Shortcomings of learning design approaches and a possible way out

Conference Item

How to cite:

For guidance on citations see FAQs.
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.
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 (Mödritscher 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 (Garcia-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.
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 1: Exemplary view of the XoMashup component

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.

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 LSL 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: One 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 LSL code for given learning situations.

Acknowledgement

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

References


