

Adaptation “in the Wild”: Ontology-based Personalization of Open-Corpus Learning Material

Sergey Sosnovsky¹, I-Han Hsiao², Peter Brusilovsky²

¹Center for e-Learning Technology, DFKI,
Campus D3.2, D-66123 Saarbrücken, Germany
sosnovsky@gmail.com

²University of Pittsburgh, School of Information Sciences,
135, N. Bellefield ave., Pittsburgh, PA, 15260, USA
{peterb, hyl12}@pitt.edu

Abstract. Teacher and students can use WWW as a limitless source of learning material for nearly any subject. Yet, such abundance of content comes with the problem of finding the right piece at the right time. Conventional adaptive educational systems cannot support personalized access to open-corpus learning material as they rely on manually constructed content models. This paper presents an approach to this problem that does not require intervention from a human expert. The approach has been implemented in an adaptive system that recommends students supplementary reading material and adaptively annotates it. The results of the evaluation experiment have demonstrated several significant effects of using the system on students’ learning.

Keywords. Open-Corpus Personalization, Adaptive Educational System

1 Introduction

From the educational perspective, the WWW can be viewed as a very large collection of learning material. For many subjects, one can find online tutorials, textbooks, examples, problems, lectures slides, etc. Nowadays, teachers often do not have to create most of course materials themselves, instead they can reuse the best content available online. For example, a teacher developing a course on Java programming might decide to use a web-based Java tutorial, an electronic version of the course book, an existing Web-based assessment system, and online code examples. Although, all these resources are useful, students might get lost in this large volume of content without additional guidance. Organizing adaptive access to the course materials would help solving the problem. Appropriate tutorial pages can be recommended to students based on their progress; an adaptive navigation technique can be implemented to facilitate the choice of the most relevant example; an intelligent tutoring system can adaptively sequence learning problems. A teacher might be able to find a system implementing one of these technologies and providing adaptive access to one of the collections of learning material. However this system will not be aware of the rest of the available content, unless it supports Open-Corpus Personalization (OCP).

OCP is one of the classic problems of adaptive information systems, in general, and adaptive educational systems (AES), in particular. Many research projects tried to propose a solution for it with different degrees of completion (e.g. [1], [2], [3], [4]). Brusilovsky and Henze in [5] presented a comprehensive overview of the problem

and draw the evolution of research addressing it. This paper focuses on the OCP based on semantic content models, as the dominant personalization approaches in the field of e-Learning rely on representation of student knowledge and learning activities in terms of domain semantics. Therefore automatic extraction of domain knowledge from Web-content becomes an important component of the problem we address here. We propose a novel approach towards a *fully automated* OCP in the context of e-Learning. It is based on harvesting coarse-grained models from semi-structured digital collections of open-corpus education material (such as tutorials and textbooks) and mapping them into the pre-defined domain ontology serving as the main domain model and the reference point for multiple open-corpus collections. Once the mapping is established, the content from the processed open-corpus collection can be presented to students in adaptive way, according to their student models computed in terms of the central ontology. The rest of this paper describes the details of the approach, the adaptive e-learning service implementing it, and the results of the evaluation experiment demonstrating several learning effects of the developed service.

2. Ontology-based OCP Approach

Information on the Web is not without structure. Authors of many online resources create them as a reflection of their own internal organization of related knowledge. They encode this organization by formatting the text with lists and headings, breaking documents into sections and pages, linking pages together, creating tables of contents, etc. The approach proposed in this paper attempts to utilize this hidden semantic layer of well-formatted content collections to achieve fully automated OCP. The entire procedure consists of the three steps presented below.

Step 1: Modeling of Open-Corpus Content in Terms of its Structure. An author creating an instructional resource tries to make it more readable and understandable by structuring it into chapters and sections. Every section is intended to represent a coherent topic. It is given a title conveying the meaning of the topic and contains the text explaining it. Their main purpose is to structure content, but they inescapably structure the knowledge, as well. A topic-based structure of such a resource can be parsed automatically and represented formally, e.g. as an RDF model. This model will have some drawbacks: (1) subjectivity; (2) poor granularity; (3) undefined semantics of topics and relations between them; (4) incompleteness. Yet, such model provides means to access the material of the collection in terms of topics, reason about the material in terms of topics and adapt the material in terms of topics.

Step 2: Mapping Extracted Model into the Central Domain Ontology. Extraction of the hidden semantic layer is not enough for two reasons. First, coarse-grained domain and content models can be effective when delivering the adaptation to students, but cannot maintain student modeling of good quality [6]. Second, a model extracted from a single collection can be used to adaptively present only the content of this collection. The learning material from different collections will be isolated. The solution is to use the central domain ontology as a reference model. It will help to model students' knowledge and to translate between the topic-based structures of individual open-corpus collections. The connection between the harvested models and the central ontology is established based on the automatic mapping of these models.

Step 3: Mediated Personalization of Open-Corpus Learning Material. Once the two models are mapped the systems can reason across them. The mapping bridge between the central ontology and the tutorial model enables two principle procedures: (1) tracing student’s actions with the tutorial’s topics, representing these actions in terms of the ontology concepts and updating the ontology-based student model; (2) requesting the current state of student model expressed in terms of ontology concepts, translating it into the open-corpus topics, and adapting students’ access to the open-corpus material. Fig. 1 summarizes the principle relation between the components of the central ontology and the open-corpus material, as well as the information flow across these relations.

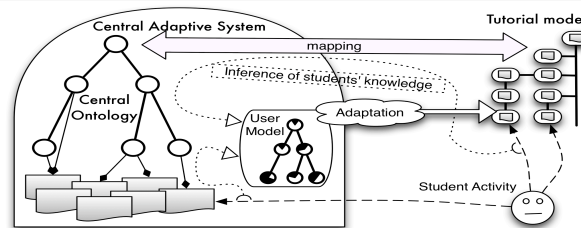


Fig. 1. Meditated personalization of open-corpus learning material

3. The Ontology-Based Open-Corpus Personalization Service

The proposed approach has been implemented in the Ontology-based Open-corpus Personalization Service (OOPS). It has been developed as a value-added service used in parallel with a central exercise system and augmenting it with adaptive access to supplementary reading material. As a central system we used QuizJET – the system serving parameterized exercises for Java programming language [7]. Both *OOPS* and *QuizJET* have been integrated with the *CUMULATE* user modeling server [8]. As an open-corpus collection of instructional material we used the electronic version of an introductory Java textbook. *QuizJET* is responsible for objective assessment of students’ knowledge. Its exercises are indexed in terms of the central ontology and it report students’ activity to the central user modeling component – *CUMULATE*, which models students’ knowledge in terms of the central domain ontology and reports it to *OOPS*. As a result, student practicing with QuizJET exercises and struggling with a difficult topic receives recommendations of relevant open-corpus reading material from *OOPS*.

The student interface of *OOPS* has two interaction phases: recommendation (when a student is presented with a list of recommended pages) and reading (when a student is studying recommended material). Left part of Fig. 2 presents a screenshot of the recommendation phase. Area “B” is the interface of the central system – *QuizJET*. Area “A” presents a list of recommendations produced by *OOPS* for the current exercise of *QuizJET*. Every item in the list is a topic label from the harvested open-corpus content collection. The order of an item in the list is determined by its relevance to the current *QuizJET* exercise computed based on the aggregated similarity of the topic and the concepts indexing the exercise. The similarity values are calculated by the ontology mapping algorithm. The recommended items are provided with adaptive annotation in form of human-shaped colored icons. The

coloring of an icon annotating a topic represents the amount of knowledge a student has demonstrated for the learning material behind this topic measured in terms of central ontology concepts mapped to the topic and provided by the central student model. The annotation level is computed as a weighted aggregate of knowledge levels for all concepts mapped into the topic. Once a student decides to accept a recommendation by clicking on a topic link, s/he goes into the reading phase of the OOPS interface (right part of Fig. 2). In this phase, OOPS provides a student with an opportunity to read the actual material behind the topic link. OOPS widget expands, and its interface changes. The expanded interface contains three main areas. Area “A” is the content area, where the content of the selected recommendation is presented. Area “B” is the navigation area, where the links to the previous and the next topics are presented, should the student choose browsing the structure of the open-corpus collection. Area “C” contains two buttons that allow the student to exit the reading phase and to report whether s/he has found the recommendation useful for the current learning task or not. Once the student leaves the reading phase, *OOPS* interface switches to the recommendation phase again.

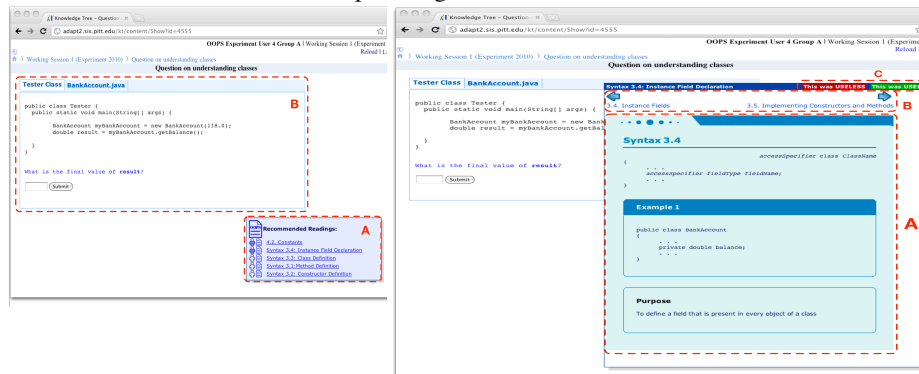


Fig. 2. Interface: *Left*: recommendation phase; *Right*: reading phase

4. The Evaluation

This section presents the results and the procedure of the *OOPS* service evaluation. It was organized as a controlled balanced experiment comparing the developed system against two control conditions. The experimental system (*open-corpus*) provided students solving QuizJET exercises with open-corpus recommendation of reading material. Another version of the system (*closed-corpus*) had the identical interface and generated recommendations from the same pool of reading material, but used traditional closed-corpus adaptation approach based on the manual indexing of recommended pages. The last configuration of the system (*textbook*) did not recommend any reading material. Instead, students using this version had a hard copy of the textbook, which was the source of reading material for the first two versions. The experiment consisted of two sessions corresponding to two sets of introductory Java topics. First set included simpler topics: from basics of variable and object handling to conditional statement and Boolean expressions. The second set covered more advanced topics: from loops to arrays and ArrayLists. Each session started with a pretest, continued with the 30 minutes work with the system and ended with the

posttest. Forty subjects with limited Java programming experience participated in the experiments. Subjects were randomly assigned to one of the four groups:

- A. Session 1_{easy topics} – *open-corpus*; Session 2_{complex topics} – *closed-corpus*;
- B. Session 1_{easy topics} – *open-corpus*; Session 2_{complex topics} – *textbook*;
- C. Session 1_{easy topics} – *closed-corpus*; Session 2_{complex topics} – *open-corpus*;
- D. Session 1_{easy topics} – *textbook*; Session 2_{complex topics} – *open-corpus*;

General Learning Effect. In order to verify that work with the system actually leads to learning, pair-wise comparisons of scores on the pre-test and the post-test have been made (Table 1). Significant learning has been registered for all groups and conditions during Session 1. For Session 2, the *open-corpus* condition and the *closed-corpus* condition resulted in significant (or bordering on significance) learning. At the same time, the *textbook*-condition led to no learning.

Table 1. General learning effect statistics ($Score_{post-test}$ VS. $Score_{pre-test}$)

Group	Session 1		Session 2	
	t(9)	p-value	t(9)	p-value
A	3.787	0.004	1.941	0.084
B	4.409	0.002	0.0	1.0
C	8.213	<0.001	2.250	0.051
D	4.077	0.03	3.361	0.008

Effect on Learning Complex Material. The main difference between Session 1 and Session 2 is material complexity. The analysis of the general learning effect suggests that the recommendation of supplementary reading material can have a positive influence on learning the complex learning material. During Session 1 (easy topics), none of the comparisons resulted in significant difference in Knowledge Gain. However, once the learning material became more complex (Session 2), the *open-corpus* system significantly outperformed the *textbook*: Knowledge Gain for the *open-corpus* condition (M=1.55; SD=1.23) is significantly higher than Knowledge Gain for the *textbook* condition (M=0.60; SD=0.97): $t(28) = 2.124$; $p = 0.043$ during Session 2 (complex topics). At the same time, no difference was observed between the closed-corpus and the open-corpus system when students were learning complex material. This is an important effect with a reasonable explanation. When learning easy material, students need less support from the system. They learn just by practicing with *QuizJET* exercises. And if they need extra reading, it is easier for them to find a relevant chapter in the textbook. On the other hand, when the material becomes complex, students can benefit from the recommendations and the adaptive annotations guiding them to the most important piece of reading material. Thus, personalized learning support results in better learning when support is needed. The comparison of open-corpus and closed-corpus conditions show that they are equally effective, which indicates that OCP produced by *OOPS* has similar quality to the traditional closed-corpus personalization.

Effect on Learning Conceptual Material. The personalization implemented by *OOPS* is aimed at achieving two instructional goals: (1) Support students solving self-assessment exercises by bringing them the most relevant reading material; (2) Balance students' learning by giving them the opportunity to read instructional texts in addition to practicing. The second means that *OOPS* should contribute better to the knowledge of important concepts and fact in the domain. The pre- and post-tests of

both sessions contained two kinds of questions: those evaluating students' practical skills in code understanding and manipulation and those checking their factual and theoretical knowledge. In order to measure the conceptual knowledge gain, only the second kind of questions was taken into account. The comparison of conceptual knowledge gain between the *open-corpus* and the *textbook* conditions shows that the hypothesis is partially confirmed. During Session 1, the conceptual knowledge gain for the *open-corpus* condition ($M=2.61$; $SD=1.75$) was higher, than for the *textbook* ($M=1.75$; $SD=1.15$), but not significantly: $t(28)=1.762$; $p=0.089$. However, during Session 2 (complex topics), the conceptual knowledge gain for the open-corpus condition ($M=0.73$; $SD=0.47$) was significantly higher than the one for the textbook condition ($M=0.30$; $SD=0.42$): $t(28)=2.403$; $p=0.023$. No significant effect was observed when comparing the open-corpus and the closed-corpus conditions.

5. Conclusion

In this paper, we have addressed the problem of OCP in the context of e-Learning and proposed a solution for it. As a proof-of-concept the adaptive e-Learning service OOPS has been implemented. It adaptively recommends and annotates pages for supplementary reading to students solving self-assessment exercises. The evaluation of OOPS has shown that students were able to achieve significant learning while using the open-corpus version of the system. *OOPS* significantly improved students' knowledge gain when they work with more challenging learning material. In comparison, students using the textbook demonstrated no significant learning while working with complex topics. *OOPS* helped to maintain a more balanced learning by significantly improving gain in conceptual knowledge, no such effect was observed for the textbook. At the same time, on no tests, we could statistically distinguish between the results of the proposed fully-automated open-corpus approach and a conventional closed-corpus technique based on a carefully handcrafted content model.

References

1. Henze, N. and W. Nejdl, *Adaptation in open corpus hypermedia*. Int. J. of Artificial Intelligence in Education, 2001. 12(4): p. 325-350.
2. Carmona, C., et al., *SIGUE: Making Web Courses Adaptive*, in *AH'2006*, Springer Berlin / Heidelberg, p. 376-379.
3. Jovanović, J., D. Gasevic, and V. Devedzic, *Ontology-based automatic annotation of learning content*. Int. J. on Semantic Web and Information Systems, 2006. 2(2): p. 91-119.
4. Apted, T. and J. Kay, *MECUREO Ontology and Modelling Tools*. Int. J. of Continuing Engineering Education and Lifelong Learning, 2004. 14(3): p. 191-211.
5. Brusilovsky, P. and N. Henze, *Open Corpus Adaptive Educational Hypermedia*, in *The Adaptive Web: Methods and Strategies of Web Personalization*, 2007, Springer Verlag: Heidelberg, Germany, p. 671-696.
6. Sosnovsky, S. and P. Brusilovsky, *Layered Evaluation of Topic-Based Adaptation to Student Knowledge*, in *Workshop on Evaluation of Adaptive Systems at UM'2005* Eduburgh, UK, p. 47-56.
7. Hsiao, S., P. Brusilovsky, and S. Sosnovsky, *Web-based Parameterized Questions for Object-Oriented Programming*, in *E-Learn 2008*, Las Vegas, NV, USA, p. 3728-3735.
8. Brusilovsky, P., S. Sosnovsky, and O. Shcherbinina, *User Modeling in a Distributed E-Learning Architecture*, in *UM'2005*, Edinburgh, UK, p. 387-391.