ECP-2008-EDU-428046

Math-Bridge

Decisions for remedy implemented

Deliverable number	*D-7.1*
Dissemination level	*Public*
Delivery date	*31 October 2010*
Status	*final*
Authors	*Eric Andrès (Saarland University)*

e*Content*plus

This project is funded under the *e*Content*plus* programme[1] ,
a multiannual Community programme to make digital content in Europe more accessible, usable and exploitable.

[1] OJ L 79, 24.3.2005, p. 1.

Project ref.no.	ECP-2008-EDU-428046
Project title	Math-Bridge European Remedial Content for Mathematics

Deliverable dissemination level	Public
Contractual date of delivery	October 31^{th}, 2010
Actual date of delivery	October 31^{th}, 2010
Deliverable number	D-7.1
Deliverable title	Decisions for remedy implemented
Type	Demonstrator/Report
Status & version	Draft
Number of pages	10
WP contributing to the deliverable	WP7
WP/Task responsible	Eric Andrès
Authors	*Eric Andrès (Saarland University)*
EC Project Officer	Marcel Watelet
Keywords	adaptation,personalization

Contents

Introduction

This deliverable is about the implementation of the decisions for the remedial scenarios detailed in deliverable 1.3 [BHFW10a]. We will briefly describe the general architecture of the tutorial component that implements the adaptive part of the Math-Bridge service. Then we detail implementation challenges and problems posed by the specified scenarios. Finally, we describe the technical implementation of the remedial scenarios.

1 Adaptive Course Generation

Course generation offers a tradeoff between pre-authored one-size-fits-all courseware and individual look-up of learning objects [BV03]. It supports learners by an on-demand assembly of sequences of learning objects. The generations take information about learning objects, the learner and his/her learning goals into account. A course generator assembles the sequences according to a set of methods that implement pedagogical knowledge. The methods determine which learning objects the course generator includes in a course, the order of these objects, the structure (sections/subsections) of a course, etc. PAIGOS, the course generation component used in the Math-Bridge service, is able to generate courses according to different scenarios [Ull08]. Competencies play an essential part in Paigos, especially for the exercise and example selection. It uses a hierarchical network of tasks to plan the sequence of learning objects. The implementation is done by declarative methods that specify how the course should be planned. These methods rely on information about the student and the content that is provided by the *mediator*, a dedicated component that reads data from the content storage and the learner model and translates these to a format suitable for course generation. The Math-Bridge service uses a customized version of Paigos to provide adaptive and personalized learning opportunities.

2 Specified Scenarios

All scenarios described in deliverable 1.3 are based on the new competency system developed by pedagogical partners [BHFW10b] and are as such different from the pre-existing scenarios. For some of these scenarios, available methods could be re-used. However, as competencies play an essential part in course generation, especially for the exercise and example selection, it was a challenge to adapt these to the new competency model. In the previous course generation methods, competencies were hard-coded, since the scenarios were designed in a project focusing on the PISA framework.

Five scenarios from D1.3 are based on existing methods, namely *LearnNew*, *Rehearse*, *Workbook*, *TrainCompetency* and *ExamSimulation*, and three had to be modified at the

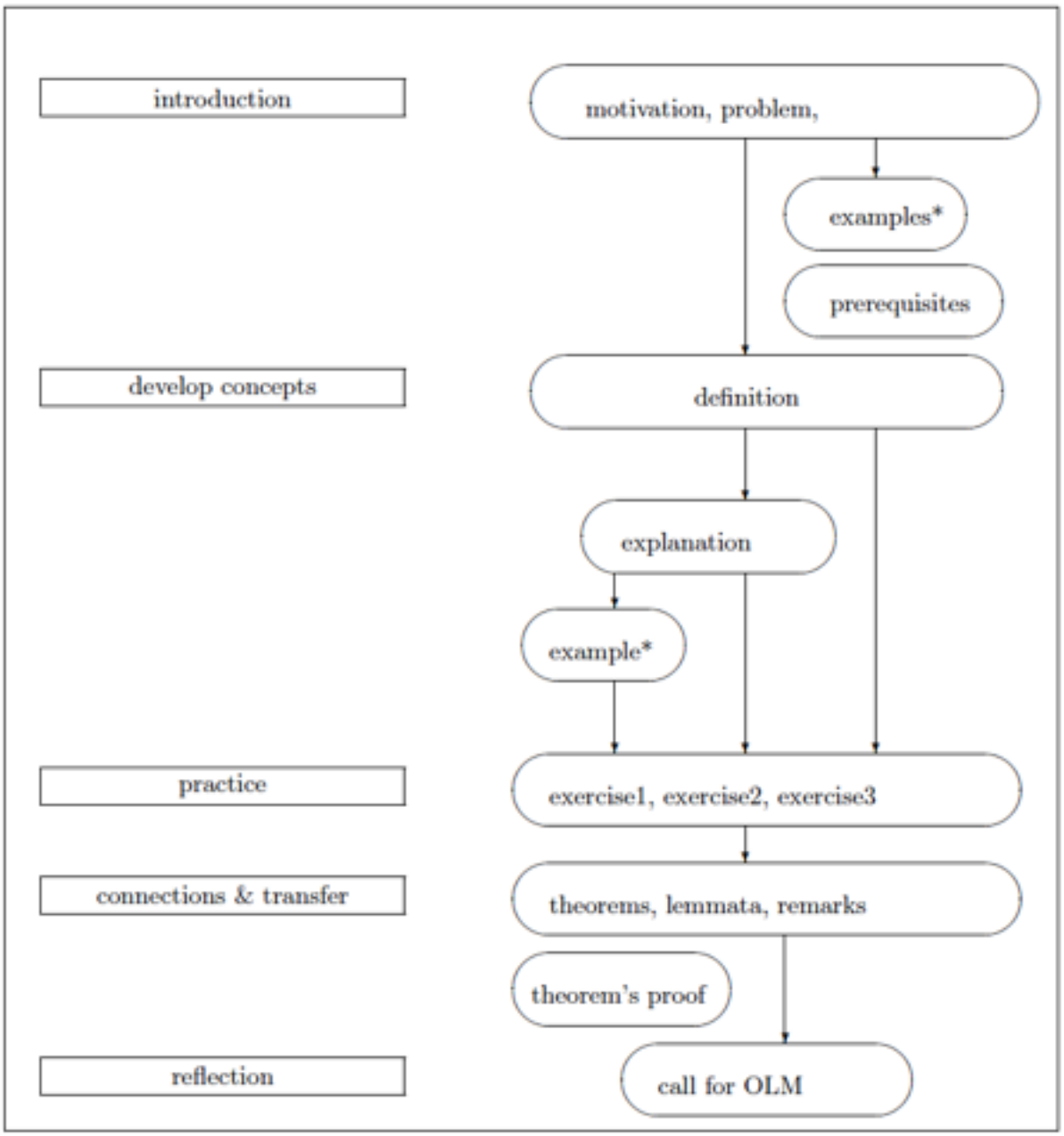

Figure 1: Scenario *LearnNew*

declarative level to reflect the new competency system. The scenarios *LearnNew* and *Rehearse* could be fully re-used at the declarative method level. We give a brief description of *LearnNew* to point out a difficulty that arises with all of the scenarios. Figure 1 shows a schematic representation of the scenario. It gives an overview of the different stages of learning targeted by the generated course. The *introduction* part is done by showing a motivating learning object first, followed by a sequence of examples and prerequisites. The selection of the examples and prerequisites depends on the availability of content and the current state of the learner model. Although the general structure of the scenario is identical in Math-Bridge, the criteria for the selection of examples and exercises are not the same than in the original version of Paigos that was based on PISA. The contents in Math-Bridge are based on the new competency system, which is also reflected in the learner model, so the *mediator* can no longer find the information it needs. Thus, the mediator had to be customized to work with the new competency model.

The original *Workbook*, *TrainCompetency* and *ExamSimulation* scenarios were based on the PISA framework and worked with these competencies in the declarative methods. Figure 2 shows a method to practice all competencies in a hard-coded fashion, the PISA competencies are explicitly listed. In order for this kind of method be used in the context of Math-Bridge, we needed to implement changes that added some flexibility with regard to the competency system.

```
(:method  (practiceAllCompetencies ?c)
             () ;; no "if" part, method always applicable
             ;; then
             (
              (practiceCompetency think ?c)
              (practiceCompetency solve ?c)
              (practiceCompetency represent ?c)
              (practiceCompetency language ?c)
              (practiceCompetency model ?c)
              (practiceCompetency argue ?c)
              (practiceCompetency tools ?c)
              (practiceCompetency communicate ?c)))
```

Figure 2: Training all competencies, original version.

In D1.3, the pedagogical partners advocated the need for complex learning objects (CLO) as an additional means to structure content:

> Within the learning material of the VEMA-material the sequence of atomic learning objects are not (always) freely exchangeable. Some sequences of atomic

> learning objects belong strictly together due to findings of didactical subject matter analysis. For instance, an "introduction" or an "explanation" forms a holistic unit of learning. They contain several atomic learning objects and intermediate text. The individual learning objects are reusable in other contexts, but it would be helpful to keep the whole learning unit as well, which could be reused as well in other contexts. We call these units "complex learning objects"

One of the specified scenarios relies on complex learning objects and is described with a set of different configurations. This gave rise to a whole set of problems that needed to be addressed, the central ones being how these complex learning objects should be represented in the system and how to make sure that a constituent part of such a CLO does not appear twice in a generated course (in the CLO and as single learning object).

3 Technical Implementation

The implementation was done by customizing Paigos and developing declarative methods for the specified scenarios. As described in section 1, Paigos gets the information about available content and the learner model from the *mediator*. It is implemented in JAVA using the JENA framework to manipulate data. We extended it with a method accepting the new competency model values as parameters. This method was implemented in such a way that it abstracts the competency being queried. Instead of requiring a call to a dedicated method, it now supports passing the competency as a parameter. This effectively enables Paigos to access the content storage and the learner model using the new competency model.

```
(:method (prove! ?c)
         ;; if
         ((learnerProperty "hasCompetencyArgue" ?c ?argue)
          (>= ?argue 3)
          (assign ?exercise (call GetResources
                                  ((class Exercise) (relation isFor ?c)
                                   (property hasCompetency argue)))))
         ((insert ?exercise))) ;; then part
```

Figure 3: Selecting an exercise of a specific competency.

Because of this modification, the methods selecting the learning objects to be inserted also needed to be adapted. To give an example, the method shown in 3 selects an exercise using the original approach. It calls a dedicated method to query the learner's competency to

argue. With the new approach, an exercise can be selected as in the method illustrated in figure 4: The competency is given as a parameter.

```
(:method (practiceCompetency! ?competency ?c)
         () ;; no "if" part, method always applicable
         ((trainWithSingleExercise ?c very_easy ?competency) ...
          (trainWithSingleExercise ?c very_difficult ?competency)))

(:method (trainWithSingleExercise! ?c ?difficulty ?competency)
         ;; if suitable exercise exists
         ((assign ?exercise (call GetResources
                              ((class Exercise) (relation isFor ?c)
                               (property hasDifficulty ?difficulty)
                               (property hasCompetency ?competency))))
         ((insert ?exercise))) ;; then part
```

Figure 4: Selecting an exercise for a competency given as parameter.

A possible solution to resolve the intertwining of the competency model and the declarative knowledge represented by the Paigos methods was already proposed in a paper presented at the International Conference on Web-based Learning in 2010[MFDU10]. We implemented the proposed approach. The following example illustrates the procedure we applied. In order to remove the dependency on the competency system from the method described in figure 2, we now initially have the Math-Bridge service supply Paigos with the set of competencies it intends to use. All methods can then reference these competencies and use them in a parametric way. Figure 5 shows the method after the parametrization has been applied to it. Instead of listing all competencies, it now accesses the supplied set of competencies, `(competencies)`, and iterates through it. This approach enables the usage of methods in a way that is decoupled from the competency system, where appropriate.

Complex learning objects are represented as groups of learning objects. This decision requires that complex learning objects be sliced into the finer-grained learning objects that are used by all components of the Math-Bridge service. A more detailed description is provided in the Math-Bridge metadata cookbook[Sos10]. We extended the *mediator* by two methods, one to query for complex learning objects and another one to access a CLO's constituent learning objects. These methods enable Paigos to retrieve and insert CLOs in the generated course. The methods that constitute the dedicated CLO scenario use the extensions of the *mediator* to retrieve available CLOs and insert them at the learning object level. We implemented a new method `insertCLO` that can be used when inserting CLO constituents in a generated course. This method will enforce the insertion of all constituents, removing duplicates as necessary in the case they would already have been

```
(:method (practiceAllCompetencies ?c)
         ()
         (practiceAllCompetenciesH ?c (competencies ?comps)))
(:method ((practiceAllCompetenciesH ?c (competencies (?head . ?rest)))
         ()
         ((practiceCompetency ?head ?c)
          (practiceAllCompetenciesH ?c (competencies ?rest))))
(:method (practiceAllCompetenciesH ?c nil)
         ()
         ())
```

Figure 5: Training all competencies, customized version.

marked for insertion earlier.

4 Bibliography

References

[BHFW10a] Rolf Biehler, Reinhard Hochmuth, Pascal Fischer, and Thomas Wassong. Pedagogical remedial scenarios. Public Project Deliverable, 2010.

[BHFW10b] Rolf Biehler, Reinhard Hochmuth, Pascal Fischer, and Thomas Wassong. Supplement to deliverable 1.1 - codebook. Restricted Project Deliverable Supplement, 2010.

[BV03] Peter Brusilovsky and Julita Vassileva. Course sequencing techniques for large-scale webbased education. *International Journal of Continuing Engineering Education and Lifelong Learning*, 13(1/2):75–94, 2003.

[MFDU10] Erica Melis, Arndt Faulhaber, Ahmad Salim Doost, and Carsten Ullrich. Supporting flexible competency frameworks. In *Proceedings of 9th International Conference on Web Based Learning - ICWL 2010*, volume 6483 of *Lecture Notes in Computer Science (LNCS)*. Springer, 2010.

[Sos10] Sergey Sosnovsky. Supplement to deliverable 1.1 - metadata cookbook. Restricted Project Deliverable Supplement, 2010.

[Ull08] Carsten Ullrich. *Pedagogically Founded Courseware Generation for Web-Based Learning – An HTN-Planning-Based Approach Implemented in PAIGOS*. Number 5260 in Lecture Notes in Artificial Intelligence. Springer, November 2008.