Pedagogical Validation of Courseware

Mark Melia

June 20, 2007

Document: MSc. to Ph.D. Transfer Report

Supervisor: Dr. Claus Pahl

Address: School of Computing, Dublin City University, Dublin 9, Ireland

Email: mark.melia@computing.dcu.ie

Abstract

Creating courses by combining Learning Objects (LOs) sourced from third parties, known as courseware, is becoming more popular of late. Although courseware has advantages over traditional courses, it can be difficult to ensure a consistent instructional approach and strategy that satisfies the original courseware construction concerns, due to the abstracted view of LOs the course creator now has. Courseware validation, validates a constructed course against a validation model which specifies correct and incorrect course behaviour. Our approach to courseware validation allows the course creator to specify a validation model using a novel information architecture which captures the major course construction concerns. Validation can then take place within the context of this archtecture. This ensures the original concerns are realised in the constructed courseware.

1 Introduction

1.1 Background

Courseware is the packaging of learning material for its focused delivery to the learner in a pedagogically sound manner, where pedagogy is the strategy or approach to the learner's learning. Courseware authoring research investigates the efficient development and deployment of quality courseware through technology and development methodologies in an intuitive way.

Recent trends in courseware technology and Technology Enhanced Learning in general have enabled new opportunities and challenges for courseware authoring. These trends include:

- standards compliant - allowing the courseware portability[15, 1]
- componentised - course elements are becoming componised into Learning Objects (LOs) - small, reusable instructional units typically a lesson, assessment quiz, or possibly a tutorial [29]. This allows for reuse of third party course material. Discovery of LOs has been aided with the advent

of the IEEE Learning Object Metadata (LOM) standard [16] and Learning Object Repositories (LORs), such as the NDLR[1], LORNET[2] and JORUM[3]

- Personalised and personalisable learning content - using technologies from Adaptive Hypermedia, Adaptive Educational Hypermedia (AEH) sets out to give the learner a more individualised experience taking into consideration learner aspects such as prior knowledge, learning preferences, learning style and knowledge acquisition [11, 12].

Authoring tools have been developed which allow for the production of standards compliant courses, based on the intuitive and efficient authoring of AEH learning content [6, 10] and the combination of LOs [28, 24].

1.2 Motivation

One of the most pressing challenges in course construction is placing the right LO in the right place in courseware [29]. This can cause problems as misplaced LOs can be counter to the pedagogy used in the courseware, which can lead to a course which can confuse, demotivate and isolate the learner and could ultimately lead to the rejection of the course by the learner [26].

This problem escalates as LOs get more complicated in design and are increasingly developed by some third party entity. Course creators rely on high level description data in the LO's metadata when making decisions on LO inclusion in courseware. LO metadata does capture more data about the LO, but this demands a more pressing cognitive load on the course creator. The course creator therefore has only a high level understanding of the LO.

The only real gage into whether the course creator has been successfully integrated LOs to test the course on actual learners [13]. Should there be problems with the courseware, unforeseen by the course creator the untested courseware could have a negative effect on learning.

The importance of pedagogical principles, which define the pedagogy being used in courseware, are therefore paramount and post-construction course validation or auditing is an essential part of a course construction methodology that checks that all aspects of courseware are pedagogically compliant [25].

1.3 Overview of Report

The next section uses the motivation presented in this section to specify a research question leading to research problems and objectives. Section 3, investigates state of the art approaches to courseware validation. In section 4, we look at the research problems and objectives we have investigated to date. Section 5, Future Work, looks at the research problems and objectives still to be addressed and proposes a schedule for addressing these issues. Our concluding thoughts are then contained in section 6.

---

[1] http://www.ndlr.ie

[2] http://www.lornet.org

[3] http://www.jorum.ac.uk

2 Research Overview

2.1 Research Question

From the motivation section, we have identified the need for the validation of courses (where validation is the identification of explicit, defined pedagogical problems in a course structure or the choice of LO), the analysis of problems found in a course and the subsequent need to correct those problems, preferably before the learner takes the course. Our research question looks at whether or not it is possible to automate the analysis of a courses to the extent where it is possible to locate and diagnose pre-defined problems that exist in a course. As course validation cannot exist in isolation but must be part of a course construction methodology we also wish to integrate validation into an existing course construction methodology. Our research question is therefore:

> How can the various pedagogical concerns that exist in course creation be explicitly defined by the course creator and used in the automated validation of courseware as part of an established courseware construction methodology?

2.2 Research Problems and Objectives

In addressing our research question we have anticipated the following research questions and challenges:

- Identification of the information needs for courseware validation.
- Define courseware validation in the context of courseware construction.
- Investigate the explicit representation of course validation information needs
- Development of an architecture for courseware validation.
- Develop a strategy for courseware validation in the context of the courseware validation architecture.
- Look at how courseware validation can be integrated with existing course specifications such as SCORM and IMS LD.
- Design and implement a proof of concept prototype application which uses the validation architecture.
- Evaluation of the course validation architecture.

3 State of the Art

Assisting the course developer in course construction has been investigated in various research projects. Many approaches have been taken to make course construction more efficient and effective.

Adaptive Hypermedia (AH) allows for personalised eLearning, the authoring of AH for education, is known as Adaptive Educational Hypermedia (AEH). The authoring of AEH using a model based approach, has been investigated in [10, 7]. Approaches to developing courseware using software engineering methodologies have been investigated in [14, 30, 27]. In this section we will examine in detail two projects which approach course validation and verification from two different angles, logic-based and concept-map based.

3.1 Logic based Course Planning and Validation

The use of logics and reasoning for course planning and validation has been investigated by the ALICE project at the University of Torino [3]. The project looks at a range of course construction activities including course verification [4] and construction [3].

Temporal projection is used for course verification. Courses represented using action theory, where each course component is an action with pre-conditions and post-conditions. Traditional AI reasoning, such as temporal projection, can be used to check that all pre-conditions in the action theory are respected.

Curricula models allow for restrictions and constraints to be placed on possible learning resource sequences. Curricula models can be formalised using temporal constraints. The models are independent from the learning resources uses and operate on the knowledge level. For example a possible constraint might be that knowledge element α must be learned before knowledge element β. Using linear-time temporal logic (LTL) to represent temporal constraints allows for the verification of the curriculum.

Should the course be deemed invalid, it is important to be able to outline to the course author the reasons for invalidating the course. Temporal explanation is used to explain the reasons for a course failing validation.

The motivation behind the Wlog system is to validate Italian student study plans. A study plan is a list of courses a student takes in University. Each year students may alter their study plan, but these alteration may have adverse affects to the students overall learning goal. Study plan creation and validation is at a very high level in course creation and management.

An attempt has been made by the authors to examine how this technology could be applied at lower level aspects of courseware creation [4], even embracing standards being developed in the area such as SCORM [1] and the IEEE LOM [16].

Although the ALICE group has looked at more fine-grained LO composition and verification [4], the majority of their work concentrates on the verification and construction of study plans, constructed of course modules [2, 3]. This is particularly interesting in the context of the Bologna Process [21], which requires course content being verified against external specification, being promoted at European level.

3.2 Concept-based Course Validation

The Concept-based Courseware Analysis tool (CoCoA), developed at Carnegie Technology Education, uses two types of relationships for validation; typed items and advanced concept roles [5].

Typed items allow for validation of the positioning of particular teaching operations.

Advanced concept roles defines a LO with regard to pre-requisite knowledge and knowledge outcome. A concept has two types of pre-requisites and two types of outcomes, strong and weak. Strong pre-requisites or outcome indicates that deep knowledge of the concept referenced is required or obtained for the specified learning item, while weak pre-requisite or outcome indicates that only surface knowledge of the concept referenced is required or obtained for the specified learning item.

CoCoA checks only sequential learning paths through learning material. This is done by simulating a learner's progression through the learning material. The tool then generates a report once all simulations are complete This report will indicate any "content holes", where the learner encounters a LO without the necessary pre-requisite knowledge needed for the LO.

CoCoA validation goes beyond just validating basic course sequencing, the tool also has a variety of rules relating to type and location of learning items in a course.

CoCoA was prototypical in nature and as such there is no consideration for TEL standards. Pedagogical problems are defined by the CoCoA developer, there is no facility for the course creator to manipulate the validation rules, which causes problems with user acceptance. CoCoA was not developed using an extensible architecture which would allow for the inclusion of unforeseen pedagogical rules in the future. It also does not reflect the complexity of validating the modern course as all courses validated using CoCoA must be linear in nature with no branching points.

CoCoA for what it was, an ad hoc prototype on the potential of course validation, demonstrates the viability of course validation and also shows that it is not a trivial problem.

3.3 Discussion

In this section we have looked at two pieces of research looking at the verification and validation of courseware. Baldoni et. al. uses logics to define knowledge-level constraints (curricula models), and also defines pre-requisites at the LO level (action theory). CoCoA, allows pre-requisites and pedagogical rules to be defined at a conceptual level. Common to each approach is a high-level constraint model which defines conceptual sequencing and also more direct rules specifying LO level rules.

Both approaches have not considered the context which with their work could fit into the wider scope of courseware authoring. They are also very restrictive in that all pedagogical rules are embedded in the programming code of each system, meaning that the course creator cannot alter the pedagogical rules and strategies.

The two approaches give a good starting point for course validation research but are too restrictive to be actually adopted by course creators, which is what our research will look at.

4 Work to Date

In this section we will examine the research problems and challenges from Section 2.2, which we have encountered to date, the possible solutions we have looked at (if applicable), and specify the solution we have decided to use and reasons for this.

4.1 Identification of Information Needs for Courseware Validation

The courseware lifecycle produces a range of data which can be used in validation. During courseware construction a range of implicit and explicit information is used to define its scope, pedagogy and content. This information once identified and defined can be used to validate the courseware post-construction/pre-delivery created. Post-delivery learner interaction information can be added to the existing information for more accurate results.

We focus on validation at the post-construction/pre-delivery stage of the courseware lifecycle as this limits the information space that must be processed and also examines the widely held belief that courseware must be tested with learners before its merits can be assessed [?]

Figure 1 outlines the courseware definition information used in course construction and the various roles which specify this information. The roles can be embodied by the course creator or outsourced to experts. The constructed courseware is defined as a package having two parts, the course structure and the learning content (LOs).

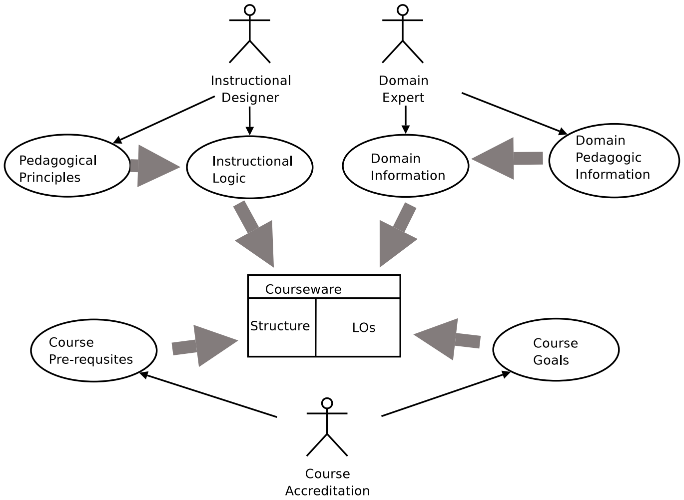


Figure 1: Course Construction Elements

The course domain information, which 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. From the state of the art we can specify two distinct types of pedagogical constraints which are placed on a course - knowledge level (pedagogical strategy) and LO level (pedagogical rules). The knowledge level constraints specify conceptual sequencing constraints, whereas the LO constraints specify rules concerning the type and position of LOs in a course. This distinction is important as the representation of the two types of pedagogical constraints must be examined separately in the context of courseware validation as they express very different aspects of the instructional strategy.

4.2 Courseware Validation in the Context of Courseware Construction

Here, we outline the role of post-construction/pre-delivery course validation in the context of courseware authoring. This investigation was necessary as we recognise validation as part of an overall courseware authoring process.

The diagram in figure 2 outlines the stages before and after validation in the courseware authoring process. Courseware validation expects two inputs - a validation model and courseware. The validation model specifies explicitly the courseware construction concerns, the coursware is then validated against these concerns. The validation process specifies the problems in the courseware in a way which can lead on its the correction.

Figure 2 includes a correction engine, as we foresee automated correction as the natural progression of a software architecture which validates a courseware. The correction engine recommends the best approach to solving course problems found by the validation engine. This is important as we must be conscious of the input needs of a potential correction engine when specifying the problems found in

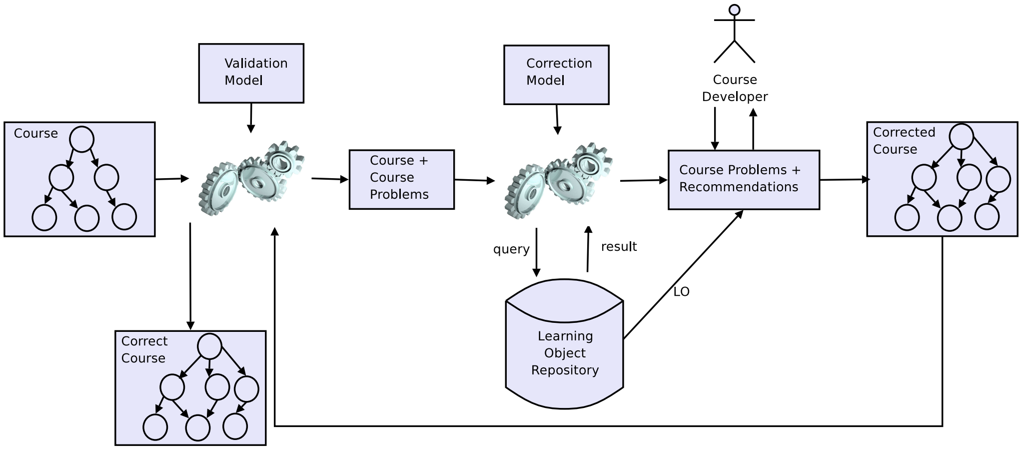


Figure 2: Course Validation and Correction Framework

courseware.

4.3 Modelling Courseware and its Construction Concerns

To address the issue of how courseware and its construction concerns should be modelled to enable validation, we have looked at related work. Two AEH authoring approaches work on the principle of separation of AEH concerns to enable a more simplistic authoring environment. We outline both these approches here and extend one into the Courseware Authoring Validation Model (CAVaM).

4.3.1 Layered Information Architecture

The LAOS architecture is a layered model-based conceptual architecture that allows for the explicit definition of specified aspects of AEH [7]. In the LOAS model there are two types of models - static and dynamic. The static elements of LAOS are:

- domain model (DM) - organizes and structures knowledge in a particular area
- goal and constraints model (GM) - used to express educational goals, which is done by specifying weights and ordering for domain concepts
- user model (UM) - used to specify the user knowledge level, interests and learning styles.
- presentation model (PM) - model variables to do with different course delivery environments

The domain model is the base model on which the goal and constraints model is built. A User model uses the domain model to describe the learners knowledge.

Dynamic elements in the LOAS model are encapsulated in the adaption map (AM), which describes how the static elements behave. The adaption model is based on the 3-tier LAG model of adaptive specification [9]. At the top level of the LAG model is adaptation strategies, which are built on adaptation languages, which, in turn are built on direct adaption rules.

An implementation based on the LAOS architecture exists, known as "My Online Teacher" (MOT)[8], which is used for authoring AEH courses.

4.3.2 Multi-Model Information Architecture

The design of the Adaptive Courseware Construction Toolkit (ACCT) is based on the separation of the "key design elements of personalised eLearning" [10]. These elements are (but not limited to) as follows:

- Narrative Structures - pedagogical structures which specify a sequence of learning activities and tasks.
- Activities - Activities are usually some task which must be performed in the eLearning environment. There can be tools associated with particular tasks.
- Subject Matter Concept Space (SMCS) - Knowledge about the subject area. The (SMCS) also provides scope for the the adaptive courseware and limited pedagogic information such as dependencies between concepts.
- Narrative Attributes - Narrative attributes describe some form of adaptation. It describes information needed to make decisions on adaptivity, which is passed to "selectors" which selects the most appropriate content.
- Learning Resources - Information stored in learning resource's metadata
- Learner Profiles - A learner profile describes a learner in terms of the knowledge the learner has.
- Personalised eLearning Designs (PEDE Narrative) - The resulting personalised course based on the above personalised eLearning design elements is a PEDE Narrative. The PEDE narrative captures all the information needed to deliver the adaptive course.

Each design element in the ACCT is not dependent on any other design element, which makes them very portable. For example, the narrative structure which specifies a good web based learning pedagogy can be used in many courses. Design elements are not a comprehensive list, other design elements can be added to extend the adaptivity created by the ACCT [10].

The combination of a concept map with pedagogic information can limit the portability of a SMCS to only those courses where the course creator agrees with the set pedagogy. In the context of the ACCT this is not a problem, as in most cases the course creator will create their own SMCS, but does not allow the concept map to be used in other contexts.

4.3.3 Findings and Conclusions

We base CAVaM on the LAOS model as it separates the domain model from pedagogic information, such as a allowing for domain model re-usability.

Figure 3 demonstrates our course construction aspect information architecture. Our architecture consists of a domain model, a goal and constraint model, a learner model, a courseware specification, and a validation model. The domain model describes the domain allowing validation to indicate 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 goal of the course and constraints on the domain such as pre-requisite conceptual constraints, the learner model is used to model any information the course creator might have about the stereotypical learner, such as assumed knowledge (possibly

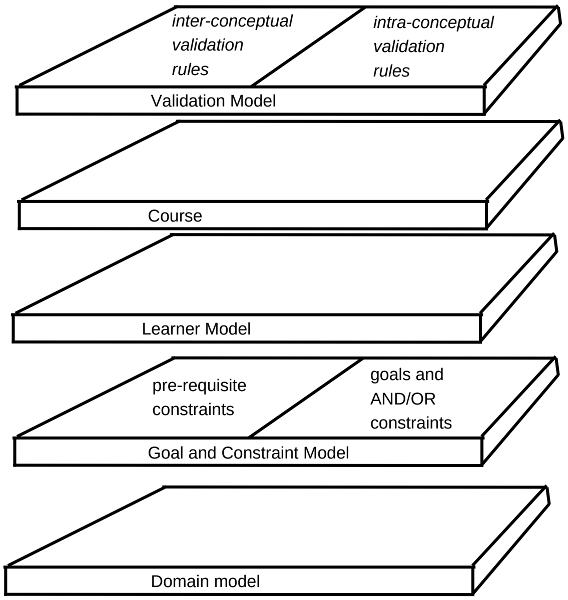


Figure 3: Course Construction Layered Aspects

the course pre-requisites). The courseware layer represents the constructed courseware. 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).

4.4 Explicit Representation of Courseware and its Construction Concerns

The information architecture provides a infrastructure which captures the courseware information we have discovered. This subsection looks at how some of the model layers in the information architecture can be explicitly defined.

Our initial investigation looked at using semantic web service technologies to validate LO compositions as we noted similarities between LOs and web services. We found much of what Semantic Web Service technologies provide for to be redundant in representing courseware such as, defining complex message flow, service choreography and orchestration, while at the same time it is difficult to instill pedagogical information into the Semantic Web Service standards.

Our investigation started by looking at explicit modeling of courseware as a way to represent the compositional aspect of courseware instructional logic with regard to LOs. UML2 is a robust and versatile modeling language primarily used in software system design, which allows for extensions to specialised domains through its profile mechanism, and through manipulation of its metamodel. The development of a UML2 profile for courseware definitions has been examined by the author in [18]. UML2 provides for extra precision and constraints to be placed on models using the Object Constraint Language (OCL) [22]. OCL is a formal language that allows a modeler to describe variant conditions which must hold in a UML2 model. We found UML2 had the facility to provide an accurate representation of courseware.

Software models can be semantically-enhanced using ontologies allowing for the representation of conceptual information such as domain information. Adding semantics to models allows for more accurate courseware validation aspect modeling. In [23, 20, 19], the use of ontologies in addressing the information needs of a course is examined by the author, this investigation looked at allowing Model Driven Development technologies to automate the transformation of course models into a course specification binding.

We found that semantic-enhancement of models allowed the representation of a wide variety of information in the context of a course. A course model could be linked to an ontology which represented a course domain. Various other constraints on the domain ontology could be used to represent, goal conditions, course pre-requisites, and pedagogical strategy details.

4.5 Courseware Validation Strategy

Using CAVaM we can hypothesize about how validation will take place. The architecture we have presented splits pedagogical validation into two parts, validating pedagogical strategy and validating pedagogical rules. Pedagogical strategy is defined at the domain and constraint model, where the course creator expresses high level conceptual constraints and the conceptual goal of a courseware on the underlying domain model. Pedagogical rules are expressed in the validation model where the course creator expresses inappropriate courseware attributes.

The next section demonstrates an algormathic approach to one type of pedagogical strategy validation, validating LO conceptual pre-requisite constraints.

4.5.1 Validating Course Pre-requisites

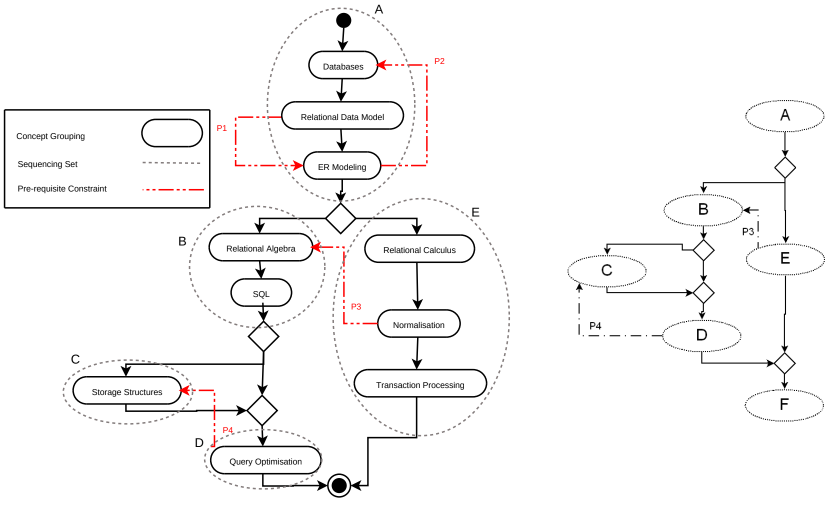
In this section we demonstrate our approach to one type of pedagogical strategy validation, pre-requisite constraint validation, as specified in the goal and constraint model. Pre-requisite validation aims to verify that the learner has any needed pre-requisite knowledge needed for a course element. In validating courseware pre-requisites, we classify the pre-requisite constraints into categories. These categories are "Pre-requisite verified", "Minor ordering error" - simple sequencing fix required, "Warning" - possible for the learner to miss some needed course material, "Error" - learner cannot view pre-requisite material due to the sequencing constraints or the pre-requisite material covered in the course.

To demonstrate how pre-requisite constraint validation will work and to demonstrate its value we will use a case study course.

Fig. 4(a) outlines the databases course model we will use for our case study. The databases course model has been divided into concept groups (black, solid ellipses), the name of each concept group indicates the name of the concept that the concept group refers to in the domain model. The dashed arrows, labeled Pχ, indicate pre-requisite constraints between concept groupings, which are derived from pre-requisite constraints between concepts in the goal and constraint model.

Fig. 4(a) specifies four pre-requisites for the database course. The directional pre-requisite arrow points to the concept grouping which teaches a pre-requisite concept, therefore according to P1 "ER modelling" must be understood before the "Relation Data Model".

Concept groupings in the course are further grouped into sequencing sets - sets of concept groups which are not linear (i.e. contain no choices in learner paths). We use letters to denote each sequencing set.



(a) (b)

Figure 4: (a) Database course model with pre-requisites (b) Course divided into sequencing sets

The algorithm firstly locates any pre-requisite constraint where the pre-requisite's source and target are in the same sequencing set. In this case P2 and P1 are identified as such constraints (source and target are in set A). The order of the concept groupings are checked where the target of the pre-requisite must be encountered before its source. P2 is satisfied, but P1 is not valid as the source of the pre-requisite is encountered before the target. The P1 constraint violation is classified as causing a "Minor ordering error", while P2 is classified as "Pre-requisite verified".

Pre-requisite constraints can be represented at the sequencing set level, where pre-requisites between concept groupings in different sequencing sets are represented as pre-requisites between sequencing sets. Fig. 4(b) depicts the sequencing sets derived from the database course outlined in Fig. 4(a). We also show two pre-requisites from the database course which are concerned with concept groupings from two different sequencing sets, P4 and P3. Pre-requisites, which have not been classified at this stage either fall into the "Warning" category (when there is a possibility through learner choice that pre-requisite concept LOs will be by-passed) or "Error" category (where it is not possible for the learner to encounter the sequencing set which contains LOs that teach the pre-requisite concept before the learner reaches the sequencing set which requires it).

When the algorithm checks P4, the target sequencing set is first located (set C). The course ordering constraints are then traversed (every path from set C) in order to find the source sequencing set of the pre-requisite, sequencing set D. In this case set D is found, this means that the pre-requisite constraint could be respected, although there is a possibility that the learner may violate this pre-requisite, depending on the learner's individual path through the courseware. P4 is classified in the "Warning" category as it is possible for the constraint to be satisfied but is subject to learner path choice. If the source of the pre-requisite is not found, which is the case for P3 we can conclude that the pre-requisite will never be seen by the learner and is classified as causing a sequencing "Error", and is categorised accordingly.

4.5.2 Pedagogical Rules Validation

Validation rules can be captured using a logics-based rule language such as JESS. A JESS engine can be used to then validate a given course against the validation rules specified in JESS. The code segment below demonstrates a rule which specifies each concept in a courses domain model must have a LO of type lecture associated with it.

```
(defquery findConceptsLOs    //QUERY: find the LOs associated with the given concept
  (declare (variables ?c))
  (concept-to-lo (concept ?c) (lo ?lo)))

(defquery findLoResourceType  //QUERY: Get the LOs with the given resource type
  (declare (variables ?x ?resourceType))
  (lom (id ?x) (resource_type ?resourceType)))


(foreach ?c ?concepts-in-dm   // for each concept in domain model
  (bind ?list (new java.util.ArrayList))
  (bind ?los-in-concept (new java.util.ArrayList))
  (bind ?concept-ok FALSE)
  (bind ?itr(run-query findConceptsLOs ?c)) //get LOs for concept
  (while(?itr hasNext)  //for each LO
    (bind ?token (call ?itr next))
    (bind ?fact (call ?token fact 1))
    (bind ?lo (fact-slot-value ?fact lo))
    (?list add ?lo) // add LO to list
  )
  (foreach ?l ?list  //for each LO in list
    (bind ?itr2(run-query findLoResourceType ?l lecture))  //does LO have resource type lecture
    (if (?itr2 hasNext) then
      (bind ?concept-ok TRUE) //if it does then concept ok
    )
  )
  (if (not ?concept-ok) then //if the ?concept-ok variable has not been changed - no lecture type
    (printout t "CONCEPT "?c" HAS NO LECTURE" crlf)
  )

)
```

5 Future Work

At the outset of this report we outlined our research question and the research problems which must be addressed in presenting a solution to our research question (section 2). In section 4 we have outlined the research problems we have investigated and outlined proposed solutions to the problem. In this section we outline the envisioned research problems we will address in the future, the challenges they present and possible approaches to addressing them. We also present our proposed scheduling for our future work.

5.1 Integration with Existing Technology-Enhanced Learning Course Specifications

Our courseware validation architecture is not coupled with any course packaging specification, as the specifications which define a course are quite immature and in constant flux. Our architecture has its own native representation of courseware.

In order for courseware validation to make an impact on existing course construction methodologies, courseware validation must slot into the existing methodology. To do this we must examine the current state of the art in course packaging specifications [1, 15] and allow for translation from the specification to our validation architecture. This is not a trivial task as the courseware interoperability specifications are based on different conceptual designs.

5.2 Proof of Concept

In order to verify the architecture which we have outlined in this document is viable, we will develop a proof of concept prototype application based on the courseware validation architecture. For this will employ a iterative development cycle. The results gained from each development cycle will be documented and used to influence further development cycles.

5.3 Evaluation of Courseware Validation Architecture

The role of the architecture we have outlined in this document is to find problems in a course. In finding problems in a course, we want the problems to be real problems, and we would also like to find as many problems in the course as possible (if not them all). In this way the role our architecture is similar to that Information Retrieval (IR) performs in information systems.

IR and the evaluation of IR systems is a mature discipline [17]. Metrics which have been developed for IR could be used for evaluating course validation. In this section we will examine these metrics and outline how they can be applied to course validation.

5.3.1 Evaluation Approach

Two main approaches can be taken to the evaluation, that of glass box evaluation, where every component in a system is assessed, and black box evaluation, where the system is tested as a whole. With regard to black box evaluation there are two dimensions, system-oriented evaluation and user-oriented evaluation. Both these approaches must be examined with respect to the work presented here. At a black box evaluation level we can evaluate the performance of the system when compared with similar systems and we can also look at user-oriented issues such as usability issues. Using a glass box evaluation methodology for evaluation we can examine the components that come together to allow for courseware validation, this allows us to evaluate the various layers of the validation architecture separately, the goal and constraints layer can be evaluated separately to the validation layer.

Within each approach are various variables which are subject to change. These include the course complexity, validation strategy. Courses can range from simple sequential courses with no branching points to highly complex and dynamic courses which adapt to a learners various needs. Validation rules may also vary in their complexity. Each of these variables must be evaluated separately, looking at the effect of the variable on the various evaluation metrics outlined.

5.3.2 Information Retrieval based Performance Measures

The classic IR performance metrics are recall and precision [17]. Recall evaluates the percentage of relevant documents from a particular database. The evaluation of recall is costly as it requires a manual assessment of what are deemed as relevant documents. Precision relates to the relevance of the documents returned. A highly precise IR system will just return highly relevant documents.

In the course validation context we have presented here, recall and precision can be used to evaluate the percentage of problems found in a course (recall) and the accuracy of problems found (precision) - is the problem found actually a problem.

The traditional trade-off between recall and precision found in all IR systems must also be examined and plotted on a ROC graph. From this we hope to get various metrics such as Mean Average Precision to evaluate the effects of varying evaluation variables mentioned in the previous section.

5.3.3 Performance Measures

Another point of evaluation is to look at how well the architecture performs in varying contexts, this will evaluate the efficiency our architecture performs, while the IR evaluation methodologies will evaluate the effectivness of the architecture. This is an important issue as course creators will not use a system which has unacceptable wait times.

The evaluation will look at a variety of courses, with varying size and complexity. Evaluation will also vary from simple pedagogical validation rules to more complex pedagogical strategy rules. The wait times will be evaluated for each of these. To put the times in context we will evaluate with course creators what they would regard as unacceptable wait times.

To identify areas for future work and improvement we will also evaluate each of the courseware validation architecture components separately, for example evaluating the course sequencing algorithm discussed in section 4.5.1 on its own.

5.4 Projected Schedule

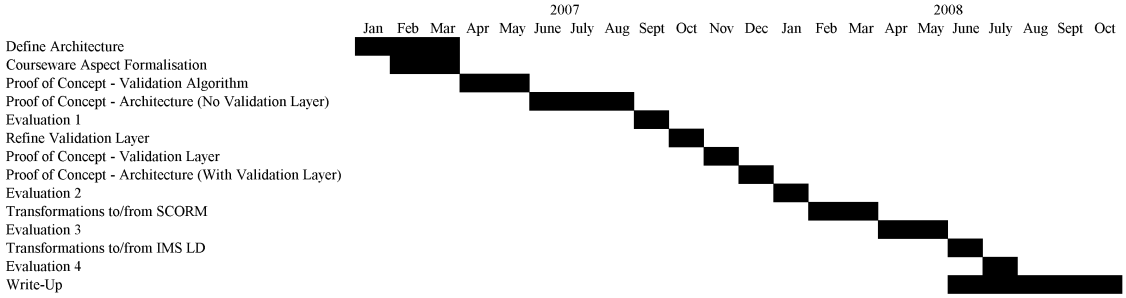


Figure 5: Projected Project Schedule

A project schedule in the form of a Gantt chart can be seen in figure 5. As shown in the Gantt chart at present we are still formulating and defining the courseware validation aspects and a validation architecture. Once we have this formulated we will develop a proof of concept program to ensure our assumptions about the validation algorithm we have defined in this report are true. Once we have done this we can go on to develop a proof of concept for the architecture we have defined in this report. The validation layer will be absent at this stage so that problems in the architecture can be easily isolated.

We will then do our first round of evaluation, to see how our architecture performs. The evaluation will be documented. We will then add a validation layer to the prototype and evaluate the effects it has on validation.

After this we will integrate our validation framework into the current state of the art in TEL, by integrating the framework with the SCORM standard, after which we evaluate the integration. We then do the same for the IMS LD standard. We integrate SCORM first as it is the simpler of the two TEL specifications.

6 Conclusions

In this report we have motivated the need for courseware validation, and presented a research question to that effect. From our research question we have been able to specify research problems and objectives. We have used these problems and objectives as a guide for our research.

To date we have identified the courseware validation aspects, which are is the information needed for courseware validation. We have placed our research in the context of courseware authoring as a whole, which outlined the need for correction after coureware pedagogical problems are found and diagnosed. We then looked at the representation of the various courseware validation aspects and bringing these aspects together under the umbrella of a courseware validation architecture which extends the state of the art in AEH.

Our most recent work in this area looks at how the validation of pedagogical strategy and pedagogical rules can take place.

In our future work we have noted the need to inter operate with the TEL specifications. We also wish to design and develop a proof of concept application to test our course validation theories. We will then use our proof of concept as a vehicle for evaluating our approach to 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, A. Martelli, V. Patti, and L. Torasso. Verifying the compliance of personalized curricula to curricula models in the semantic web. In *Proceeding of the Semantic Web Personalization Workshop at the Third European Semantic Web Conference (ESWC2006).* Springer-Verlag LNCS Series, 12th June 2006.

[3] M. Baldoni, C. Baroglio, and V. Patti. Web-Based Adaptive Tutoring: An Approach Based on Logic Agents and Reasoning about Actions. *Artificial Intelligence Review*, 22:3–39, 2004.

[4] 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.

[5] P. Brusilovsky and J. Vassileva. Course sequencing techniques for large-scale web-based education. *International Journal Continuing Engineering Education adn Lifelong Learning*, 13(1/2):75–94, 2003.

[6] A. Cristea and C. Stewart. *Web-Based Intelligent e-Learning Systems: Technologies and Applications*, chapter Automatic Authoring of Adaptive Educational Hypermedia . IDEA Publishing Group, 2005.

[7] 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.

[8] A. I. Cristea, D. Smits, and P. de Bra. Writing MOT, Reading AHA! - converting between an authoring and a delivery system for adaptive educational hypermedia. In *Proceedings of The Third International Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia at AIED05*, June 19th 2003.

[9] A. I. Cristea and M. Verschoor. The LAG Grammar for Authoring the Adaptive Web . In *Proceedings of The International Conference on Information Technology: Coding and Computing (ITCC'04) Volume 1*, pages 382–386, April 5th-7th 2004.

[10] D. Dagger. *Personalised eLearning Development Environments*. PhD thesis, University of Dublin, 2006.

[11] P. DeBra and L. Calvi. Aha: a generic adaptive hypermedia system. In *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia*, http://wwwis.win.tue.nl/ah98/Proceedings.html, 20-24th June 1998. Eindhoven University of Technology.

[12] J. Eklund and P. Brusilovsky. InterBook: An Adaptive Tutoring System. *UniServe Science News*, 12:8–13, 1999.

[13] R. Gagne, W. Wager, K. Golas, and J. Keller. *Principles of Instructional Design*. Wadsworth, California, USA, 5th edition, 2005.

[14] S.-C. Hu. Application of the UML in modeling SCORM-conformant Contents. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT03)*. IEEE Computer Society, 2005.

[15] 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.

[16] IEEE Learning Technology Standards Committee. LTSC WG12:Learning Object Metadata, 2002.

[17] W. Kraaij. *Variations on Language Modeling for Information Retrieval*. PhD thesis, University of Twente, 2004.

[18] M. Melia, R. Barrett, and C. Pahl. A Model-Based Approach to SCORM Sequencing. In *Proceeding of the Sixth Annual Irish Educational Technology User's Conference (EdTech06) - Research Track*. ILTA, 2006.

[19] M. Melia and C. Pahl. Semantic Model-Driven Development of Leanring Technology Systems. In *Proceedings of the TInformation Technology and Telecommunications Conference IT&T2005*. TecNet, 2006.

[20] M. Melia and C. Pahl. Semantically-enabled Model Driven Course Development. In *Proceeding of the First European Conference on Technology Enhanced Learning (EC-TEL06) - Doctoral Consortium Session*. EC-TEL06 Workshop Proceedings, 1st-4th October 2006.

[21] C. of EU Rectors' Conferences and the Association of European Universities. The Bologna Declaration on the European space for Higher Education: an explanation. Technical report, 2000.

[22] OMG. OCL 2.0, 2003. OMG Final Adopted Specification.

[23] C. Pahl and M. Melia. Semantic Modelling of Learning Objects and Instruction. In *Proceeding of the First European Conference on Technology Enhanced Learning (EC-TEL 2006)*. Springer-Verlag LNCS, October 1st-4th 2006.

[24] G. Paquette, M. Leonard, K. Lundren-Cayrol, S. Mihaila, and D. Gareau. Learning design based on graphical knowledge-modelling. *Journal of Educational Technology and Society*, 9(1):97–112, 2006.

[25] 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.

[26] J. W. Samples. The pedagogy of technology - our next frontier? *Connexions*, 14(2):4–5, 2002.

[27] J.-M. Su, S.-S. Tseng, J.-F. Weng, K.-T. Chen, Y.-L. Liu, and Y.-T. Tsai. An Object Based Authoring Tool for Creating SCORM Compliant Course. In *19th International Conference on Advanced Information Networking and Applications*, volume 1, pages 209–214. IEEE, 2005.

[28] The RELOAD Project. The RELOAD Metadata and Content Packaging Editor. Available from: http://www.reload.ac.uk/editor.html.

[29] D. A. Wiley. *The Instructional use of Learning Objects*, chapter Connecting Learning Objects to Instructional Design Theory: A definition, a methaphor and a taxonomy. Association for Educational Communications and Technology, 2001.

[30] J.-T. D. Yang, Y. Pao-Tan, and W. C. Chen. An ontology-based course editor (obce) for scorm-compliant learning objects. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies*. IEEE Computer Society, Aug 30th - Sept 1 2004. Looks at ways of using ontologies as the starting point in leanring object creation, very similar idea to MIKAEL.