Ontology Evolution for Learning Content Management Systems

Claus Pahl, Muhammad Javed, Yalemisew M. Abgaz
Centre for Next Generation Localization (CNGL),
School of Computing, Dublin City University, Dublin 9, Ireland
{cpahl|mjaved|yabgaz}@computing.dcu.ie

Abstract: Learning content management systems have become common tools to support the development and delivery of digital learning content for many years. We investigate here how semantic annotation can support the management of change and evolution of learning content. We introduce a semantic information model that supports multi-layered learning content structures, based on which an ontology-based evolution framework is developed. Ontology-enhanced traceability approach is the solution.

Introduction

Learning content management has received some attention given that a broad range of content formats are supported by Web technologies. Learning content management systems (LCMS) with different types of structured content (e.g. Web pages, animations, software) require specific solutions to deal with the different representations (Kenny and Pahl, 2005). With many years in operation, LCMS become subject to change and evolution. We propose an ontology-based annotation of learning content in LCMS to control and manage content and LCMS evolution. Ontologies allow us to use change traces for change and evolution management (Gray et al., 2006). Ontology-based content management uses a trace model between the ontologies as abstract models and learning content artefacts as the key technique to identify the impact of change and to predict its implications. Our central contributions are

- a semantic information model for annotated learning content,
- a framework for change management with a trace model at the core.

Our approach adds ontological layers on top of a content model, which allows us to reason about content change using the ontological model. We add traces as processable elements in a connecting layer – resulting in an ontology-based enhanced trace model for content and ontology change activities. These techniques result in a significant enhancement for LCMS and semantic content change and evolution management.

The paper is organised as follows. We start with a discussion of learning content management applications and their evolution (based on courseware system we have run and maintained for 10 years as a Web-based course support system) in the next section, i.e. a semantically annotated LCMS architecture and evolution scenarios are introduced. In the following section, we discuss applications of the information model for content change management. Then, we define our change and evolution framework. We discuss our approach after that, including related work and end with some conclusions.

Learning Content Management Systems Evolution

Content-Centric Learner Management Systems

Learner management systems are software applications that can be accessed to process learning content. We take a content-centric perspective on these learner management systems here. We consider here learning content as any digital information – whether static or dynamically created, whether executable or not. Structured and unstructured text, however, will still be the central artefact type. We use a layered information model to structure the learning content management systems, in which we distinguish between learning content components (such lecture material, tutorial or lab activities), the content management (consisting of access mechanisms based on navigation and search features) and ontologies (to annotate content in terms of the subject domain, but also educational aspects) for our case study application. The ontologies support a range of multi-facetted models on the information aspects.

While the role of the semantic information model is looked at in the context of change and evolution, it also used to support content management activities. This is important, as an investment into semantic annotation is beneficial for a range of activities. We can consider access to content through queries and content retrieval. An ontology-based solution allows the adaptive, contextualised use of information architecture for query processing and information retrieval (Cristea et al, 2007). A query often is of the form "I want to learn about models …". A learning object descriptor (educational activity and learning subject) can be inferred.

Evolution and Change in LCMS

At the centre of our case study is a content management system for an interactive, multimodal learning environment – the Interactive Database Learning Environment IDLE (Pahl et al., 2004). We have in particular investigated the content access layer for the application system. The IDLE system is a database learning environment (the subject is databases) for a particular course (the format is an undergraduate module) that provides content in the form lectures, tutorials and labs. Different media types (from text to animations) are used. Based on the ontologies, the learning content management manages content and provides access in different forms (content-centric table of content, assessment elements or organisational information). This system has supported courses for more than 10 years to support lectures, tutorials and labs and has been used by more than 1000 students.

During this period, the system has been subject to extensive change and evolution. The content has been continuously corrected, improved and extended. The delivery support infrastructure has evolved over the years to incorporate significant new features. In (Pahl, 2003), we have presented a change classification scheme that we use here to summarise the changes in more detail in the following two categories:

- *Content* [changes were continuous, but moderate in extent]
  - *Subject evolution*: The course subject itself evolves – driven by external factors. Some additions to content have been made to reflect change in the technological landscape.
  - *Content improvement*: Content is changed in order to improve the material in a planned process. Minor corrections to address presentation and technical issues have been carried out.
- *Format* [the course had to be reorganised – shortened in the overall weight – as the most important change in this category, which has caused changes across all sections.]
  - *Staff*: Changes relating to educators, course developers, or technical support staff. In total 4 lecturers – with 3 of them familiar (of which 2 permanent academics and 1 replacement instructor) and 1 unfamiliar with the underlying technology) were involved. Staff training has become an issue as a result.
  - *Students*: The student body changes in terms of numbers, qualifications, or mode of learning. The learner profile changed slightly (no effect after one significant increase of numbers affecting technical infrastructure support).
  - *Timetabling*: Changes related to where and when a course takes place. Part-time evening delivery and extended content access at that time has been a problem for a period.
  - *Syllabus*: Content and organisation of the course content can change. Content has been shortened as part of a degree restructuring (see above).
  - *Curriculum*: Organisational needs require changes in level, extent, or prerequisites of courses. Content was changed caused by a syllabus update (see above).

While some of these seem manageable, the impact over longer periods is noticeable. In the given context of database principles, the subject domain is relatively stable, but in more volatile contexts, significantly more severe change can be expected. As pointed out in (Pahl, 2003), the cost of change and evolution can be unsustainable. As IDLE has been semantically annotated based on domain and infrastructure ontologies, we explain here how the aim of predicting and controlling the impact of change and evolution can be supported using ontology annotation.

Ontology-Annotated Learning Content Management Systems

Central activities of ontology-based content information modelling are content identification (content component of a courseware application), LCMS modelling, and layering of application components (dependencies and references), i.e. relating previously modelled separate application models. The activities shall be applied to the IDLE system. *Learning objects* forms the core of the application and consists of the learning system content components, which are accessed through the user interface. The application system is IDLE, which consists of application-specific components such as lecture system, tutorial system and lab system (learning components / objects). In this learning object context, a second category of components can be identified. Learning environments consist of generic content elements: Web pages (media type used in lectures) as hypertext content, animations (media type used in tutorials) as dynamic content and exercises (used on labs) as interactive content. *Access and Organisational Files* implement the access system by providing different ways to access information in order to support some educational activity: the table of content organises content primarily to support the central learning activity; assessment is another central element. The learning activity is about learning the subject, guided by table of contents (narrative), the assessment activity is about determining the level of understanding (reflection), and self-organisation as an activity comprising of supportive activities to manage learning activities for the learner.

An ontology-based information architecture modelling (Boyce et al., 2007) aims at ontology identification based on the identified core artefacts (usually the latter serve as the default structure), ontology modelling/construction (cross-ontology hierarchies that take in concepts from all central artefacts) and ontology mapping and association describing the function of each of the ontologies and the corresponding dependencies. We can formalise the content conceptual model based on these observations. An overall *hierarchy of concepts* (across different artefacts) captures central concepts and their (mainly taxonomical) hierarchy. We use the inclusion symbol here to denote a hierarchical relationship to relate concepts already identified as components earlier on:

system ⊇ learning object ⊇ query ⊇ topic (types: concept, reference, task) ⊇ activity ⊇ task

Ontologies and the *ontology hierarchy* emerge from the discussion of the artefacts earlier on. We start bottom-up here (see Fig. 1) with the format, which determines and organises educational activities, which in turn accesses (through the content management) and guides the subject domain-specific learning objects. The content management focuses on learning object components in order to management and locate content, we subject domain components from the function they realise in the educational domain.

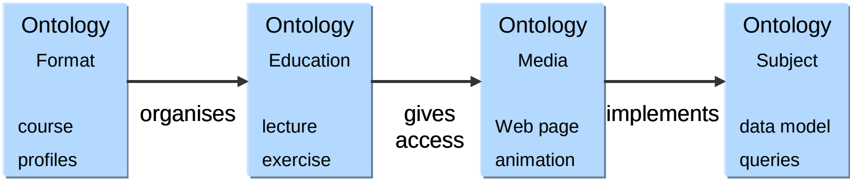

Fig. 1. Ontology Hierarchy.

Separate ontologies can be identified. For this learning content domain, three specific relationships have been introduced (e.g. organises) in order to clarify the roles that individual ontologies play in relation to each other. This will become important later when for instance change impacts have to be determined.

A trace model is the integrating layer between ontological conceptual model (top) and content architectural model (bottom), consisting of a model definition (based on content artefacts and ontologies) creating an explicit link between the layers, that can become the primary object of manipulation for content access and management activities, and the consideration of ontology-based content annotation in trace generation. *Instances* in ontology terms are components of the artefacts. Examples from two artefacts are learning object features: a flash sequence is an animation, which in turn is a dynamic content component; or an access file: an entry in a table of contents is a subject domain element, an assignment is an assessment instance. Access files internally make references to concepts from other ontologies such as the media and domain ontologies. These annotations need to be clearly identified and linked to the content concepts as annotations. The *annotation of content* (access files and content management tables – and also learning objects) through ontology instances links components of artefacts to corresponding concepts. For instance, exercise and multiple choice questions are assessment access components that refer to an educational activity. These annotations need to be made explicit as traces. We can use an example from an access file. These include complex phrases based on basic ontological concepts. The expression "multiple-choice-question assessment on data model" is a learning activity instance that refers to MCQ, assessment, and data model as educational and subject concept instances.

Change and Evolution

Change will ultimately affect all artefacts – access files being updated, learning objects being improved. The latter will cause knock-on effects on the access files and also the content management (which is the ontology-based trace model). The trace model itself can be used to identify change impacts locally and across artefacts.

We distinguish two categories of changes – changes to the content artefacts (learning objects and content management infrastructure artefacts) and changes to the ontologies as the knowledge on top of the artefact layer (Gruhn et al., 1995).

- *Ontological Changes* (Plessers et al., 2007) can include adding database language above query language in the taxonomical hierarchy of subject concepts, or Audio could be added below Media Types. Changing query properties in the content management ontologies, for instance tutorials and labs could be 'active learning' as a newly introduced subsuming concept or online submission/correction could be classified as an ‘assessment’. Ontological change could reflect changes in the general terminology used. The trace model would indicate further artefacts that are impacted. The trace model acts as a dependency relationship.

- *Content Changes* can affect any of the artefacts. Of particular interest here are cascading impacts on other artefacts. 1) Learning object changes can have an impact on access files: a new learning object feature (e.g. new media format being used) could result in a new media type (in media ontology) being added, a concept (subject domain ontology), a new format elements (new type of assessment) result in a new educational activity being added. This list is presented in decreasing likelihood of change to the corresponding ontology. 2) Content Management changes could be the addition of a new format type, requiring a new access types in the access ontology, or a new learning object location/identification mechanism, requiring a new ID to be added to the access ontology. 3) Access changes usually do not have structural effects on the ontology, however instance-level changes to the application or software ontology are possible.

Again, the trace model defines the dependency relation that allows determining change impact (Gray et al., 2006).

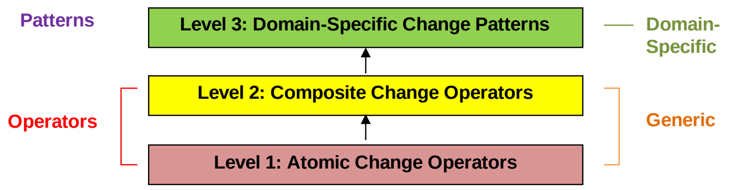

Fig. 2. Layered Change and Evolution Framework.

Based on our observation of common changes in all versions of the ontologies, we defined a framework of change operators and patterns (Fig. 2). The first two layers are based on generic and structural change operators. The next layers covers domain-specific change patterns.

- Level One Change Operators - elementary changes which are atomic tasks: These change operators are the elementary operations used to perform a single Add/Delete task on a single targeted entity. A single operator performs a single task that can add a concept, add a property or delete a concept, etc. on the constituent components of the ontology.
- Level Two Change Operators - aggregated changes to represent composite, complex tasks: Many evolution tasks cannot be done by a single atomic operation. These change operators are identified by grouping atomic operations of level one to perform a composite task. For example, to delete a single concept “chapter” in a course ontology, removing the concept from the concept hierarchy is not sufficient. Before we remove the concept, we have to remove it from the domain and the range of properties like “hasChapter” or “contains” that are linked to it. In addition, we need to either remove its subconcepts or link them to the parent concept.
- Level Three Change Operators – domain-specific change patterns: The changes at a higher level of granularity, which are frequent in a domain, can be represented as domain-specific patterns - which are often neglected by the lower-level compositional change operators. Domain-specific perspective links the structural changes to the aspects represented in domain ontologies. In order to execute a single domain-specific change, operations at level one and two are used. An example is the introduction of a new chapter, which involves the definition of the corresponding domain concepts and the addition of the chapter to access structures (e.g. table of contents).

Ontological or content changes have effects on the artefacts of learning content management systems. We define effects as a consequential change of the state of the ontology or artefacts of the content due to the application of a change operation on one or more of the elements in the ontology. Effect determination process is a process that iidentifies, analyses and determines the effects of a change before the change become effective. The core contribution of effect determination process is the identification of the effects of a requested ontological and content change on a given artefact. Fig 3 depicts the overall architecture of effect determination process.

We categorized effects in to structural effect and semantic effects. Structural effect is an effect that occurs on the structural relationship between the elements of the ontology. Structural effects are possible consequences on the taxonomy of the ontology due to a structural change. This includes the linkages between concepts and the content via the trace model. Semantic effect is an effect that occurs on the interpretation of the ontology and its elements. Semantic impacts are possible inconsistencies and invalidities that arise for the interpretation of the ontology due to structural changes. Beyond structure, constraints/restrictions such as size restrictions on content (e.g. wrt. the number of chapter instances) can be validated.

The overall effect determination process uses the following basic parameters. The type of change requested (addition, deletion) the target element (concept, property, axiom), the number and type of dependent elements on the target entity (subclass, equivalent class) the consistency rule violated (identity invariant) and their severity on the integrity (Abgaz et al., 2010). We further considered the number of instances invalidated (lose their interpretation against the ontology) whenever a change is implemented.

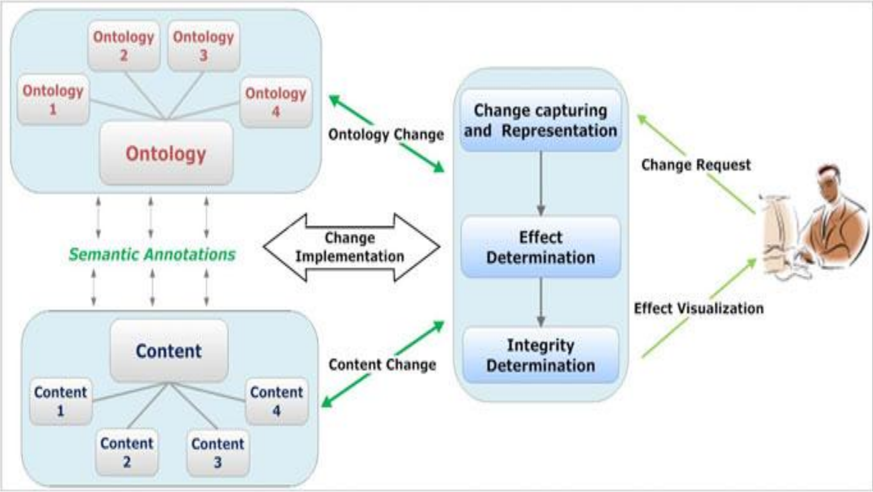

Fig. 3. Architecture of effect determination process.

We have identified and categorized change operations based on their effects, identifying parameters (type of operation and target element) that play significant role in determination of effects and categorization of the change operations based on the severity of the effects. We identified factors for determining severity of the effects like the cascaded operations, the time required (number of operations) and the human involvement to resolve complex choices. From this, an overall cost indication of proposed changes can be calculated in advance of the implementation of change. The effect determination process identifies the possible effects ontological and content changes have on the artefacts of learning content management systems, presented to a content manager before permanent implementation of the change.

Discussion

The core of our solution is our information architecture – comprising of the modelling process definition and the formalised model. Both the process and the model have been empirically validated by modelling rich educational systems as the one we have used as our case study. The IDLE management (Kenny et al., 2005, Pahl et al., 2004) has caused difficulties and has demonstrated the advantages of modelling. Particularly, adaptive elements of IDLE have benefits (Holohan et al., 2006). Beyond the support in construction and authoring activities, models can have a wider impact. For instance, (Pahl et al., 2004) discusses the formative evaluation (data mining and usage analysis) as part of a structured change management strategy. Models help to interpret mining results regarding learner behaviour.

Content management has been widely addressed. One specific area is ontology-based change and evolution. Most work has focussed on activities and operators on ontologies (Javed et al., 2009). Qin and Atluri (2009) also make a first step towards content; they investigate the impact of change on the validity of instances. Their work also comprises a formal model, but does not include the extension to semantically annotated content and instances as non-atomic, internally structured artefact.

The main contributions here in this investigation towards this aim are the formalisation of an information model and the support of the LCMS change and evolution process. An ontology-based solution for the traceability model is the key factor. The ultimate benefit arises from the use of traceability properties, providing the support for change management activities: targeted, adaptable access having the benefit of accuracy of navigation and retrieval and change management having the benefit of higher automation degrees and predictability.

While the validity of the data models and the adequacy of the process activities have been demonstrated, a number of aspects shall be investigated in the future. We will not only look at different formats, but also the multi-linguality of the Web as a dimension in our content management approach.

Acknowledgement

This material is based upon works supported by the Science Foundation Ireland under Grant No. 07/CE/I1142.

References

Boyce, S. and Pahl, C. (2007). The Development of Subject Domain Ontologies for Educational Technology Systems. Journal of Educational Technology and Society (ETS). 10(3):275-288.

Brusilovsky, P. (2000). Concept-based courseware engineering for large scale web-based education. In Davies, G. and Owen, C., editors, Proceedings of WebNet2000, World Conference of the WWW and Internet, 69–74, AACE.

Cristea, A. and Aroyo, L. (2002). Adaptive authoring of adaptive educational hypermedia. In Proceedings of the 2nd International Conference on Adaptive Hypermedia (AH2002), 122–132, Springer-Verlag.

Dagger, D. (2006). Personalised eLearning Development Environments. PhD thesis, University of Dublin.

Gray, J., Lin, Y., and Zhang, J. (2006). Automating Change Evolution in Model-Driven Engineering. Computer 39(2) 51

Gruhn, V., Pahl, C., and Wever, M. (1995). Data Model Evolution as Basis of Business Process Management. In 14th International Conference on Object-Oriented and Entity Relationship Modelling O-O ER'95, LNCS 1021.

Holohan, E., McMullen, D., Melia, M., and Pahl, C. (2006). The generation of e-learning exercise problems from subject ontologies. Proceedings Sixth International Conference on Advanced Learning Technologies, ICALT'2006. Pages 967-969.

Javed, M., Abgaz, Y., and Pahl, C.. (2009). A Pattern-based Framework of Change Operators for Ontology Evolution. The 4th International Workshop on Ontology Content OnToContent 2009. Springer-Verlag. LNCS.

Javed, M., Abgaz, Y., and Pahl, C. (2012). Composite Ontology Change Operators and their Customizable Evolution Strategies. 2nd Joint Workshop on Knowledge Evolution and Ontology Dynamics EvoDyn 2012 - Collocated with the 11th International Semantic Web Conference (ISWC 2012). Boston, US.

Kenny, C. and Pahl, C. (2005). Automated tutoring for a database skills training environment. ACM SIGCSE Symposium 2005. St. Louis, US. Pages 58-64. ACM Press.

Kenny, C. and Pahl, C. (2009). Intelligent and adaptive tutoring for active learning and training environments. Interactive Learning Environments 17 (2), 181-195

Murray, S., Ryan, J. and Pahl, C. (2003). A tool-mediated cognitive apprenticeship approach for a computer engineering course. Proceedings 3rd IEEE International Conference on Advanced Learning Technologies, ICALT'2003.

Qin, L. and Atluri, V. (2009). Evaluating the validity of data instances against ontology evolution over the semantic web. Information and Software Technology. 51(1):83–97

Pahl, C. (2003). Evolution and Change in Web-based Teaching and Learning Environments. Computers & Education. 40(1): 99-114.

Pahl, C. (2002). An evaluation of scaffolding for virtual interactive tutorials
C Pahl. World Conference on E-Learning in Corporate, Government, Healthcare, and ...

Pahl, C., Giesecke, S. and Hasselbring, W. (2007). An ontology-based approach for modelling architectural styles. Proceedings European Conference on Software Architecture ECSA'2007. Pages 60-75.

Pahl, C., Barrett, R. and Kenny, C. (2004). Supporting Active Database Learning and Training through Interactive Multimedia. Intl Conference on Innovation and Technology in Computer Science Education ITiCSE'04. ACM.

Plessers, P., De Troyer, O. and Casteleyn, S. (2007). Understanding ontology evolution: A change detection approach. Web Semantics: Science, Services and Agents on the World Wide Web. 5(1):39–49