Using Schema Analysis for Feedback in Authoring Tools for Learning Environments

Harrie PASSIER & Johan JEURING
Faculty of Computer Science, Open University of the Netherlands
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
Email: harrie.passier@ou.nl & johan.jeuring@ou.nl

Abstract. Course material for electronic learning environments is often structured using schema languages. During the specification and development of course material, many mistakes can be made. We introduce schema-analysis as a technique to analyse structured documents, and to point out mistakes introduced by an author. With this technique we are able to produce valuable feedback.

Introduction

Electronic learning environments (LE's) are complex tools. Non-computer experts often write courses using such tools. Authoring tools have been developed to support the development of LE's. To improve the quality of LE's, an authoring tool should include mechanisms for checking the authored information on for example accuracy and consistency. Murray [7] mentions several such mechanisms. In this article we introduce schema-analysis as a technique to analyse course structure and domain ontology, which we represent by the languages IMS Learning Design (IMS LD) [5] and RDF. Using these flexible languages an author can easily make mistakes, which can be partly prevented by using templates. Some drawbacks of templates are loss of flexibility and problems with maintainability. With schema-analysis we maintain flexibility, are able to produce feedback when an author makes a mistake, and leave the author, as a didactic professional, free to accept or not accept the feedback information [3]. To show the technique at work we have defined and implemented six analyses to determine some quality aspects of a course: completeness, timely, recursiveness, correctness, synonyms and homonyms.

In this paper we briefly: explain what we mean with schemata, introduce the languages we use to represent them, describe in functional terms the analysis functions, and discuss some related work. A full paper and the (Haskell) code can be obtained from: http://www.ou.nl/info-alg-inf/Medewerkers/en_Passier.htm. The results presented in this paper are part of a project in which we investigate general feedback mechanisms to learners as well as to authors [8].

1. Schemata and representations

An ontology specifies the objects in a domain of interest together with their characteristics in terms of attributes, roles and relations. Using an ontology many aspects of a certain domain can be represented, for example categories and composition [9]. A composite object contains objects related to other objects using ' has_part' or 'uses' relations and has structure. Such a structure description is called a script or a *schema*. In this article we focus on schemata. We use RDF to represent a domain ontology. The basic building block of RDF is a triple: <resource, property, value>, which defined concepts and related concepts. We use IMS LD [5] to represent the structure of a course. In this paper we focus on the Activity-model of IMS LD. To be able to add more specific annotations to content and structure we introduce two new elements in the Extra-p element: *Definition* and *Example*. Furthermore, we introduce a new attribute *Educational-strategy* of the element Activity with two possible values: *Inductive* (definitions after examples) and *Deductive* (examples after definitions). Introducing such elements will make it possible to structurally analyse educational material. Listing 1 shows some relevant elements related to the activity-model together with the newly defined elements example and definition. The new elements and attribute are marked in bold.

```
<!ELEMENT Activity  %Activity-model; >
<!ATTLIST Activity
        …
        Educational-strategy (Inductive | Deductive) >
<!ENTITY %Activity-model "(Metadata?, …, Activity-description)" >
<!ELEMENT Activity-description  (Introduction?, What, How?, …, Feedback-description?) >
<!ELEMENT What  %Extra-p; >
<!ENTITY %Extra-p "(…| Figure | Audio | Emphasis | List | … | Example | Definition)*" >
```

Listing 1. Parts of the activity-model in IMS LD definition

The elements *example* and *definition* include a short description, the central concept and the related concepts.

2. Schema analysis to detect authoring problems

We perform two types of analyses: 1) the analysis of structural properties of a schema, for example the recursive property, and 2) the comparison of a schema with one or more other schemata, for example to test the correctness of a definition in a course against an ontology.

We have developed six analysis functions that can be used to signal possible mistakes:

Completeness – We distinguish three kinds of (in)completeness: (1) within a course, (2) within an domain ontology and (3) between a course and an domain ontology. If a concept is used in a course, for example in a definition or an example, it has to be defined elsewhere in the course. A course is complete if all concepts used appear in the set of defined concepts. Completeness can also be applied to an (domain) ontology, and between a course and an ontology. The first one checks whether all used concepts in the ontology are defined in the same ontology, the second one if all used concepts in a course are defined in the ontology.

Timely – A concept can be used before it is defined. This might not be an error if the author uses an inductive instead of a deductive strategy to teaching, but issuing a warning is probably helpful. Furthermore, there may be a large distance (measured for example in number of pages, characters or concepts) between the definition and the use of the concept, which is probably an error.

Recursive concepts – A concept can be defined in terms of itself. Recursive concepts are often not desirable. If a concept is recursive, there should be a base case that is not recursive. Recursive concepts may occur in a course as well as in an ontology.

Synonyms – Concepts with different names may have exactly the same definition. For example, concept *a*, with concept definition (*a*, [*c,d*]), and concept *b*, with concept definition (*b*, [*c,d*]), are synonyms.
Homonyms – A concept may have multiple, different definitions. If for example concept *a* has definitions (*a*, [*b,c*]) and (*a*, [*d,f*]), then these two definitions are homonyms.
Correctness – The concepts in a course should correspond to the same concepts in its domain ontology.

The implementation of these analysis techniques are based on mathematical results about fixed points[4].

3. Related work

Although many authors underline the necessity of feedback in authoring systems [1][3][7], we have found little literature about feedback and feedback generation in authoring systems.

Jin et al [6] describe an authoring system that uses ontology's enriched with axioms to produce feedback to an author. On the basis of the axioms the models developed can be verified. It is not clear how general the techniques are. Aroyo et al. [1][2][3] describe an ontology based authoring framework, which monitors the authoring process and prevents and solves inconsistencies and conflicting situations. They list five requirements for such environments. We think that our framework[8] satisfies these requirements and that schema analysis supports these requirements. Stojanovic et al present an approach for implementing eLearning scenarios using the semantic web technologies XML and RDF, and make use of ontology based descriptions of content, context and structure [10]. A high risk is observed that two authors express the same topic in different ways (homonyms). This problem is solved by integrating a domain lexicon in the ontology and defining mappings, expressed by the author itself, from terms of the domain vocabulary to their meaning defined by the ontology.

4. Conclusions

This paper discusses schema analysis as a general technique to analyse structural aspects of learning environment related material.

References

[1] Aroyo L., Dicheva, D., Authoring support in concept-based web information systems for educational applications, in Int. J. Cont. Engineering Education and Lifelong Learning, Vol. 14, No. 3, 2004.
[2] Aroyo L., Dicheva D., The new challenges for e-learning: The educational semantic web, Educational technology & Society, 7 (4), 59 – 69, 2004.
[3] Aroyo, L. Mizoguchi, R., Towards Evolutional authoring support systems, Journal of interactive learning research 15(4), 365-387, AACE, USA, 2004.
[4] Davey B., Priestly H, Introduction to lattices and order, 2e edition, Cambridge University Press, 2001.
[5] IMS LD: http://www.imsglobal.org/learningdesign/index.cfm.
[6] Jin L., Chen W., Hayashi Y., Ikeda M., Mizoguchi R., An ontology-aware authoring tool, Artificial intelligence in Education, IOS Press, 1999
[7] Murray, T., Authoring intelligent tutoring systems: An analysis of the state of the art. International Journal of AI in education, 10, 98 - 129, 1999.
[8] Passier, H., Jeuring, J., 2004. Ontology based feedback generation in design-oriented e-learning systems, Proceedings of the IADIS International Conferencee-Society 2004, Avila, Spain.
[9] Russell, S, Norvig, P., Artificial intelligence, A modern approach, Prentice Hall Int. editions, 1995.
[10] Stojanovic, L. Staab, S., Studer R., eLearning based on the semantic web, in WebNet 2001 – World conference on the www and internet, Orlando, Florida, USA, 2001