

Shortcomings of Learning Design Approaches And a Possible Way Out

Felix Mödritscher, Fridolin Wild
Institute for Information Systems and New Media
Vienna University of Economics and Business Administration,
Austria
{felix.moedritscher|fridolin.wild}@wu-wien.ac.at

Abstract: Shifting away from traditional instructional design to younger research streams like personalized, workflow-based or collaborative e-learning, learning design (LD) has become an important issue in the field of technology-enhanced learning. Nevertheless, current LD approaches turn out to be rather unhandy or costly in teaching and research practice. In this paper, we discuss these shortcomings and propose an alternative solution approach which is based on a web application mashup, learner interactions, and a semantic layer for tool recommendations. As the evaluation of our first prototype is in progress, we can not highlight first experiences, but outline benefits and possible application scenarios in this position paper.

Introduction

Over the last decade, the nature of e-learning changed from a content-based, single learner, self-paced model to multi-learner, multi-tools activity sequences (Dalziel, 2003). According to (Koper & Olivier, 2004), the concept of learning design copes with the definition of multi-role workflows, collaborative peer-interaction, personalization and tutorial support in order to create and run e-learning environments which enable more complex competencies. In further, various learning design specification languages have emerged in the last years, among them the commonly known IMS Learning Design (IMS LD, 2007). In this paper, we explain, why specifications like IMS Learning Design are not applicable in research and teaching practice. Furthermore, we introduce an alternative, more learner-centered approach to learning design.

Overview and Shortcomings of Existing Learning Design Approaches

Typically, web-based e-learning solutions follow one of these technological approaches: Whether they are realized as a monolithic, full-featured system, like Moodle and WebCT, or they are utilizing several tools and services, as can be found in research projects like iCamp (www.icamp.eu) or APOSDLE (www.aposdle.tugraz.at). In some cases developers also integrate single tools into e-learning platforms, such as the embedment of communication technology like Skype into the Moodle system. Independently of the technological solution approach, Mödritscher and Sindler (2005) state that content-based, single learner activities are not sufficient to reach high-level learning objectives. Therefore, the selection of learning activities and tools has to be given (or suggested) by a teacher or the e-learning platform. In literature, the following approaches to learning design can be identified.

First of all, the learning process can be observed and manipulated directly by the teacher, as shown by a study in (Mödritscher, 2006). Although this method is didactically the best one, it is rather costly for larger courses due to the tremendous effort for the teacher. Secondly, research in the field of adaptive e-learning proposes pre-defined rules or even the application of machine learning methods in order to adapt the learning process on the basis of didactical models and learner profiles (Park and Lee, 2004). For instance, Specht and Burgos (2007) describe how to define adaptation rules using IMS Learning Design. Thirdly, younger approaches focus on service-oriented architectures of e-learning systems (Conlan, 2005) and, in further, also on concepts of workflow management to specify the learning process (Helic, 2006). Again, IMS Learning Design can be used for defining the learning process as well as for service orchestration purposes (Vogten et al., 2006). Finally, other solution approaches, like APOSDLE (Ghidini et al., 2007), aim at applying various tools on the basis of a semantic model, whereby the platform might even consist of heterogeneous technologies. Indeed, the underlying semantics could be translated to some ontology language.

Concluding these major approaches, each one – even the one about the teacher's designing the learning process manually – might lead to some kind of XML-based specification language, as given e.g. with

IMS Learning Design. Nevertheless, creating and running courses which include pedagogical models on the basis of such specification languages is, for certain scenarios, rather disadvantageous. On the one hand, practitioners face the problem that there are hardly good authoring tools out there and that using such specification languages is semantically not unique. Thus, it is unclear how e-learning systems interpret a learning process specified with IMS Learning Design, e.g. using the Reload Editor. Moreover, such a learning design approach requires a teacher more effort as well as additional technical and didactical skills. On the other hand, system developers have to spend a lot of time in implementing an interpreter for a learning design specification language, although the utility of such a methodology is still unclear. Yet, even developer communities of open source platforms like Moodle or .LRN are aware of specifications like IMS Learning Design and try to consider it in their system (Burgos et al., 2007) or realized it already (OpenACS, 2006).

An Alternative Methodology to Utilize E-Learning Tools

Because these rather disadvantageous perspectives of learning design mainly effects researchers and educators in the field of higher education, we propose an alternative to current learning design approaches. Precisely, we try to combine two interesting streams in practice: (1) a web application mashup solution and (2) an underlying semantic model for educational scenarios. The depicted methodology is being developed within the scope of the research project iCamp as a solution approach for the so-called “iCamp Space”.

While a mashup solution “*combines multiple sets of data streams into a unified user experience*” (Kulathuramaiyer, 2007), a web application mashup is able to display various web-based tools into one aggregated view. A solution approach like our XoMashup component has to consider certain issues, as explained in detail in (Mödrischer et al., 2008) and summarized here:

- **Cognitive support:** Concluding from mashup visualization techniques (Spoerri, 2007), displaying different applications next to each other requires some kind of cognitive support for users, in order to reduce their cognitive load on working with the system. In accordance with iGoogle, MyYahoo, Netvibes, and other providers of personalized websites, we realized a portal-like OpenACS component, namely the XoMashup application, which allows users to arrange tools along a grid layout.
- **Controllability:** Derived from former experiences in the field of personalized e-learning (García-Barrios et al., 2005) and in accordance with online learning in higher education (Kieslinger et al., 2006), an web application mashup has to give a user the control over the arrangement of and interaction with the tools. Therefore, our XoMashup component allows a user to re-arrange, minimize, maximize, reload and close each window (see also figure 1). Further, it should be possible to launch web applications and even add new ones to the system.
- **Technical requirements:** As usual, browser-based solutions do cause technical restrictions. In the case of a web application mashup, it is necessary to start full web applications with all its scripts and style-sheets as a part of the mashup page. Thus, we implemented our mashup solution on the basis of “iframes”, which is the only way to guarantee an own environment for each tool, but may not be supported by all browsers. Further, the usage of iframes enforces the prevention of DOM manipulation operations which would reset the content of an iframe. Consequently, the grid-based windowing system of XoMashup is realized with absolute positioning and the manipulation of CSS directives.

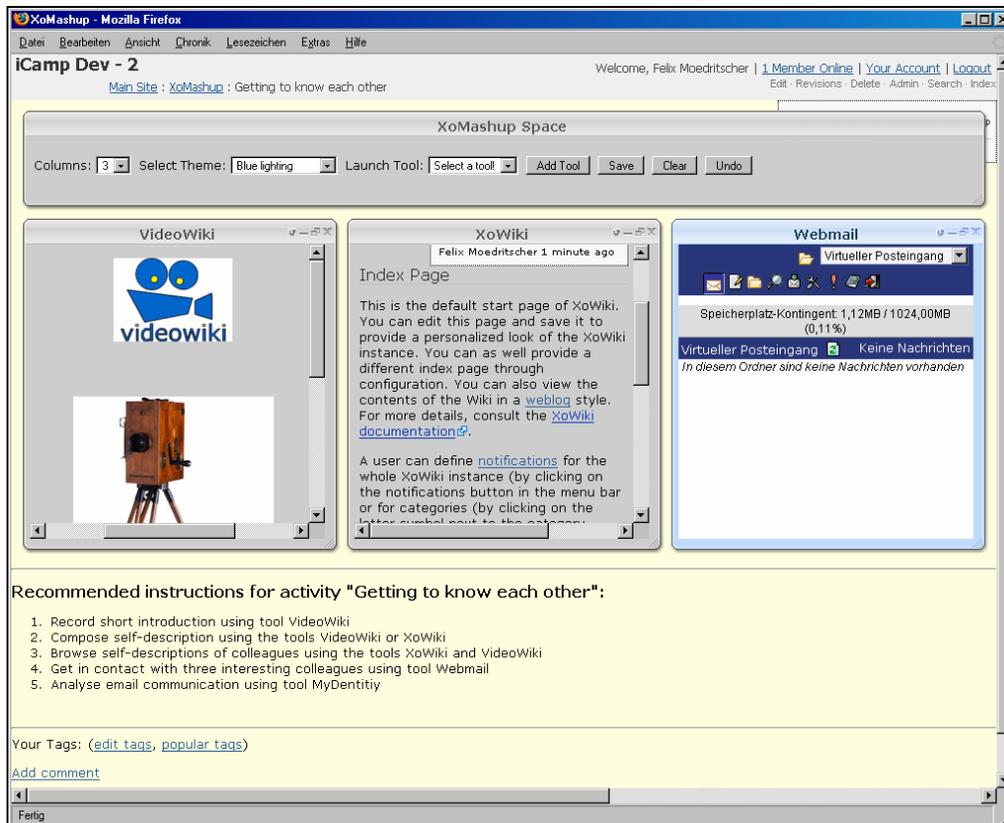


Figure 1: Exemplary view of the XoMashup component

Overall, this web application mashup component can be seen as a very flexible solution for many application areas. Nevertheless, without some kind of underlying semantics it would be nothing more than a customizable portal system, lacking of any didactical support for learners, such as guidance or a personalization strategy. Therefore, we build up a semantic layer which allows recommending tools and tool combinations for pre-defined educational scenarios.

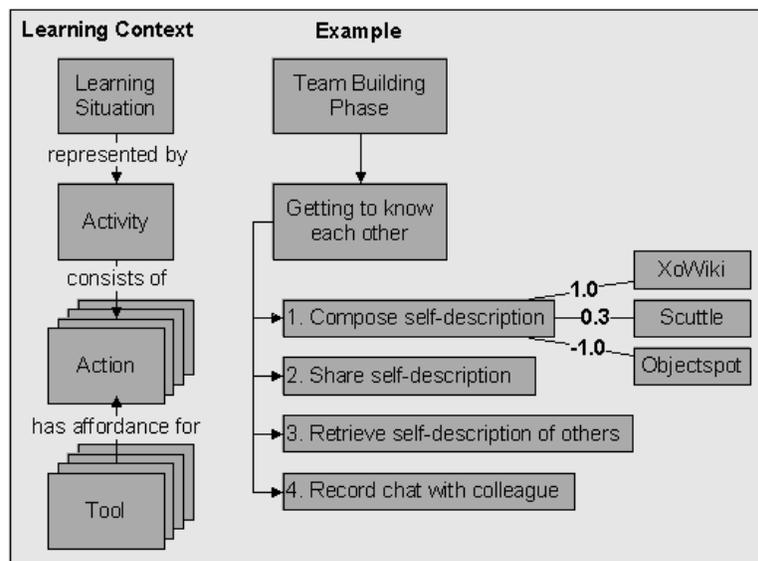


Figure 2: Semantic model of learning activities in the iCamp Space

Following the Activity Theory model (Engeström, 1987) and its application within the INCENSE system (Akhras and Self, 2000), we derived a simplified semantic model shown in figure 2. Basically, we break down the learning context into situations which describe the physical and social environment of learners. In such a situation, a learner is engaged in a so-called activity which might include also tools or other people. Thus, a learning activity is meant to be our basic instructional entity in which learners experience a domain and construct knowledge actively. In our model, each activity consists of list of actions which the user has to perform sequentially. For one of these actions, tools or tool combinations of the iCamp Space can be utilized by the learner, whereby each tool has a certain affordance for each action.

Consequently, this simple model allows the recommendation of the following aspects: (1) actions and action-sequences (activity patterns) for each activity, (2) tools and tool combinations for each action, (3) the usage of the tools, and (4) their arrangement. In the very first approach, we are focusing on the first two issues. Initialization of this model can be done in two ways: (a) by a teacher or other learners or (b) from the scratch, e.g. by letting the learners choose the actions for an activity, or on the basis activity patterns. In the following, the initial model for the tool recommendations is adapted on the basis of the learning behavior, which can be derived by the user interactions with the system.

The Learner Interaction Scripting Language and its Application

To put this semantic model into practice, we foster an alternative way to XML-based ontologies which are applied in many research approaches and commercial solutions currently. Precisely, we specified a domain-specific language called “Learner Interaction Scripting Language” (LISL) in order to allow learners to control the web application mashup, calculate the semantic model and track learner interaction with the iCamp Space.

```
lisl> initialize mashup space with 3 columns
lisl> define [...] action “compose”
lisl> insert action “compose”
lisl> define tool “xowiki” [with] url “http://xowiki.icamp.eu”
lisl> define object “self-description”
lisl> compose object “self-description” using tool “xowiki”
lisl> define action “share”
lisl> insert action “share”
lisl> define tool “wordpress” with url “http://wordpress.icamp.eu”
lisl> share object “self-description” using tool “xowiki”
lisl> launch tool “wordpress”
```

Figure 3: Exemplary LISL code

As highlighted with a few lines of sample code in figure 3, this natural-like language is capable to support three important aspects of learners who interact with the web application mashup: First of all, this scripting language can be used to calculate the semantic model for recommending certain tools for given educational scenarios. This calculation is done on the basis of define, start-action and launch/drag/close-tool statements in the source code. Secondly, learners can control the web application mashup using LISL commands. However, all relevant interactions – like launching or closing a specific tool – can be achieved via UI elements (see figure 1), although it is also possible to perform these operations with LISL code, i.e. to “script” one’s own learning environment. In addition, a user can also define own commands to start actions of and ‘navigate’ through the current activity. Thirdly, the usage of this scripting language allows an extensively logging of learner interaction with the iCamp Space, i.e. also the interactions with the UI elements. Thus, the underlying semantic model can be refined on the basis of this logging information.

In practice, the Learner Interaction Scripting Language is realized as a command line interpreter within the XoMashup component, whereby the single statements are materialized sequentially into Wiki pages using the underlying XoWiki storage facilities. Opening such a mashup page invokes two execution levels: On the one hand, the recommendation model is calculated on the basis of the whole LISL code, including the past log entries of the learner and even of other learners. On the other hand, the web application mashup is rendered according to the current activity and its actions which were started or already achieved by the learner. Thus, a learner can not only revisit the latest UI arrangement of the exit page, but also the last mashup arrangement of activities which were already started. Moreover, it is possible to export the LISL code (without user-specific information) and hand it over to other learners, for instance in the form of activity patterns. Finally, learning

experiences can be shared, e.g. by means of a refined model for recommending actions and tools, or all interactions of a learner group can be examined closer, e.g. to build up a valid didactical model. In this context, the LISL scripts could also be used to generate a learning design description of this didactical model (e.g. with IMS LD) by mining all the scripts of the learners who successfully completed a certain activity.

Conclusions and Next Steps

Concluding this paper, we pointed out disadvantageous aspects of using learning design specifications in e-learning research and practice. We also highlighted the importance of user interactions with the learning tools, which justifies our approach to learning design in two ways: On the one hand, we propose a web application mashup to create an aggregated view of different web-based learning tools. On the other hand, we track learner interactions using our scripting language to (1) build up a semantic model for recommendations, (2) create the learning tool mashup, and (3) refine this model for learning activities. The strength of this approach is twofold: Firstly, it is very handy for learners, i.e. comfortable, if they use the UI elements only, and powerful, if they also start scripting. Secondly, it enables researchers to examine interaction patterns by analyzing the scripts of the learners.

However, the whole approach has not been evaluated in practice yet. Although parts, like the web application mashup solution, are already in use within the iCamp project and contributed to the OpenACS community, the scripting language and the semantic model for action and tool recommendations have to be examined in terms of utility and usability. This evaluation is part of the next iCamp trials which are not finished yet. If this approach is practicable, we also will head towards the application of data mining techniques to automatically generate a XML-based description of learning design (like IMS LD) from multiple sets of LISL code for given learning situations.

Acknowledgement

The iCamp project is funded by the EC IST 6th Framework Project (Contract number: 027163).

References

- Akhras, F.N., & Self, J.A. (2000). System Intelligence in Constructivist Learning. In: *International Journal of Artificial Intelligence in Education* 11 (pp. 344–376).
- Burgos, D., Tattersall, C., Dougiamas, M., Vogten, H., & Koper, R. (2007). A First Step Mapping IMS Learning Design and Moodle. In: *Journal of Universal Computer Science* 13(7) (pp. 683-690).
- Conlan, O. (2005). The Multi-Model, Metadata Driven Approach to Personalised eLearning Services. Dissertation. University of Dublin.
- Dalziel, J. (2003). Implementing learning design: The Learning Activity Management System (LAMS). In G. Crisp, D. Thiele, I. Scholten, S. Barker, & J. Baron (Eds.): *Interact Integrate Impact: Proceedings of the 20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education* (pp. 593-596).
- Engeström, Y. (1987). *Learning by expanding: an activity-theoretical approach to developmental research*. Helsinki: Orienta-Konsultit Oy.
- Helic, D. (2006). Technology-Supported Management of Collaborative Learning Processes. In: *International Journal of Learning and Change* 1(3) (pp. 285-298).
- Garía-Barríos, V.M., Mödritscher, F., & Gütl, C.: Personalisation versus Adaptation? A User-centred Model Approach and its Application. In K. Tochtermann, & H. Maurer: *Proceedings of the International Conference on Knowledge Management* (pp. 120-127).

- Ghidini, C., Pammer, V., Scheir, P., Serafini, L., & Lindstaedt, S. (2007). APOSDLE: learn@work with Semantic Web Technology. In K. Tochtermann, W. Haas, F. Kappe, A. Scharl, T. Pellegrini, & S. Schaffert: *Proceedings of the International Conferences on New Media Technology and Semantic Systems* (pp. 262-269).
- IMS (2007). IMS Global Learning Consortium: *Learning Design Specification*. Retrieved from <http://www.imsglobal.org/learningdesign/> (2007-11-22).
- Kieslinger, B., Wild, F., & Arsun, O. (2006). iCamp: The Educational Web for Higher Education. In: *Proceedings of the European Conference on Technology Enhanced Learning* (pp. 640-645).
- Koper, R., & Olivier, B. (2004). Representing the Learning Design of Units of Learning. In: *Educational Technology & Society* 7(3) (pp. 97-111).
- Kulathuramaiyer, N. (2007). Mashups: Emerging Application Development Paradigm for a Digital Journal. In: *Journal of Universal Computer Science* 13(4) (pp. 531-542).
- Mödritscher, F. (2006). e-Learning Theories in Practice: A Comparison of three Methods. In: *Journal of Universal Science and Technology of Learning* 0(0) (pp. 3-18).
- Mödritscher, F., Neumann, G., García-Barrios, V.M., & Wild, F. (2008). A Web Application Mashup Approach for eLearning. In: *Proceedings of the OpenACS and .LRN Conference* (pp. 105-110).
- Mödritscher, F., & Sindler, A. (2005). Quizzes are not enough to reach high-level learning objectives! In P. Kommers, & G. Richards (Eds.): *Proceedings of the World Conference on Educational Multimedia, Hypermedia & Telecommunications* (pp. 3275-3278).
- OpenACS (2006). ANN IMS-LD v.0.1d tagged and released. Available at: http://openacs.org/forums/message-view?message_id=467112 (2007-12-19).
- Park, O., & Lee, J. (2004). Adaptive Instructional Systems. In D. H. Jonassen (Ed.): *Handbook of Research on Educational Communications and Technology* (pp. 651-684). Mahwah: Lawrence Erlbaum Associates.
- Specht, M., & Burgos, D. (2007). Modeling Adaptive Educational Methods with IMS Learning Design. In D. Burgos (Ed.): *Adaptation and IMS Learning Design: Journal of Interactive Media in Education* (pp. 1-13).
- Spoerri, A. (2007). Visual Mashup of Text and Media Search Results. In: *Proceedings of the International Conference Information Visualization* (pp. 216-222).
- Vogten, H., Martens, H., Nadolski, R., Tattersall, C., Van Rosmalen, P., & Koper, R. (2006). Integrating IMS Learning Design and IMS Question and Test Interoperability using CopperCore Service Integration. In: *Proceedings of International Workshop in Learning Networks for Lifelong Competence Development: TENCompetence Conference*.