real learner, but showing such LD constructs was part of the design criteria (criteria 6) as was the overall layout adapted from Edubox (criteria 10).

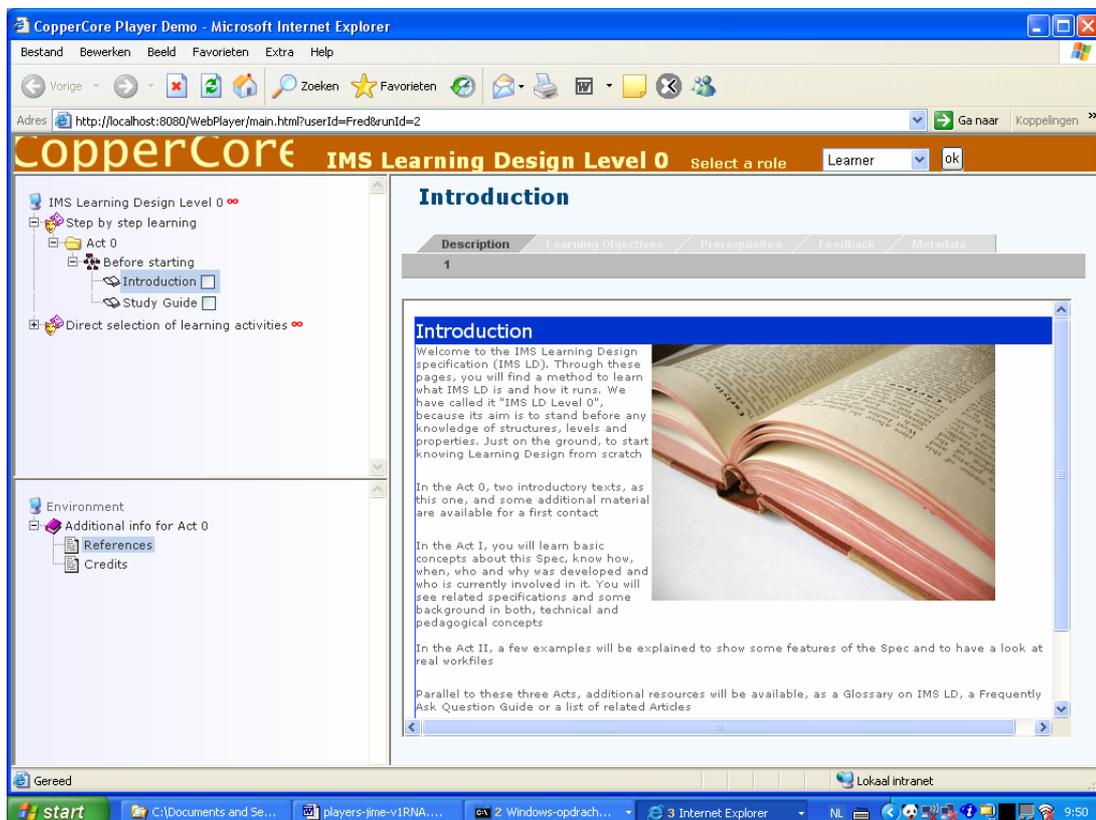


Figure 1: the initial release of the level A CopperCore LD player

2 Enhancing LD players

The JISC funded project Service-based Learning Design system (SLeD) had as its aims to offer an improved player system for learning design by separating out the player functionality from the underlying engine. This was already partly achieved within the existing CopperCore Engine but the new work would enhance the ways in which the CopperCore Engine could be used by adopting a web services approach wherever it was appropriate; thus communication between the player and engine used web services and additional end user tools such as search and conference systems were coded to allow web services. This approach proved valuable in allowing others to take advantage of the changes to CopperCore, for example the Reload system [4] is now able to prepare a 'dummy run' which can be validated and tested using the same communication routes with CopperCore.

This work was integrated with the OUNL roadmap for enhancing CopperCore capabilities to support the higher levels of LD. Level A support means that a complete planned design can be presented provided its structure can be pre-determined for each role within it. Supporting level B allows designs to include properties and conditions that determine progress in a more dynamic way; while it may be possible to rework many existing learner tasks into level A many existing designs of educational activity assume some of the capabilities needed for level B. Level C adds a notification service needed in more distributed implementations. These changes led to release of an updated reference player to demonstrate the new features (criteria 3 and 6) and release of a separately developed player SLeD to demonstrate new ways to communicate with the engine and provide a path for less constrained development of the player system.

2.1 Updated release

The updated release of CopperCore that resulted from carrying out the SLED project included levels B and C for both the CopperCore LD Engine and CopperCore LD Player.

Figure 2 shows a screenshot of the CopperCore LD Player playing the UOL 'Geo-Quiz-2' (Burgos, 2005). In this level B example, the selected role is 'Student', and for the selected activity 'Questions set' the Questions set is shown in the right pane. In this example, the Questions are modelled within LD (instead of QTI).

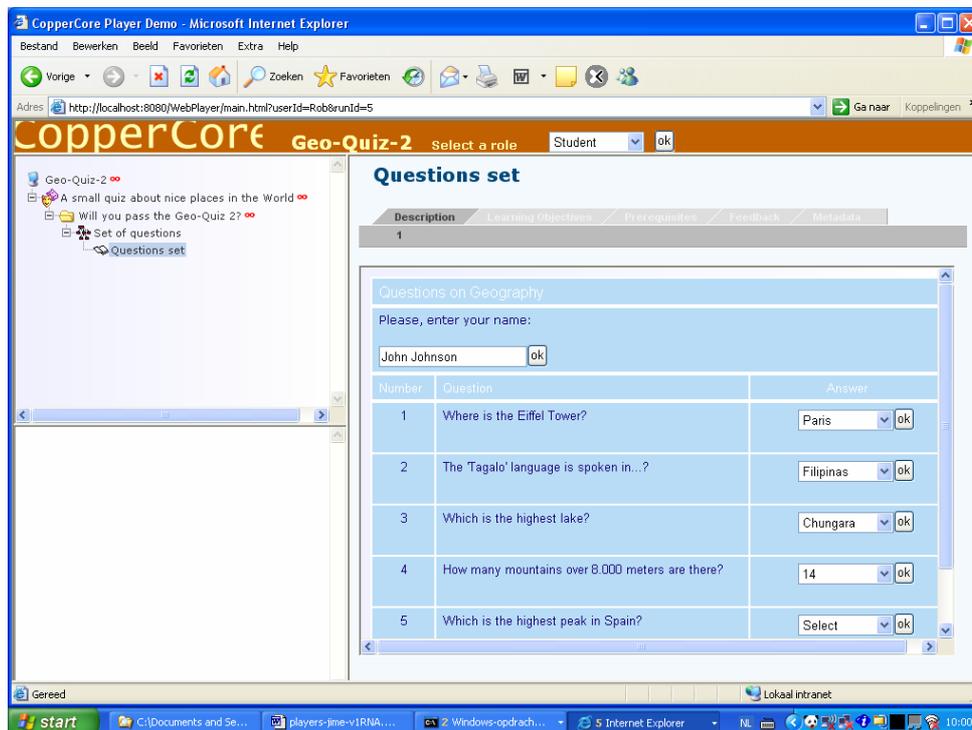


Figure 2: A quiz showing level B features of the reference player

In figure 3 the view of the player after 'completion' of the activity 'Questions set' in the previous example shows a new activity 'Results and feedback' and the presentation now shows the processing of the properties returned by answering the quiz. This example demonstrates how the reference player can support the new level B capabilities within the engine. It also illustrates how the lack of integration of LD and QTI can be a drawback at the user level, since a separate approach is needed to handle the 'Results and feedback' for the activity 'Questions set'. While alternative constructs could be used on a case-by-case basis to avoid this being apparent to the end user, it is preferable that such aspects are handled by QTI players developed for this purpose.

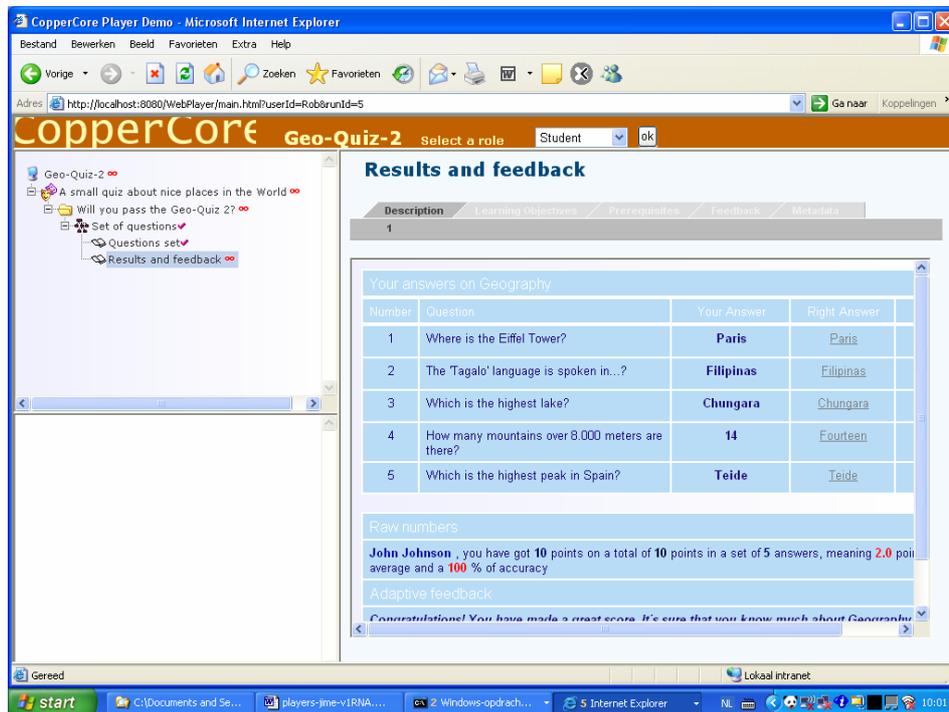


Figure 3: Results and feedback given on processing of quiz answers

2.2 The SLeD architecture and system

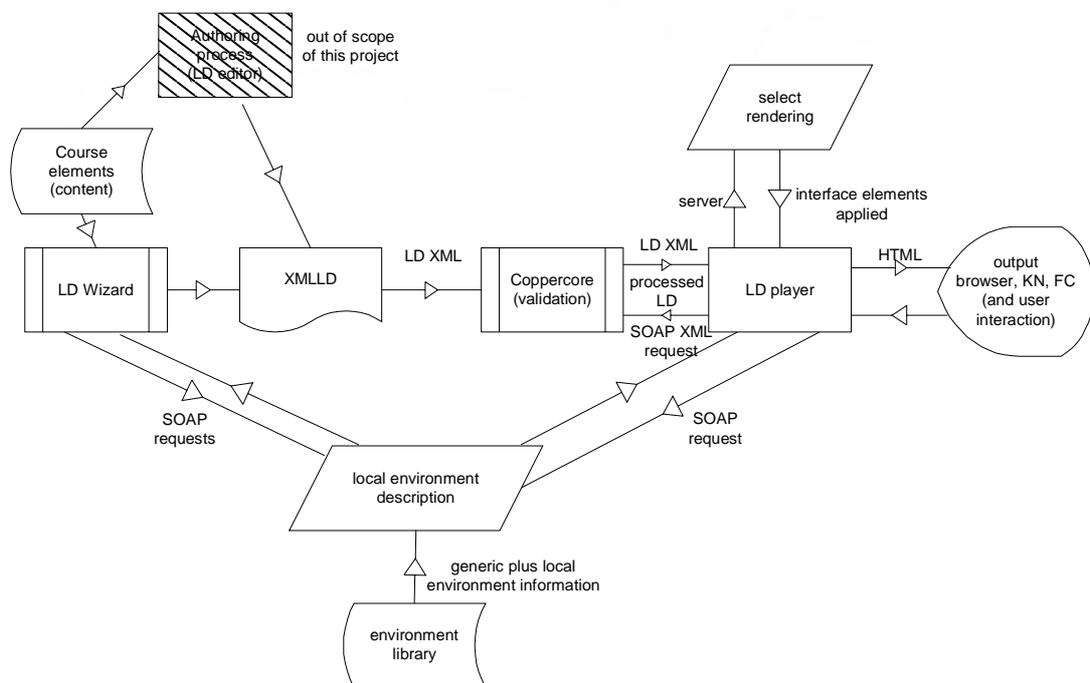
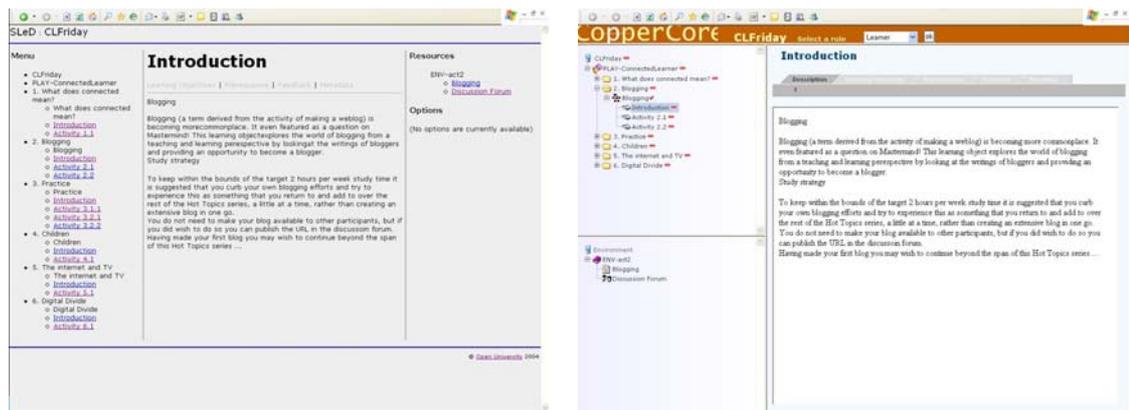


Figure 4: The architecture for the SLeD player and overall system design

Figure 4 shows the adopted architecture for the overall system design in the Service-based Learning Design project. Starting at the top-left of this figure it can be seen that authoring of the LD compliant UOLs was out of scope for the project. Work on editors has been apparent across a range of projects, notably RELOAD, but also CopperAuthor, ASK-LDT, Cosmos, MOTplus, and others (for an updated listing of LD tools see the UNFOLD site [5]). Within scope was work on a simplified wizard to allow variants on existing UOLs to be produced, the changes to CopperCore to allow service-based communication and enhanced features, end-user tools as services, ways to adjust the appearance of the player, and the player itself. The wizard system was implemented using a simple forms interface to allow transformations of the XML files that describe each UOL; this was a valuable way to extend the range of examples but is separate from the engine and player function so will not be considered further. The player was developed alongside changes in the underlying engine to support level B/C and to communicate via web services. End-user tools were implemented for carrying out searches and for conferencing. The design of each of these tools used the web service approach and a small amount of coding to the player to

communicate back with the service and present the results to the user. This version of the player was released as an alternative to the CopperCore LD Player. As the focus for SLeD was to show the viability of the architecture and a new development path in this release the apparent difference between the two players in minimal, see figure 5 which shows the same unit of learning in each player. However, a key difference emerges when accessing a discussion forum. The CopperCore LD Player can only indicate that the design requires a forum while the SLeD player is able to instantiate the forum and run it through the conferencing service (Figure 6).



a) SLeD view

b) CopperCore LD Player view

Figure 5: The same unit of learning in default view in SLeD and CopperCore LD Player



Figure 6: The forum service available from the SLeD player.

New work on the player under SLeD-2 is now taking advantage of the architecture to focus on three areas that relate to the three issues identified from the initial release of the CopperCore LD Player:

- a) developing translator packages to incorporate new tools, for example investigating the use of services provided by the Moodle [6] learning environment;
- b) a web-based administrator interface;
- c) and, new designs for the user interface.

2.3 Integration with Reload

The CopperCore LD player can be used in the test environment as part of a reference implementation for a learning management system. One example is the use of the CopperCore LD Engine and CopperCore LD player in conjunction with the Reload LD Editor. This integration with Reload does however limit the flexibility of the CopperCore Coursemanager, as in the Reload integration one can only look at the run time behaviour for one (dummy) user in one (dummy) run, although there is scope to place that user in different roles. CopperCore offers more flexibility through the use of its Command Line Interface to CopperCore (CLICC) as a way to create users, runs and assigning users to one or more runs and accessing the UOL in the context of a role (the active role). This enables testing of quite complex LD, in which the state of one user might influence the states of other users. Such complex testing is beyond the scope of the current Reload integration. However, for simple Learning Designs the Reload integration is more intuitive and faster to use.

3 Conclusion

The reference player released together with the CopperCore Engine has proved valuable in enabling people to understand and start to use Learning Design. This has been accompanied by other work on authoring and editing tools. The results is that the community of users, particularly supported by the UNFOLD project [7], has now developed a collection of example Learning Designs, all of which can be run in the CopperCore LD player. As originally released, however, the player was limited in its design aims to be a demonstrator and while new work on the engine and player have allowed enhancement to support new features this remains the aim. The SLeD player has now taken a revised approach that is less restricted. Further work will now focus on the SLeD player to show that it can integrate with a larger range of toolkits and provide a more flexible way to demonstrate and develop Learning Designs. This work is now underway at the Open University of the Netherland and The Open University in the UK that will first link Learning Design with QTI and then extend the ability to link to a generic set of tools.

Acknowledgements: The SLeD development, SLeD demonstrator and SLeD-2 toolkit projects are funded by the UK JISC under the e-Learning Framework programme.

4 References

- Burgos, D. (2005). *Geo-Quiz 2*. Retrieved July 5, 2005, from <http://dspace.learningnetworks.org/handle/1820/373>
- Burgos, D, Berbegal, N., Griffiths, D., Tattersall, C., Spoelstra, H., Kew, C., Vogten, H., Nadolski, R.J., Martens, H., O'Neill, O., & Koper, E.J.R. (2004) *IMS Learning design Level 0*. Retrieved July 5, 2005, (registration required) from http://moodle.learningnetworks.org/file.php/20/Examples_OUNL/IMSLD_Level0_HTML/index.html
- Tabbers, H.K., Kester, L., Hummel, H.G.K, & Nadolski, R.J. (2004). Interface design for digital courses. In W. Jochems, J.J.G. van Merriënboer, & E.J.R. Koper (Eds.). *Integrated E-learning: Implications for pedagogy, technology and organization* (pp.100- 111). New York: RoutledgeFalmer.
- Tattersall, C., Vogten, H., & Hermans, H. (2005). The Edubox Learning Design player. In: Koper, R and Tattersall, C, (Eds.) *Learning Design, A Handbook on Modelling and Delivering Networked Education and Training*. pp 303-310. Berlin: Springer.

5 Footnotes

- [1] <http://coppercore.org>
- [2] <http://ldplayer.sourceforge.net>
- [3] <http://www.imsglobal.org/question/>
- [4] <http://www.reload.ac.uk/>
- [5] http://www.unfold-project.net/general_resources_folder/tools/currenttools
- [6] <http://moodle.org/>
- [7] <http://www.unfold-project.net:8085/UNFOLD>