An Information Architecture for Courseware Validation

Mark Melia and Claus Pahl

School of Computing, Dublin City University, Dublin 9, Ireland
mmelia@computing.dcu.ie

Abstract. A lack of pedagogy in courseware can lead to learner rejection. It is therefore vital that pedagogy is a central concern of courseware construction. Courseware validation allows the course creator to specify pedagogical rules and principles which courseware must conform to. In this paper we investigate the information needed for courseware validation and propose an information architecture to be used as a basis for validation.

1 Introduction

The combination of instructional logic with componentised learning content known as Learning Objects (LOs) using learning technology is known as courseware. To produce quality courseware, course creators aim to apply specific pedagogical principles to courseware they create. This can be difficult, especially when there are seemingly more pressing issues for courseware delivery, such as standards compliance and deadlines. Unfortunately, the neglection of pedagogy can lead to a course which confuses, demotivates and/or isolates the learner, ultimately leading to the rejection of the course [9].

The importance of pedagogy are therefore paramount. [8] specifies post-construction course validation or auditing as an essential part of a holistic course construction methodology. Checking courseware adheres to some specified pedagogy, which the course creator is committed to, is now possible to automate due to the formal separation of learning design from content [1, **?**] and the annotation of learning content with metadata [7].

From our investigation we have found the literature does not address the diversity of information available for courseware validation. The CoCoA tool maps a course to a concept map so to reason about learning material in the context of the concept map [3]. Baldoni et. al. have investigated the use of logics in courseware validation [2]. In our research we attempt to address the diversity of information available for courseware validation. In this paper we introduce an information architecture which allows for the explicit representation of the information needs of courseware validation. In order to validate courseware, we must identify the various information elements necessary for courseware construction, and bring these elements together under the context of an information architecture for courseware validation.

2 Identification of Information Needs for Courseware Validation

For the course creator, course validation is a complex task, which involves evaluating courseware using the masses of information and knowledge available post construction. This might mean delegating different tasks to experts. Validation of course structure and pedagogy may be delegated to an instructional design expert, while content related issues, could be delegated to a subject matter expert.

In order for the courseware validation process to be automated the knowledge and information the course creator calls on to make course validation decisions must be identified.

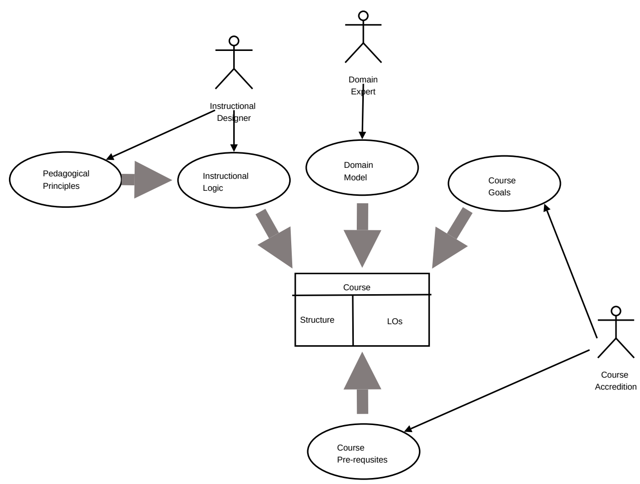

Fig. 1. Course Construction Elements

In figure 1 we outline the different course creation elements coming together to create a course. The course domain information is specified by a domain expert. The scope of the course is usually set by some external force such as the body who accredits the course by setting the pre-requisites and the learning goals of the course. The instructional logic is specified by the instructional designer based on his or her knowledge of pedagogical principles.

3 Layered Architecture for Course Validation

3.1 Architectural Overview

Course aspects are the information which define the ideal scope, content, and design of a course. During course construction these aspects can unknowingly be compromised by the course creator (For example in a case where aspects are set by an individual or body which is not involved in course construction) . By making these aspects explicitly available at the post-construction/pre-delivery stage of the course life-cycle, aspects can then be used to determine the validity of courseware.

By identifying the implicit aspects of course creation and making them explicit and available post-course construction/pre-delivery, we hope to be able to validate the courseware using the aspects identified in figure 1. The aspects of course construction which we have identified must be formalised in order for their use in course validation.

Course construction aspects can be explicitly represented in a layered architecture, similar to that used in the LAOS architecture discussed in [4].

Our layered architecture consists of a domain model, a goal and constraint model, a learner model, a course specification, and a validation model. The domain model describes the course domain in a formalised fashion, this allows for a validator to indicate whether there are problems with how the course is structured with reference to the domain being taught. The goal and constraints model allows the course creator to specify the goals of the course and constraints on the domain such as pre-requisite constraints in the context of the domain model. The learner model is used to model any information the course creator might have about the learner, such as assumed knowledge (possibly the course pre-requisites). The course layer is a formalism of the course, typically using one of the popular courseware specifications [1, 6]. The validation model allows the course creator to specify what is and what is not valid in a course (i.e. the pedagogical principles for the course).

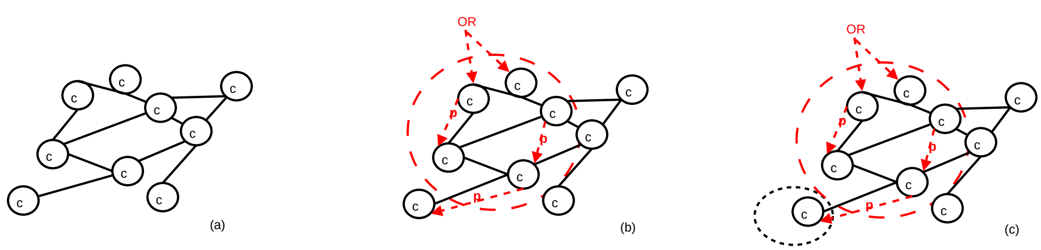

Fig. 2. (a) Domain model, (b) goal and constraint model layered on domain model, (c) learner model layered on domain model and goal and constraint model

3.2 The Domain Model

A domain model at its simplest level is a conceptual graph, where a node represents a concept and an edge represents a relationship between concepts. Figure

2(a) demonstrates the formalisation of a domain model. A domain model should be pedagogically neutral.

Ideally a domain model should be located from a respected third party, such as the curriculum development ontology being developed for the U.S. National Science, Mathematics, Engineering, and Technology Education Digital Library (NSDL) [5]. Domain models may be altered by the course creator to align it with the course creators view of the domain.

3.3 The Goal and Constraints Model

The goal and constraint model represents fundamental pedagogical information. Overlaid on the domain model, the goal and constraint model can specify the goal concepts in a domain model (what concepts the course aims to teach), and instructional constraints such as pre-requisite concepts indicating a conceptual ordering constraint.

In figure 2(b) the goal and constraint model is in indicated by a dashed line, modeled on top of the domain model. The dashed ellipse in the goal and constraint model indicates the goals of the course (the concepts to be covered in the course. The dashed arrows labeled “p” denote a pre-requisite relationship between concepts. The “OR” relationship between two concepts indicates that it will suffice for the purposes of this course that only one of the concepts referenced by the “OR” relationship will suffice.

3.4 The Learner Model

A learner model can be used to capture learner knowledge at a given point in time, and also learner preferences and details. In course validation we can use a learner model to capture the assumed knowledge the learner will have when initially taking the course (the course pre-requisites). The learner model is a necessary layer of the validation architecture as there may be concepts in the course which are not within the scope of the course as specified by the goal and constraint model. One of the concepts outside the scope of the course may be a pre-requisite of one of the concepts in the course, but knowledge of this pre-requisite concept is assumed by the course creator. In order for a validation engine to recognise the learner’s initial assumed knowledge it must be modeled. Figure 2(c) demonstrates a concept which is outside the goals of the course (dotted circle) and is a pre-requisite of a concept, which is a goal concept. The pre-requisite concept is assumed knowledge as it is part of the learner model (dashed circle), therefore the absence of the pre-requisite concept in the course will not cause the course to be invalid.

3.5 The Course Model

A course structure can be modeled as a Directed Cyclical Graph (DCG). Learning resources can be found at numerous points of the DCG. A learners traversal through the DCG depends on variables, such as assessment results, learning styles, feedback, which are assessed at run-time at branching points in the course. Learning resources in the course are annotated using some standardised annotation language such as IEEE LOM [7]. This annotation maps the learning resource to the concept(s) it addresses in the domain model.

The diagram in figure 3 demonstrates how each LO in a course can be associated with a concept in a domain model, by referencing the concept the LO addresses (LO conceptual annotation is indicated in the diagram with an arrow from a LO to a concept). LO conceptual associations can be used to group LOs. Grouping are made up of LOs concerned with the same concept. This type of LO grouping is demonstrated in the diagram in figure 2 (dotted circles indicates conceptual groupings). Once we can group LOs by concept we can discriminate pedagogical strategy between strategies concerned with inter-conceptual pedagogical issues (strategy concerning sequencing of the actual conceptual groupings) and those concerned with intra-conceptual pedagogical (pedagogical strategy concerning the LOs within each conceptual grouping).

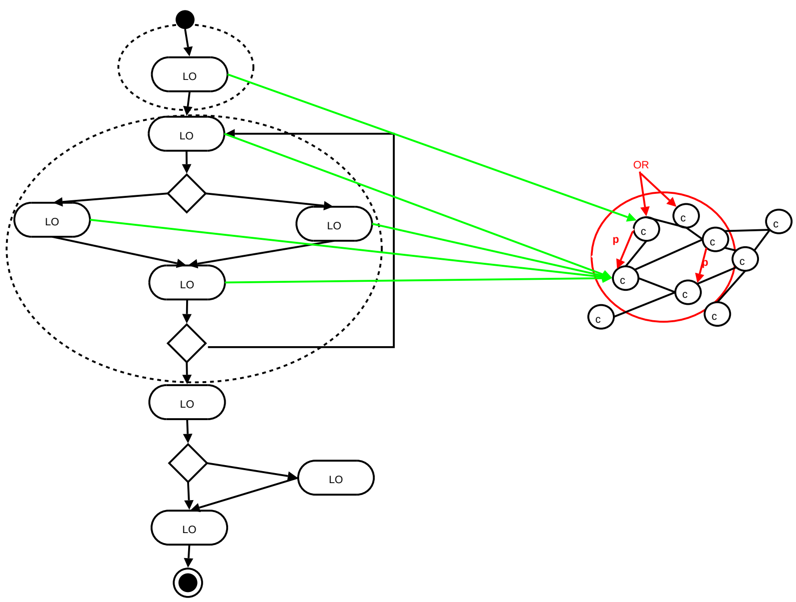

Fig. 3. Grouping LOs according to the concept they cover

3.6 The Validation Model

The Validation Model looks to capture pedagogical rules, which the course model must adhere to. At the validation model layer the course creator can express undesirable properties of a course.

Validation can be split into two distinct parts, validation which concerns one domain concept (i.e. intra-conceptual validation) and validation which looks at how the course proceeds from one concept to another in the course (i.e. inter-conceptual validation). Typical intra-conceptual validation will ensure that each concept teaches that concept in a uniformed manner, such as "an example will be provided with each new concept presented to the learner". Inter-conceptual sequencing might ensure that the conceptual sequencing strategy is one where more general concepts are presented to the learner before more specialised ones.

4 Discussion

In this paper we have given an overview of the information needs for the validation of courseware. We have then taken the information needs identified and constructed a layered architecture, which allows for the explicit representation of each of these information needs for the purposes of courseware validation.

References

1. Advanced Distributed Learning. SCORM 2004 Overview, 2004. Available from: http://www.adlnet.gov/scorm/index.cfm.
2. M. Baldoni, C. Baroglio, V. Patti, and L. Torasso. Reasoning about learning object metadata for adapting SCORM courseware. In *Proceeding of the International Workshop on Engineering the Adaptive Web: Methods and Technologies for personalization adn adaptation in the Semantic Web (EAW204)*. Springer-Verlag LNCS Series, Aug 2004.
3. P. Brusilovsky and J. Vassileva. Course sequencing techniques for large-scale web-based education. *International Journal Continuing Engineering Education adn Life-long Learning*, 13(1/2):75–94, 2003.
4. A. I. Cristea and A. de Mooij. LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators. In *Proceedings of The Twelfth International World Wide Web Conference (WWW03), Alternate Track on Education*. ACM, May 20th - 24th 2003.
5. A. Gupta, B. Ludascher, and R. W. Moore. Ontology services for curriculum development in nsdl. In *Proceedings of the Second Joint Conference on Digital Libraries (JCDL02)*. ACM, July 13-17 2003.
6. H. Hummel, J. Manderveld, C. Tattersall, and R. Koper. Educational modelling language and learning design: new opportunities for instructional reusability and personalised learning. *International Journal of Learning Technology*, 1(1):110–126, 2004.
7. IEEE Learning Technology Standards Committee. LTSC WG12:Learning Object Metadata, 2002.
8. P. V. Rosmalen, H. Vogten, R. V. Es, H. Passier, P. Poelmans, and R. Koper. Authoring a full life cycle model in standards-based, adaptive e-learning. *Journal of Educational Technology and Society*, 9(1):72–83, 2006.
9. J. W. Samples. The pedagogy of technology - our next frontier? *Connexions*, 14(2):4–5, 2002.