Gap Detection in Web-based Adaptive Educational Systems

Dominik Jednoralski[1], Erica Melis[1], Sergey Sosnovsky[1], and Carsten Ullrich[2]

[1] DFKI, Campus D3.2, D-66123 Saarbrücken, Germany
[2] Shanghai Jiao Tong University, 200030 Shanghai, China
dominik.jednoralski@dfki.de, melis@dfki.de, sosnovsky@gmail.com, ullrich_c@sjtu.edu.cn

Abstract. Content development for adaptive educational systems is known to be an error-prone task. Gaps can occur when the content is created, modified or when the context of its usage changes. This paper aims at improving the existing practises of learning content quality control in adaptive educational systems by automating detection and management of gaps. Several categories of gaps are identified to account for structural, linguistic, and semantic content inconsistencies in learning content collections. An effective filtering mechanism have been implemented in order to slice the collected gap data into categories that are relevant for the current authoring context. Evaluation of the developed tool demonstrates its utility.

1 Introduction

Authoring of high-quality content for adaptive educational systems has always been a expensive procedure requiring significant expertise and time investment [1]. The modern trends of knowledge representation and content modeling [2–5] have made this task even more elaborate. Nowadays, “ideal” learning content has to be designed and annotated with metadata in such that it is sharable and reusable by teachers and authors; easily discoverable and accessible by students; and ready for use in different adaptive educationanl systems and in different learning contexts. These requirements have one important practical implication for the learning content authoring process: the content creators need new kinds of authoring support tools at their disposal to ensure the high quality of the content.

We are trying to address this issue by proposing a rather new approach to learning content quality control based on the exhaustive verification of the content metadata. As a practical implementation of this approach, we describe a tool that helps authors to automatically detect several categories of gaps in the developed learning content.

Resolution of content gaps and inconsistencies is important for any e-learning system. As a category of users, students are very susceptible to the system’s failures and are rarely able to recover from those on their own. Erroneous instructional material presented to a student can lead to confusion, frustration, lack of motivation and, essentially, poor learning. This is even more crucial for adaptive educational environments, as they automatically sequence and present learning objects for an individual student, instead of following a manually crafted course. Consequently, they require a larger variety of learning objects to cover possible learning situations.

Newly developed learning content should be ready for sharing and reuse, as it greatly improves the effectiveness of content authoring, especially for adaptive systems [6]. However, developing reusable content introduces a new set of challenges from the perspective of learning gaps. Content developed for a certain category of students and a certain educational context can be reused in completely different settings (e.g. different students' background / major / culture / country, different adaptation strategy, etc.). To be fully reusable in another context, a learning object needs appropriate metadata annotations characterising its usage in this context. It is a difficult task for an author to manually ensure exhaustive metadata annotation of the learning content for all potential usage scenarios.

Many existing learning content collections have been developed and/or evolved as a collaborative effort of many authors (e.g. Ariadne[1], Curriki[2], and Connexions[3]). Collaborative and cooperative authoring introduces interesting opportunities to improve authoring practices of educational content. However, if several authors with different backgrounds develop a single content collection by extending and modifying each other's work, a source of potential content gaps is created. Gaps can be introduced when one is trying to contribute into learning object metadata, translation, or structuring. Manual control of inputs from multiple authors in order to detect this kind of gaps is not practical. For all those reasons, we have developed a gap-detection tool as a part of the authoring component of the ActiveMath adaptive learning environment [7].

2 Learning Content and Metadata in ActiveMath

ActiveMath uses a fine-grained knowledge representation to encode learning materials, based on OMDoc references and LOM representation standards [8] [4]. Each learning object corresponds to about one paragraph of text and has a type specifying its primary function. ActiveMath's learning objects can be divided into two main categories, namely fundamentals and auxiliaries. Fundamentals include general types of (mathematical) knowledge, such as definitions and theorems. Auxiliaries contain additional information about the fundamentals in the form of examples, exercises and texts.

In addition, metadata further describes pedagogical, organisational and semantic aspects of the learning objects. Exercises and examples have a difficulty level and train specific competencies. Auxiliaries always provide information about one or more fundamentals by means of the for relation. Authors can

[1] http://www.ariadne-eu.org/

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

[3] http://www.cnx.org

also encode pedagogical and domain-dependent relations between the learning objects.

Several ActiveMath components make use of this fine-grained knowledge representation. PAIGOS is a course generator that assembles sequences of learning objects based on the individual learning goals and taking into account their prior knowledge [12]; ActiveMath student model uses the relations to perform intelligent updating of the students' knowledge while they interact with the content [13]; the search component allows to discover learning objects based on their metadata [14].

3 Detection of Content Inconsistencies and Gaps

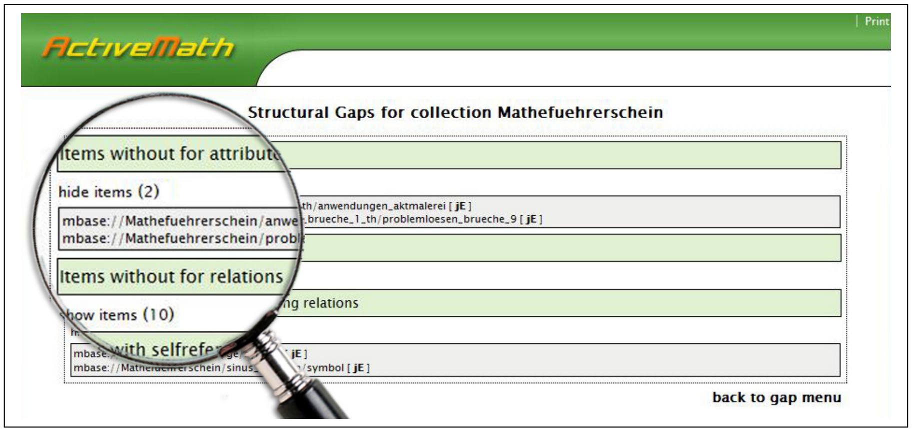
ActiveMath has a very rich metadata model, which allows it to implement effective learning strategies based on multiple parameters. Although, not every learning system is based on such a comprehensive metadata model, we believe, the gap detection problem is universal for the majority of adaptive educational systems, as most of them require metadata authoring to some extend. In order to better understand the sources of gaps, reasons of their occurrences, and the ways to deal with them, we have divided content gaps into three main categories: structural, linguistic and didactic.

3.1 Structural Gap Detection

There are three main sources of structural gaps:

1. The metadata or the content itself of the learning object is unrecoverably corrupted and cannot be read/restored/presented correctly by the educational system. This can be a result of a syntactic error. Such gaps should usually be caught during authoring; nevertheless, they stay undetected frequently.
2. A learning object lacks metadata connecting it to the rest of the educational material. For example, an exercise misses the relationship to the target concepts it helps to train.
3. A learning object with incorrect relational metadata. There can be multiple types of relations between learning objects within a content collection: prerequisite / outcome relations, links between exercises and concepts, relations specifying ordering of learning objectives. Inconsistencies in such relations are hard to find; however, detection of typical cases has been implemented (e.g. a learning object referring to itself or linking to a non-existing learning object).

In the case of ActiveMath, the syntactic correctness of the learning objects' metadata and content is verified by XML parsers when the learning objects are uploaded to the system. The gap detection tool takes care of recognizing and presenting to the author the former two kinds of structural gaps. Learning objects of certain types in ActiveMath have to be connected to other learning objects in order to form a coherent content collection, e.g. definitions have to be connected to symbols they define; proofs need a relation to theorems or lemmas, for which they are drawn; exercises train concepts, and this has to be reflected in their metadata. The gap detection tool is aware of these rules and when it finds a learning object that misses a required relation, a gap is being detected. A more sophisticated approach is required for verifying semantic correctness of metadata relations. At the moment, the gap detection tool is capable of detecting two gaps of this kind: a learning object related to itself, and a learning object related to a non-existing item. Checking of correct types of metadata references is planned as one of the future tasks. Figure 1 demonstrates the results of structural gap detection presented to an ActiveMath author. Several learning objects are missing the for relations, which are supposed to connect them to the target concepts.



Structural Gaps for collection Mathefuehrerschein

Items without for attribut

mbase://Mathefuehrerschein/anwe th/anwendungen_aktmalerei [jE]
mbase://Mathefuehrerschein/probl brueche_1_th/problemloesen_brueche_9 [jE]

Items without for relations

ng relations

with selfrefe

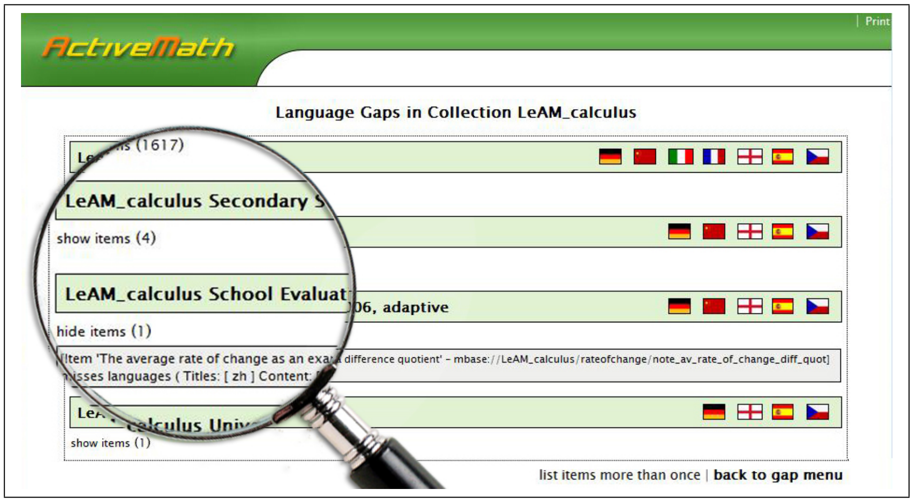
mbase: ge/ jE]
mbase://Mathefuehrerschein/sinus_ /symbol [jE]

Fig. 1. Result page of Structural Gap Detection

3.2 Linguistic Gap Detection

Preparing multilingual mathematical learning material is a delicate and demanding task that should take into account not only the translation of texts, but also differences in mathematical notations and cultural differences in mathematical education. Regarding content gap detection, an extra level of complexity is added, as current translation approaches do not reflect continuous development of content collections. Content items added after the translation effort is finished are unlikely to be translated into all target languages. If learning objects are modified afterwards, these modifications are usually done only in one language. These practices result in learning content collections that are marked as multilingual, but lack consistent translations for many learning objects.

The developed version of the gap detection tool does not attempt to solve all the mentioned challenges of exhaustive verification of learning content multilinguality. In the current ActiveMath authoring platform, detection of incomplete translations of learning objects and translation versioning is managed through collaborative authoring. At the same time, the existing linguistic gap detection component is responsible for verifying that each learning object appears only in linguistically valid contexts and that the metadata translations match the content translations. For example, a learning object that has not yet been translated into Spanish should not be referenced in a course designed for Spanish students; a learning object having its content translated into Spanish should have a title in Spanish, as well. Figure 2 demonstrates the results of linguistic gap detection performed for three multilingual ActiveMath courses.



Language Gaps in Collection LeAM_calculus

Le is (1617)

LeAM_calculus Secondary S

show items (4)

LeAM_calculus School Evaluat 06, adaptive

hide items (1)

[Item 'The average rate of change as an exa difference quotient' – mbase://LeAM_calculus/rateofchange/note_av_rate_of_change_diff_quot]
 isses languages (Titles: [zh] Content:

LeA calculus Univ

show items (1)

list items more than once | back to gap menu

Fig. 2. Result page of Linguistic Gap Detection

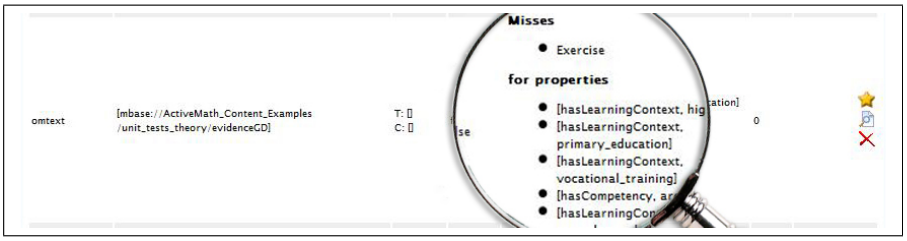
3.3 Didactic Gap Detection

Ensuring that existing learning objects provide correct metadata and appropriate translations is not enough for building a gap-free content collection. To be didactically complete the collection should contain enough learning objects for all possible (or at least, typical) learning situations. This is especially important for adaptive learning environments as they sequence learning objects dynamically based on the current state of the student model and, hence require a greater variety of learning objects in terms of their didactic properties. When an adaptive system fails to find a learning object with desired parameters, it tries to deal with the situation by finding the next best learning object. The latter option will put an extra burden on the student; s/he can receive, for example, an exercise that is too hard or too easy, or has been developed for another major (and therefore uses unfamiliar terminology) or educational level (and therefore the level of detail is unfitting).

To avoid such unfavorable learning situations, the authors are forced to take extra care when thinking about the necessary learning objects and how they fit possible students' needs. Unfortunately, no authors are capable of to reliably predict all of the potential adaptive scenarios. A brute force approach to this problem would be to design the exhaustive body of content, where for every learning object a full set of alternatives with all possible variations of metadata exists. This leads to overestimation of the number of gaps, since many of them will not be relevant in practice.

We have implemented a didactic gap detection component that relies on a more practical approach to verification of didactic completeness of learning content collections. Instead of implementing learning items for all possible situations, we first try to see which situations are possible. The didactic gap detection component employs the ActiveMath's course generation engine PAIGOS as a test bed, by providing it with generated typical user models and receiving gap reports. The rationale behind this approach is that the entire population of typical students taking the course 1) can be characterised with a limited set of typical user profiles, and 2) is well-known to the author developing the course. Hence s/he can feed these typical user profiles to the gap detection component, which will launch the course generator in the debugging mode and intercept its requests for learning objects with non-existing characteristics. Such requests will be then presented to the author as didactic gaps.

Figure 3 demonstrates an example of a didactic gap observed in ActiveMath content; a learning exercise with particular characteristics has been requested but not found by the course generator.



omtext

[mbase://ActiveMath_Content_Examples
/unit_tests_theory/evidenceGD]

T: ▯
C: ▯

se

Misses

- Exercise

for properties

- [hasLearningContext, hig
- [hasLearningContext, primary_education]
- [hasLearningContext, vocational_training]
- [hasCompetency, ar
- [hasLearningCon

ation]

0

Fig. 3. Representation of a detected Didactic Gap

4 Evaluation

The evaluation has focused on the following two aspects of the gap detection tool performance: utility estimated in terms of the tool's ability to detect gaps in the content that have been verified by human authors before; and interactivity measured by the response time a general-purpose PC would perform gap detection of a regular content collection. Evaluation of these two parameters allowed us to better understand the general value of the tool and provided some insights for its future development.

4.1 Utility Evaluation

For evaluating the performance of the developed tool we applied it for detecting gaps inside two well-established collections of ActiveMath learning content.

Structural gap detection has been applied to the collection *Mathefuehrerschein* consisting of 467 learning objects connected to each other by 1002 metadata relations. This collection has been developed in 2006 for the University of Applied Sciences, Dortmund, and has been available for more than 5000 students for self-evaluation of math background knowledge.

Even after several years of intensive testing and usage with real students, the collection still contained gaps. The automatic gap detection tool has found 40 structural gaps in this collection. 38 of these gaps were learning objects missing the connection to the domain concepts they train and the last 2 gaps were self-references.

The linguistic gap detection has been applied to the *LeAM-Calculus* collection. This collection has been developed in 2007-2008 to for teaching the domain of Calculus in five European languages: English, German, Spanish, Hungarian, and Czech. The collection contains 1620 learning objects that have been combined into 8 different courses covering different topics of Calculus on several educational levels (from secondary school to the first year university). Over the last two years these courses were used by more that 2000 students from several countries.

The linguistic gap detection component has found 163 gaps in the *LeAM-Calculus* collection. 144 of these gaps were learning objects referred from a course in a certain language, but missing both the title and the content translations into this language. The remaining 19 gaps indicate learning objects, for which the content has been translated into a language, but the title has not.

To evaluate the utility of the didactic gap detection tool we used it to generate a course on the part of the *LeAM-Calculus* collection (used in the previous subsection). The PAIGOS course generator was setup to generate two versions of a small course “Introduction to Differential Calculus” with didactic gap detection functionality enabled. Both versions of the course were generated for the exactly the same pedagogical conditions (learning strategy, set of learning goals). However, the simulated student profiles used by the didactic gap detection tool were different.

Table 1 outlines the main parameters of these learning profiles, as well as the differences between the generated courses. Student profile #1 models a rather poor student who did not know much before the course and does not learn very fast. Student profile #2 models a good student who was quite experienced in the beginning and also acquired new knowledge much faster. The course generator created different course for them (based on the same collection of learning material) and the gap detection tool has been able to discover quite different sets of didactic gaps.

Table 1. Didactic gap detection for two student profiles.

	Student profile #1	**Student profile #2**
Initial knowledge	25%	75%
Speed of learning	slow	fast
Default competency	Conceptual	Multistep
Number of learning objects in the course	*115*	*85*
Number of didactic gaps found	*34*	*29*

4.2 Interactivity Evaluation

The developed gap detection tool is envisioned as a part of the ActiveMath authoring environment that the authors of learning material can use for verifying their content collections. Hence, it is important to realise how fast the gap detection tool can produce the results in order to estimate its potential to be an interactive authoring component. In order to do so, we have taken the timings for all three of the gap detection experiments described in the previous subsection. The timings have been measured on a regular computer [4].

For the structural content analysis, the gap detection took, on average, 20.800 sec. The number of learning objects in the *Mathefuehrerschein* content collection is 467. Hence, for the experimental setup, the current implementation of the structural gap detection takes 20.800 / 467 = 44.54 ms per learning object.

The content collection used with the linguistic gap detection component contains 1620 learning objects. It the gap detection tool 56.861 seconds to produce the result of the linguistic content analysis. Thus, on average, the system needs approximately 56.861 / 1620 = 35.10 ms to scan a learning object for linguistic errors.

In order to analyze how much time the didactic gap detection requires, we first performed course generation procedure in isolation, and then repeated it for the same settings (student profile #1) with the gap detection component working. The first run of course generation took 41.253 sec to complete. With the didactic gap detection activated, the same course has been generated in 65.807 sec (60% increase in time). The course contained 115 learning objects. Consequently, it took approximately 65.807 / 115 = 572.23 ms to scan a learning object for didactic gaps.

[4] AMD Athlon XP3200+, 1 GB RAM, Java(TM) SE Runtime Environment Build 1.6.0 06-b02

4.3 Evaluation Results

The first part of the evaluation experiment provides preliminary evidence on the utility of our approach. The two content collections, on which we tested our gap detection tool, were developed and verified by teams of authors, and the were heavily used for several years by thousands of students. Yet, in the first case, for 1002 metadata relations, 40 structural gaps have been observed ($40 / 1002 \approx 4\%$). Among 1620 learning objects translated into five languages (this is $1620 \times 5 = 8100$ translations) of the second collection, 163 linguistic gaps have been discovered ($163 / 8100 \approx 5\%$). Finally, in the adaptively generated course approximately every third learning object introduced a didactic gap (it does not mean that 30% of the learning objects were incorrect, rather that they were suboptimal for this particular student). These numbers indicate a good potential for automatic gap detection and underline the demand for such solution.

The results of performance evaluation have shown us that the current implementation of the gap detection mechanism is rather slow, especially the didactic gap detection. Although, common web-servers are faster than the regular-purpose PC we used in our experiment, the content collections can be larger, hence implementing gap detection as an interactive authoring component might not scale very well. The current approach we have taken is to keep the detected gaps in the database; thus, the gap detection service launches only once, and after that the author interacts with the stored gap base.

5 Conclusion

This work addresses an important issue of providing automatic support of quality control for educational content. A special gap detection tool has been implemented to provide such support within ActiveMath platform. Three categories of content gaps have been identified: structural, linguistic and didactic. The developed tool can successfully detect gaps of all three categories and present them to the authors in a structured way.

The evaluation experiment has shown us that even in a mature learning content collection that were uses for several years by thousands of real students, our tool can automatically detect structural, linguistic and didactic gaps. The performance evaluation has provided us with the insights for the improving the scalability of the approach.

The future development of the tool is planned in two main directions. First, we plan to implement partial automatic (or semiautomatic) correction of the detected gaps. In some cases, missing metadata elements can be suggested to the author by using the metadata that have been already provided, or based on the similarity between the learning objects.

Our next plan is to convert the ActiveMath metadata into Web-ontology language (OWL), and implement a more semantically-aware gap detection tool. OWL will help us more efficiently control the types of metadata elements and learning objects and will allow us to discover more categories of gaps.

Acknowledgements

This material is based upon work supported under the project “ECP 2008 EDU 428046 Math Bridge” funded by the **eContentplus Programme** of the European Union.

References

1. Murray, T.: Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art. International Journal of Artificial Intelligence in Education. 10, 98–129 (1999)
2. van Harmelen, F., Fensel, D.: Practical Knowledge Representation for the Web. Proceedings of the Workshop on Intelligent Information Integration III99 during IJCAI-99, Stockholm, Sweden, August 1999 (2000)
3. ADL. Sharable Content Object Reference Model. The SCORM Content Aggregation Model, Advanced Distributed Learning Initiative (2001)
4. IEEE. IEEE standard for learning object metadata (IEEE 1484.12.1.2002), Learning Technology Standards Committee (2002)
5. IMS. IMS Learning Design Specification. Boston, The IMS Global Learning Consortium (2003)
6. Aroyo, L., Dicheva, D.: The New Challenges for E-learning: The Educational Semantic Web. Educational Technology & Society, 7 (4), 59–69 (2004)
7. Melis, E., Goguadze, G., Homik, M., Libbrecht P., Ullrich, C., Winterstein, S.: Semantic-Aware Components and Services of ActiveMath. British Journal of Educational Technology. 37(3), 405–423 (2006)
8. Kohlhase, M.: OMDoc: Towards an Internet Standard for the Administration, Distribution, and Teaching of Mathematical Knowledge. In: Proceedings of Artificial Intelligence and Symbolic Computation, pp. 32–52. Springer Berlin / Heidelberg (2000)
9. Melis, E., Libbrecht, P.: Methods for Access and Retrieval of Mathematical Content in Activemath. In: Takayama, J. G. N., Iglesias, A. (eds.), Proceedings of ICMS-2006, LNCS, vol. 4151. Springer, Heidelberg (2006)
10. Autexier, S., Hutter, D., Mossakowski, T., Schairer, A.: Maya: Maintaining Structured Documents. In: OMDoc – An open markup format for mathematical documents [Version 1.2]. LNAI, vol. 4180, pp. 296–300. Springer, Heidelberg (2006)
11. Krieg-Brückner, B., Mahnke, A.: Semantic Interrelation and Change Management. In: OMDoc – An open markup format for mathematical documents [Version 1.2]. LNAI , vol. 4180, pp. 274–277. Springer, Heidelberg (2006)
12. Ullrich, C.: Pedagogically Founded Courseware Generation for Web-Based Learning- An HTN-Planning-Based Approach Implemented in PAIGOS. In: Lecture Notes in Artificial Intelligence (5260). Springer, Heidelberg (2008)
13. Faulhaber, A., Melis, E.: An Efficient Student Model Based on Student Performance and Metadata. In: Proceedings of ECAI, pp. 276–280. IOS Press, Patras (2008)
14. Libbrecht, P., Melis, E.: Methods to Access and Retrieve Mathematical Content in ActiveMath. In: Proceedings of ICMS 2006, pp. 331–342. Springer, Heidelberg (2006)