Learning Design Patterns: Exploring an inductive analysis approach

Learning Design Patterns: Exploring an inductive analysis approach

dr. ir. Francis Brouns
ICT Developer
Educational Technology Expertise Centre
Open Universiteit Nederland
PO Box 2960
NL-6401 DL, Heerlen
The Netherlands
Tel.: +31 45 5762624
Fax: +31 45 5762800
francis.brouns@ou.nl

dr. Rob Koper
Head Learning Technology Development Programme
Educational Technology Expertise Centre
Open Universiteit Nederland
PO Box 2960
NL-6401 DL, Heerlen
The Netherlands
Tel.: +31 45 5762624
Fax: +31 45 5762800
rob.koper@ou.nl

drs. Jocelyn Manderveld
Educational Technologist
Educational Technology Expertise Centre
Open Universiteit Nederland
PO Box 2960
NL-6401 DL, Heerlen
The Netherlands
Tel.: +31 45 5762624
Fax: +31 45 5762800
jocelyn.manderveld@ou.nl

dr. Jan van Bruggen
Educational Technologist
Educational Technology Expertise Centre
Open Universiteit Nederland
PO Box 2960
NL-6401 DL, Heerlen
The Netherlands
Tel.: +31 45 5762929
Fax: +31 45 5762802
jan.vanbruggen@ou.nl

dr. Peter Sloep
Senior Educational Technologist
Educational Technology Expertise Centre
Open Universiteit Nederland
PO Box 2960
NL-6401 DL, Heerlen
The Netherlands
Tel.: +31 45 5762624
Fax: +31 45 5762800

peter.sloep@ou.nl

Peter van Rosmalen
Researcher
Educational Technology Expertise Centre
Open Universiteit Nederland
PO Box 2960
NL-6401 DL, Heerlen
The Netherlands
Tel.: +31 45 5762624
Fax: +31 45 5762800
peter.vanrosmalen@ou.nl

dr. Colin Tattersall
Researcher
Educational Technology Expertise Centre
Open Universiteit Nederland
PO Box 2960
NL-6401 DL, Heerlen
The Netherlands
Tel.: +31 45 5762929
Fax: +31 45 5762802
colin.tattersall@ou.nl

ing. Hubert Vogten
Senior ICT Developer
Educational Technology Expertise Centre
Open Universiteit Nederland
PO Box 2960
NL-6401 DL, Heerlen
The Netherlands
Tel.: +31 45 5762624
Fax: +31 45 5762800
hubert.vogten@ou.nl

Abstract

Learning design patterns assist the development of effective courses, because patterns capture successful solutions. Pedagogical patterns are commonly created by human cognitive processing in "writer's workshops". Inductive techniques could be used to detect or determine patterns in existing data, or learning designs. This assumes that the learning designs are available in a format that is machine interpretable. The IMS Learning Design specification enables the formal coding of learning designs. We explain that we expect patterns to occur in the method section of a learning design and in particular in acts. We explore several inductive techniques that could be applied to existing learning designs in order to detect and determine patterns and discuss how these could be applied to create new learning designs.

Keywords

Learning design, Patterns, Pattern detection, IMS Learning Design

Introduction

How do we develop effective online courses? At least three different approaches can be followed: (1) designs based on instructional theory, (2) designs based on best practices and (3) designs based on patterns in best practices. These three approaches can be illustrated with the following three scenarios, all starting with the designer having some ideas about the course title, the learning objectives, the course content, the target group and the setting of the course.

In the first scenario, the designer starts by selecting a pedagogical approach that might be appropriate for the design task. Such an approach could be a personal theory or a formal instructional design theory like the ones described in e.g. Reigeluth (1999). The designer creates the course inspired by the prescriptions in the theory. These theories give only limited concrete examples how to solve specific problems in the design task, so many parameters are left to the designer to decide and fill in. When the theory is appropriately tested, well selected and well applied, chances are high that the course is effective.

In the second scenario, the designer does not want to read and study instructional design theories and models. Instead, the designer searches for a comparable successful example course, in the same or another subject field. When an example is found that exactly fulfils the needs, the course can be re-used. However, in most cases only one or more comparable examples can be found that can be used as a guide for structuring and developing the new course. After evaluating the course, the course can be made available for other teachers, including the evaluation remarks. When the examples are effective, well selected and well restructured, chances are high that the course is effective.

In the third scenario, the designer searches for so-called course patterns that can be used to develop the new course. A course pattern is an *abstraction of a set of* best practices to fulfil a recurrent design problem (Alexander, 1979). Such a pattern is a kind of template that can be filled in to create the new course. Patterns can be chained like building blocks to create the new course. After evaluating the course, the course is made available for other teachers, including the evaluation remarks, and new patterns can be derived from this and other courses.

The first two approaches are well understood and applied, but the latter has not been explored and applied until recently. In the patterns approach pedagogical patterns are created for often occurring problems to help designers build their courses. The pedagogical patterns capture effective designs in a domain and are described according to a particular format. The patterns usually are created via deductive technologies, i.e. in "writer's workshops" where experts get together and write down the pattern (Bergin, Eckstein, Manns, Sharp, & Voelter, 2005; E-LEN, 2004; Hernández Leo, Asensio Pérez, & Dimitriadis, 2004; Jones & Stewart, 1999; McAndrew, Goodyear, & Dalziel, 2005). When an inductive methodology is applied, patterns are not created by human cognitive processing, but are detected from patterns in the data, i.e. patterns in courses. The condition for such an approach is that the courses must be structured in a machine interpretable way to *visualise* the data and/or to detect the patterns in the data. The idea explored in this article is whether the recent IMS Learning Design specification (IMS, 2003a; Koper & Olivier, 2004) and its predecessor Educational Modelling Language (EML, 2000; Koper & Manderveld, 2004) is suitable for the detection of re-usable course patterns. We will shortly describe IMS Learning Design, followed by a discussion about the issue of patterns in literature. We will then explore several techniques and methods for pattern recognition and indicate which methods and techniques we consider to provide the best options for reaching our goal. Then we suggest how patterns can be used as a basis for new learning designs. Finally we conclude by presenting future work.

IMS Learning Design

IMS Learning Design (IMS, 2003b) is a semantic notation for units of learning in e-learning, e.g. for courses and lessons. The IMS Learning Design (IMS LD) specification consists of an information model and provides a XML schema to describe units of learning in an interoperable way. The specification supports the use of a wide range of pedagogies by providing a generic and flexible language, and is based on a pedagogical meta-model (Koper & Manderveld, 2004).

A conceptual representation of IMS LD is given in figure 1, while figure 2 provides an alternative, tree representation of IMS LD. A *person* assumes a *role* in the teaching-learning process, typically a learner or a staff role. When in a particular role, the person has to perform (learning or support) *activities* to attain certain *outcomes*. The *activities* are performed in an *environment* consisting of *resources*, *learning objects* and *services* in support of the learning process. *Activities* can be combined into *activity-structures*. In an activity *sequence*, the activities have to be performed in the order as specified in the structure. In an activity *selection*, a given number of activities may be selected from the number present in the selection. *Conditions* can be used to personalise learning designs; for example to specify that a certain activity can be skipped when the learner fulfils a certain prerequisite. *Properties* are used to create and store user information like dossier and portfolio data and are required for personalisation. The pedagogy is expressed in the *method* determining the activities that have to be carried out by a particular *role* at a specific moment to meet *learning-objectives* (specifying outcomes for learners) given certain *prerequisites* (specifying entry level for learners).

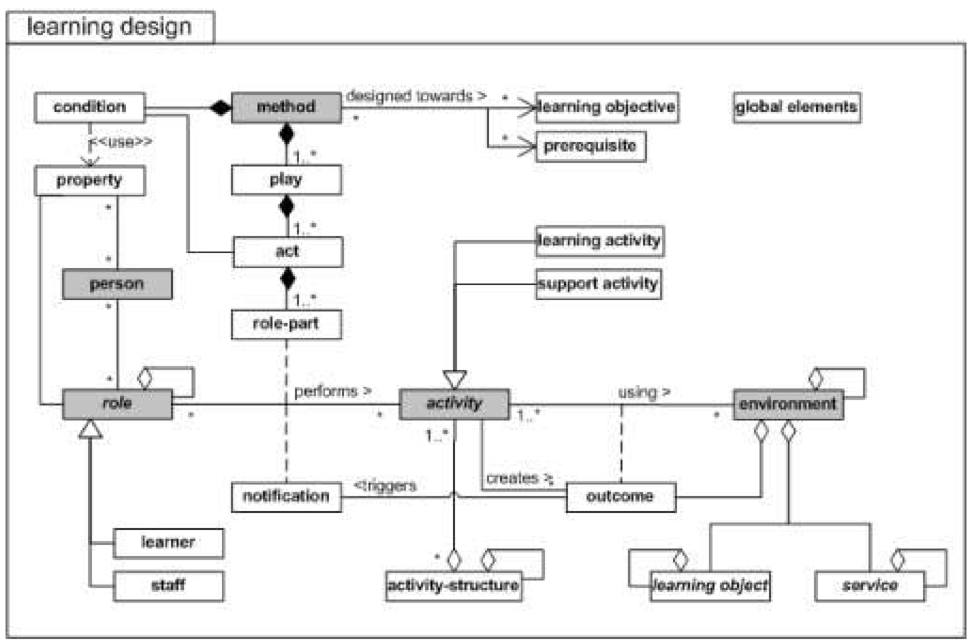

Figure 1. Conceptual representation of IMS Learning Design

The method is the core of the IMS LD specification where the learning process is modelled. All other concepts are referenced from the method (see also figure 2). The metaphor of a theatrical play script is used to describe the method. A *play* consists of sequences of *acts* and acts consist of *role-parts.* The role-parts assign the activity to the roles in an act. The activity describes what the role has to do. The act also specifies the environment in which to perform the activity. When there is more than one role in an act, they run in parallel. In short, a method consists of one or more concurrent plays; a play consists of one or more sequential acts; an act consists of one or more concurrent role-parts. Each role-part assigns an activity or activity-structure to a particular role.

A small example of acts is given in the 'What is Greatness' use case description (Tattersall, 2004b). In the first act, activities are assigned both to the learners and the tutor. The learners first receive an introductory activity. In the second activity the learners enter their initial thoughts. In the same act, the tutor monitors and ends this process. In the second act, the learners are asked to reflect on the work done by the users in the first act. The tutor receives all responses, provides feedback to the learners and determines per learner whether the activity is completed. The pattern would consist of two acts. The first act contains two role-parts. The first role-part links an activity-structure consisting of two learning-activities (first 'introduction', than 'enter initial thoughts') to the learner. The second role-part assigns a support-activity 'monitor the process' to the tutor. In this act, the learners have to perform the two activities in sequence; the tutor activity runs in parallel to this. The tutor determines when the second act starts. The second act also consists of two role-parts. The first role-part of the second act assigns the learning-activity 'respond to others' to the learners; the second role-part assigns the support-activity 'respond to initial thoughts' to the tutor. Again the tutor determines when the act is completed.

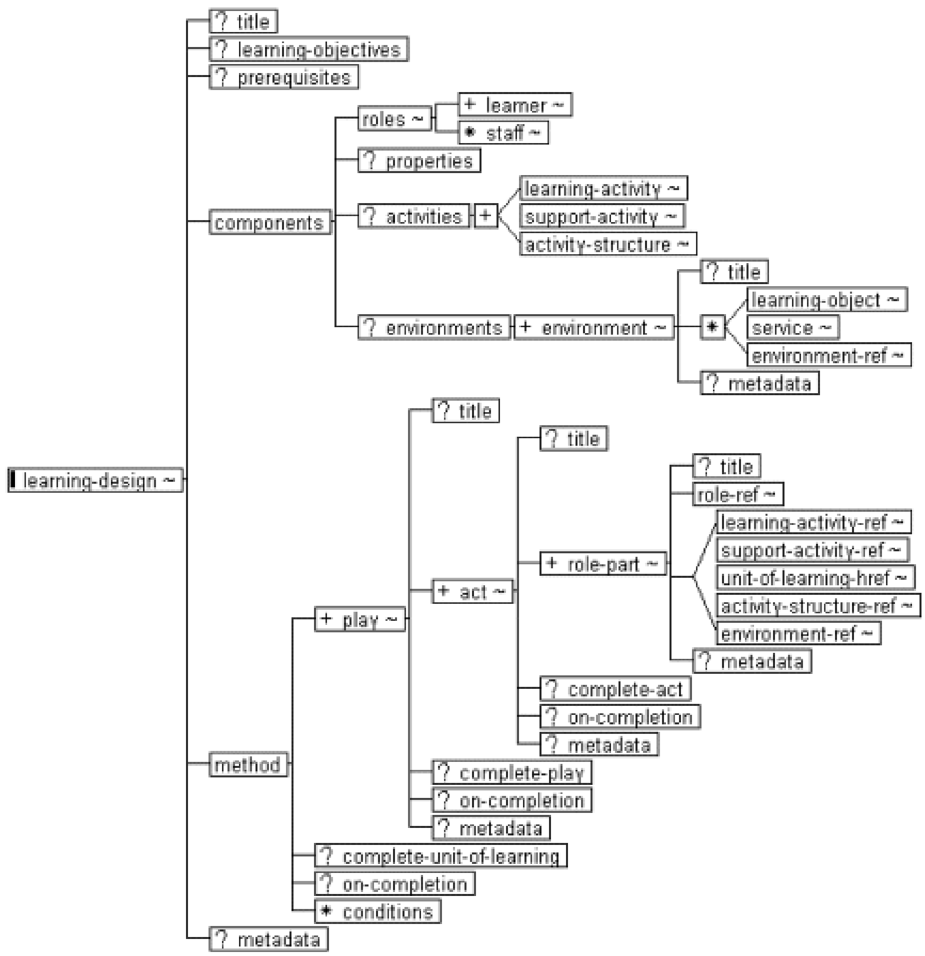

Figure 2. Tree representation of the IMS LD concepts

The basic question elaborated in this article is the question whether it is possible to identify learning design patterns in a series of IMS LD documents. Before we explore this issue, we will discuss the concept of a pattern in more detail.

Patterns

Alexander (1979) was one of the first to describe the principle of patterns based on his observations in the field of architecture. His ideas have been taken up by others; initially in artificial intelligence, programming and software engineering (The Hillside Group, 2005), then increasingly in education (Avgeriou, Papsalouros, Retalis, & Skordalakis, 2003; Bergin et al., 2005; Jones & Stewart, 1999).

The most commonly found definition of a pattern states that a pattern is a model of a solution for a recurrent problem that can be used repeatedly in many contexts (The Hillside Group, 2005; Alexander, 1979; Bergin et al., 2005). The solution has been proven in practice as the one that is most suitable or the solution most often applied for the problem in the particular situation. The solution is generic and abstract, but analysed from a set of best practices. As Alexander (1979) points out, patterns arise out of observed regularity. A solution is considered to be a potential pattern only after one has noticed that similar solutions are applied to a recurrent problem. This observed regularity then is taken as a pointer to analyse the data further in order to try and create an abstraction from it that can function as a pattern.

Patterns in software engineering, or design patterns (Fowler, 1997), are mainly used to facilitate the re-use of successful designs, provide access to proven techniques, and allow choices between alternative solutions. In education, patterns are used for similar purposes. Patterns capture effective learning designs, allow newcomers in education to learn from more experienced developers, provide solutions that can be applied in many circumstances, function as a communication medium and means and enhance knowledge management and knowledge transfer (Alexander, 1979; Bergin et al., 2005; E-LEN, 2005; Hernández Leo et al., 2004; Jones & Stewart, 1999; McAndrew et al., 2005).

When we apply the definition of patterns as given above to learning design we see two aspects of patterns that are important. Firstly, the identification of learning design patterns in a set of units of learning, and secondly, the application of learning design patterns to create the learning designs for new units of learning. In the following paragraphs we will take both aspects into account. First we explore techniques for determining patterns in learning designs.

Methods for pattern detection

Several approaches to detect patterns in learning designs are possible. Learning designs can be inspected, visualised or analysed by various techniques. One can simply look at the learning designs, either as raw data or in a run-time environment, and try to determine patterns. However, this is a very time-consuming and laborious process and imposes many restrictions. A better method is to try to automate (part of) the analysis.

A simple method is to count the frequency and combination of IMS LD elements. We looked at the frequency of elements in a total of 34 distance learning competence based courses, developed for higher education institutes and the Open Universiteit Nederland. The courses were designed to be delivered via the Internet, some contained additional face-to-face components. Average size of the course was 100 to 200 hours. Table 1 provides the frequency of a few IMS LD elements occurring in the courses.

	learning-activity	environment	learning-object	service	play	act	role-part
course 1	54	92	200	1	1	1	3
course 2	9	28	11	9	2	2	3
course 3	11	28	293	7	1	2	10

course 4	16	16	23	7	1	1	5
course 5	6	7	25	2	1	1	5
course 6	56	116	155	4	1	1	2
course 7	30	14	26	2	1	1	4

Table 1. Frequency of selected IMS LD elements

The data in the table shows a wide variety between the courses in the frequency of IMS LD elements. This was to be expected, because the courses addressed different domains and were not of identical size. As the data in the table illustrate, only limited information can be gained from simple frequencies, and it is difficult to recognise any patterns. It is more likely that frequency in combination with other aspects like order or type of elements provide more relevant information. For instance, when a learning design consists of several activities, it is interesting to know that the first one is always an introduction, and the last one provides self-assessment and so on.

Natural language techniques like Latent Semantic Analysis (LSA), techniques like data, text and web mining might be more suitable, because they can analyse, classify, categorise, visualise data and perform analyses to detect patterns. Mining techniques usually combine several other methods, such as LSA, natural language techniques, and algorithms to analyze XML data. Below we give a short description of LSA and introduce some XML analysis techniques.

Latent semantic analysis and indexing

LSA is a theory and a method to extract and find concepts and similarity of concepts in large bodies of text, relying on the meaning of words, terms and phrases in the contexts in which they are used (Landauer, Foltz, & Laham, 1998). Concepts are found by analysing the material for the existence of specific terms or phrases in the context. The number of concepts is presented in frequency tables. Latent Semantic Indexing (LSI) and clustering techniques reduce the frequency table by retaining only the most significant concepts (Dom, 2000; Han et al., 1998; Jain, Duin, & Mao, 2000). LSA is particularly suitable for clustering and classification of documents and has been applied to a wide range of document types, including XML and web documents (Dom, 2000).

Analysis of XML

XML data are always hierarchical and presented in trees. The XML tags present the syntax in which to express the semantics. There are several visualisation techniques for XML data. The growing use of XML for documents and interchange of data resulted in numerous efforts to analyse XML (documents) in order to detect changes (Cobéna & Abiteboul, 2002; Wang, DeWitt, & Cai, 2003) or similarities (Weis & Naumann, 2004; Zhao, Bhowmick, Mohania, & Kambayashi, 2004) in XML documents or to combine XML data into a new XML dataset (Guha, Jagadish, Koudas, Srivastava, & Yu, 2002). Le Grand (2002) describes the intricacies of visualising XML topic maps. Topic maps (XTM, 2003) represent the structure of information resources used to define topics, and the associations between topics. The resources are XML documents, thus hierarchical. Topic maps are not hierarchical and can become very complex. A conceptual analysis, based on formal concept analysis, where a lattice of concepts is generated, is presented as a solution. Le Grand (2002) also mentions applications based on the graph theory to visualise the topic maps. Others have reported the use of graphs for the representation of trees (Shash, Wang, & Giugno, 2002).

The frequency tables produced by LSA and LSI techniques tend to increase rapidly with the growth of the document database. Yoon et al. (2001) proposed an approach that decomposes the frequency table into a matrix, taking the structure of XML documents into account. Here XML documents were represented as bit maps and Bit Cube (a 3-dimensional representation of the bit maps). Operations carried out on these matrices resulted in faster information retrieval. Several of these matrix operations can be directed towards pattern recognition.

The combination of algorithms developed for web mining and detecting changes in websites and XML tree transformations as described by Zhenchang (2002) might also be useful for comparison of learning designs. The changes in the website are determined by looking for changes in the XML trees (XML tree comparisons, tree isomorphism, repetition finder, tree-to-tree corrections) and algorithms for mining change patterns.

Recognising patterns in learning designs

How could we apply the above mentioned techniques to recognise patterns in learning designs?

Whatever technique will be used, it will be difficult to distinguish patterns in an unsorted collection of learning designs. It is more likely that pedagogical patterns will be detected in learning designs that already have been grouped on a common basis. Jain et al (2000) and Dom (2000) showed that the application of filters resulted in a more effective pattern recognition. A good filter for learning designs would be a classification on the basis of the type of unit of learning, e.g. categorised into classes or subsets on the basis of the learning objectives. Each subset then contains only those learning designs that target specific learning objective; e.g. learning designs for problem-based learning, learning designs for collaborative learning etc. Classification can be done on the basis of concepts that are common in the learning designs and learning objectives. Techniques like LSA or text mining are based on determination of concepts, and therefore are likely candidates for this first classification. This method would classify learning design according to learning objective and results in rather broad patterns. The pattern consists of the (XML) elements and constructs that occur in all learning designs belonging to one subset.

These rather large subsets have to be classified further to become meaningful. This could be done by looking at the titles supplied for the elements/constructs (such as activity, environment, role, learning-objects, etc) and (summaries of their) content. An appropriate method to type or classify an activity is to look at the use of particular verbs in the activity description. An activity presents an instruction to the user. The verbs used, e.g. make, read, write, assess, etc, determine the type of instruction. Again, natural language technologies such as LSA offer the most logical choice to be applied in determining the type of the element.

As IMS Learning Design documents are expressed in XML, techniques to analyse structured data like graph theories, tree comparison, topic maps, etc, could be used in addition to classification techniques. Techniques based on structured data can be used not only to extract patterns via concepts in the content data, but also to look for patterns in the XML tree itself.

Where to find patterns in Learning Design?

Method or act as building blocks

We expect patterns to occur in the learning design *method* of a learning design. In order to be as flexible as possible, we need to identify small re-usable components in the method section. A method comprises plays, acts and role-parts. In theory there are several solutions to identify patterns in learning design methods. First, the complete method section can be used to model a pattern. Especially when units of learning are small (e.g. a few study hours) this would be a good approach. When units of learning are larger (e.g. up to 300 study hours), smaller components must be identified in order to allow for enough flexibility. One can argue that these large units of learning should be built from smaller, independent units of learning. So, one approach could be always to base a pattern on a method section of a unit of learning and in the case of larger units of learning, first to split the unit of learning up into smaller, independent ones. Another approach is to build units of learning on *acts*. Acts are the smallest, independent sections within a method. To build a new unit of learning, several act patterns can be combined to create the new method. Even the content of two acts can be merged (under certain conditions) to create a new act.

We assume that these two approaches are the only two that can be used. Role-parts, the first lower level of aggregation, can be dependent on each other (e.g. a learning activity and a support activity can be related). The smallest independent, and thus re-usable, entity is the *act* within a method section. However, as can be seen from the previous table 1, in some designs only a few acts are used per method. The use of acts is not independent of the type of pedagogy modelled. Typically in universities having open admission and individualized learning paths, like the Open Universiteit Nederland, acts are not used extensively. One of the major uses of an act is to synchronize the activities of the individual learners: for instance to model week or month blocks of work. However, the concept of an act can still be used for patterns, but then the idea of merging acts must be allowed for (to avoid the side-effect of synchronizing when it is not necessary).

Refining the pattern

Although the pattern is to be found in acts, additional information is present in the learning design that is important in determining the pattern. For example, the conditions that are imposed on the method or act or the use of properties influence the pattern. It is highly likely that the pattern not only consists of the structure of the elements, but also in the content, i.e. descriptive text of the activity and the environment. The content is not created in the same document as the learning design. Instead the content is referenced by linking to resources (see figure 3). Figure 3 illustrates that the activity-description does not contain any content, but references, through the item element, a file, 'hw-item-1.html', which holds the actual textual description.

In an IMS LD document the method, and underlying act, refer to declared components like roles, activities, environments, activity structures, properties and conditions to minimise redundancy and increase re-usability (see figures 2 and 3).

To take additional information both from the referenced elements and the content into account, it suffices no longer to use a simple transformation of the method; instead it calls for a complex processing of the method and other sections of the IMS LD document. To look for patterns in a method or act, a transformation is needed to resolve the references to the roles, activities and environments. The structure of the pattern can be found by applying for example XML tree comparison or analysis techniques to the play and acts. This should be combined with text, data mining and LSA techniques to determine type of activity and environment from the supplied titles and content. The pattern would consist of acts detailing the activities or order of activities to be performed by certain roles in specified environments and resources.

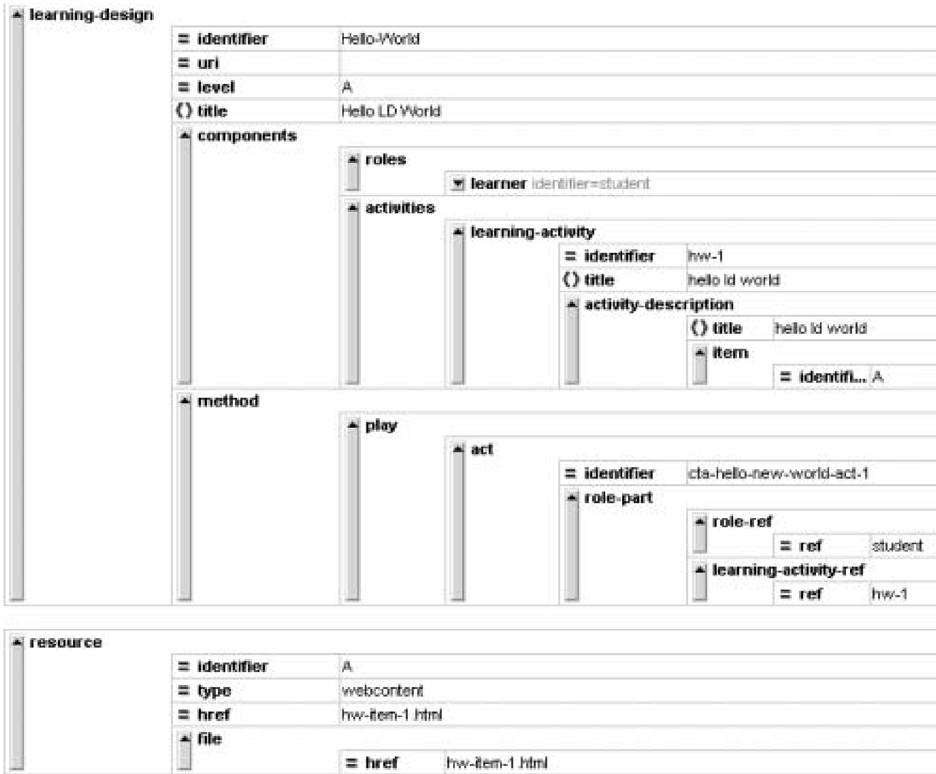

Figure 3. A simple unit of learning illustrating the declarative and referencing structures of learning design (Courtesy Tattersall (2004a)).

The process to identify patterns in learning design

Given the discussion above, the process of identifying patterns in learning designs can be summarized as follows. Patterns are expressed as a method or as an act. At least two units of learning are needed to identify a pattern (the input-set). These units of learning must be classified as best practice within a certain field, e.g. for a certain type of learning objectives. It is not necessary that the units of learning contain the same content, only structural similarities will be identified as a pattern. As indicated before, the method of a unit of learning contains references to roles, activities and environments (see also figures 2 and 3). Additionally, a learning design does not contain the actual content, but refers to resources containing the textual descriptions. Therefore a transformation to resolve the references is needed before pattern detection can start. Also transformations may be needed to make the structures in the XML document of a similar form when alternative modelling approaches are possible. The patterns matching mechanism will first detect structural similarities in the XML structures of the input-set. The output will be a document that contains the common elements in the input-set. Then the pattern can be refined by adding additional relevant aspects that require a more extensive and complex processing of the document. Several of the above mentioned techniques could be applied to different stages of the analysis. XML tree comparison and regular expression techniques will likely be used for the patterns in the method, consisting of specific orders of elements, while text, data mining and LSA will be used to determine type of activity, environment and roles by examining titles and content resources.

Using learning design patterns

Learning design patterns can be used to create new learning designs. In order to accomplish this, templates need to be created based on the patterns. A formal and a human readable description of the pattern have to be created and elaborated. The pattern should be supplemented with examples and additional information detailing its use and intention. The pattern should not only consist of a description but include the learning design formalised in IMS LD as well. The latter serves as a template containing dummy example text to be replaced by the author. The patterns in the form of models must be available to course developers together with the IMS LD templates.
Because learning design patterns are based on classes of units of learning that are considered best practice, it would be good when a description becomes available of the experience in practice together with the underlying units of learning. How effective, efficient and attractive were they in practice? In what setting were they applied? These data can help to determine which pattern is more suitable given the situation of the course to be developed.

Conclusion

There are three approaches for developing effective online learning designs or courses. This article looked at the application of patterns as template for development of new courses. Patterns are defined as effective solutions that are applied to recurrent design problems. Although commonly deductive approaches are used to determine patterns, we suggest inductive techniques to analyse existing material. This is possible, because the IMS Learning Design specification can be used to code designs in a formal manner. In this article we explored the technologies and methods that could be used to recognize patterns in learning designs. The next step is to validate this approach. We will apply the method and techniques discussed above to the existing IMS LD coded courses in search for patterns. The patterns are then translated into template and author's instructions to form the basis for the development of new learning designs. Many advantages can come from this approach. It provides the possibility to build better courses, grounded in actual experience. It also opens up the possibility to further research the effectiveness of certain learning design models and principles. It can assist inexperienced course developers and it can prevent re-inventing the wheel all the time, and to communicate successful designs.

Acknowledgements

The authors wish to thank the management and staff of the Schloss Dagstuhl International Conference and Research Center for Computer Science for providing a pleasant, stimulating and well organised environment for the writing of this article.

References

The Hillside Group (2005), retrieved April 28, 2005, from http://hillside.net/.

Alexander, C. (1979). *The Timeless Way of Building*. New York: Oxford University Press.

Avgeriou, P., Papsalouros, A., Retalis, S., & Skordalakis, M. (2003). Towards a Pattern Language for Learning Management Systems. *Educational Technology & Society, 6(2)*, 11-24.

Bergin, J., Eckstein, J., Manns, M., Sharp, H., & Voelter, M. (2005). *Pedagogical Patterns,* retrieved April 28, 2005, from http://www.pedagogicalpatterns.org.

Cobéna, G., & Abiteboul, S. (2002). Detecting Changes in XML Documents. *Proceedings of the 18th International Conference on Data Engineering (ICDE'02), San Jose, California, February 26 - March 01, 2002*, 41-52.

Dom, B. (2000). Pattern Recognition Meets the World Wide Web. *Proceedings of the International Conference on Pattern Recognition (ICPR'00), Barcelona, Spain, September, 03-08*.

E-LEN. (2004). *Design patterns and how to produce them*, retrieved April 28, 2005, from http://www2.tisip.no/E-LEN/outcomes.php.

E-LEN (2005). *e-LEN project*, retrieved April 28, 2005, from http://www2.tisip.no/E-LEN/.

EML (2000). *Educational Modelling Language,* retrieved April 28, 2005, from http://eml.ou.nl.

Fowler, M. (1997). *Analysis Patterns: Reusable Object Models*. Boston: Addison-Wesley.

Guha, S., Jagadish, H. V., Koudas, N., Srivastava, D., & Yu, T. (2002). Approximate XML joins. *Proceedings of the 2002 ACM SIGMOD international conference on Management of data, Madison, Wisconsin,* 287-298.

Han, E., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., et al. (1998). WebACE: A Web Agent for Document Categorization and Exploration. In K. P. Sycara & M. Wooldridge (Eds.), *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, 408-415.

Hernández Leo, D., Asensio Pérez, J. I., & Dimitriadis, Y. A. (2004). IMS Learning Design Support for the Formalization of Collaborative Learning Patterns. *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04), Joensuu, Finland, August 30 - September 01, 2004*, 350-354.

IMS. (2003a). *IMS Learning Design Information Model. Version 1.0 Final Specification*, retrieved April 28, 2005, from http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html.

IMS (2003b). *IMS Learning Design Specification,* retrieved April 28, 2005, from http://www.imsglobal.org/learningdesign/index.cfm.

Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(1), 4-37.

Jones, D., & Stewart, S. (1999). The case for patterns in online learning. In P. De Bar & J. Legget (Eds.), *Proceedings of Webnet'99 Conference, Honolulu Hawaii, 24-30 Oct 1999*, 592-597.

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

Koper, R., & Manderveld, J. (2004). Educational modelling language: modelling reusable, interoperable, rich and personalised units of learning. *British Journal of Educational Technology, 35(5)*, 537-552.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). Introduction to Latent Semantic Analysis. *Discourse Processes, 25*, 259-284.

Le Grand, B. (2002). *XML Topic Maps Visualisation.*, xmluk.org, retrieved April 28, 2005, from http://www.xmluk.org/slides/RAL_2001/LeGrand/XMLUK1.pdf

McAndrew, P., Goodyear, P., & Dalziel, J. (2005). *Patterns, designs and activities: unifying descriptions of learning structures,* retrieved April 28, 2005, from http://kn.open.ac.uk/public/document.cfm?docid=5295.

Reigeluth, C. M. (1999). *Instructional-design theories and models (volume II)*. London: Lawrence Erlbaum Associates.

Shash, D., Wang, J., & Giugno, R. (2002). *Algorithmics and Applications of Tree and Graph Searching,* retrieved April 28, 2005 from http://www.acm.org/sigmod/pods/tut/sh.ppt

Tattersall, C. (2004a). *Hello world,* retrieved April 28, 2005, from http://hdl.handle.net/1820/309

Tattersall, C. (2004b). *What is Greatness,* retrieved April 28, 2005, from http://hdl.handle.net/1820/313

Wang, Y., DeWitt, D. J., & Cai, J.-Y. (2003). X-Diff: An Effective Change Detection Algorithm for XML Documents. *Proceedings of the 19th International Conference on Data Engineering (ICDE'03), Bangalore, India, March 05 -08, 2003*, 519-530.

Weis, M., & Naumann, F. (2004). Detecting duplicate objects in XML documents. *Proceedings of the 2004 international workshop on Information quality in information systems, Paris, France,* 10-19.

XTM (2003). *TopicMaps.Org,* retrieved April 28, 2005, from http://www.topicmaps.org/

Yoon, J. P., Raghavan, V., Chakilam, V., & Kerschberg, L. (2001). BitCube: A Three-Dimensional Bitmap Indexing for XML Documents. *Journal of Intelligent Information Systems, 17*(2/3), 241-254.

Zhao, Q., Bhowmick, S. S., Mohania, M., & Kambayashi, Y. (2004). Discovering frequently changing structures from historical structural deltas of unordered XML. *Proceedings of the Thirteenth ACM conference on Information and knowledge management, Washington, D.C. USA*, 188-197.

Zhenchang, X. (2002, August 14). *On Trees and Graphs,* retrieved April 28, 2005, from http://www.cs.ualberta.ca/~stroulia/Presentations/Trees_and_Graphs.ppt