TOWARDS A GENERIC INTEGRATION OF ADAPTIVE ASSESSMENT SYSTEMS WITH LEARNING ENVIRONMENTS

Christian Saul, Felix Dürrwald
Fraunhofer Institute Digital Media Technology, Business Area Data Representation & Interfaces, Ilmenau, Germany

Heinz-Dietrich Wuttke
Faculty Informatics and Automation, Ilmenau University of Technology, Ilmenau*, Germany*

Keywords: Adaptive Assessment Systems, Learning Environments, Integration, Interoperability.

Abstract: Personalization is becoming a crucial factor in many areas of life including education. Currently, a learning environment (LE) such as moodle, ILIAS, OLAT and dotLRN is far from being able to adapt the assessment to the students' individual context, prior knowledge and preferences, because personalization is still insufficiently implemented or even not addressed in this system. In contrast, an adaptive assessment system (AAS) takes the students' characteristics into account in order to personalize the assessment, which may result in more objective assessment findings. This paper analyzes how current open standards and specifications can be used to achieve integration between LEs and AASs seamlessly, so that they can profit from each other. For that reason, both LEs and AASs requirements are analyzed, because they have great influence on the further considerations. LEs require control information as well as assessment information from the AASs, which in turn require student as well as assessment information from the LEs. As a result of the paper, an interworking of several standards and specifications (OPAQUE, IEEE LTSC PAPI, IMS LIP, IMS QTI, etc.) is proposed, which can be used to achieve integration between LEs and AASs.

1 INTRODUCTION

Assessment in the educational settings is defined as classifying, usually in measurable terms, knowledge, skills, attitudes and beliefs of a student.

The functionalities to be provided by an assessment system are: (1) identify students and define which students have to take which tests at what times and which questions make up each test, (2) record the scores and other information that result from students attempting the test and provide reporting facilities for this information, (3) display questions to students and process the responses to generate scores and feedback and (4) allow authors to create and configure the questions.

Currently, a learning environment (LE), also known as learning management system (LMS), such as moodle (http://www.moodle.org), ILIAS (http://www.ilias.de), OLAT (http://www.olat.org) and dotLRN (http://www.dotlrn.org) realizes number 1 and 2 almost satisfying (Wuttke et al., 2008). LEs usually have facilities to administrate and support students during assessment, but they are relatively weak in addressing functionalities 3 and 4. Moreover, they are limited in posing question types and using algorithms to analyze and process students' responses. In addition, if authors want to create advanced questions and tests or content with high levels of interaction, they are limited to the tools provided by individual environments, which, in turn, restrict the reusability and interoperability of the created content. This often leads to an "one-size-fits-all" approach, where all students are presented the same questions. From a pedagogical point of view, however, personalization support is crucial to keep up the motivation and interests of the students, which are critical success factors in the assessment process.

An adaptive assessment system (AAS) poses a better alternative. It takes the students' individual context, prior knowledge and preferences into account in order to personalize the assessment, which may result in more objective assessment findings. The benefit of integrating LEs and AASs

Saul C., Dürrwald F. and Wuttke H..
TOWARDS A GENERIC INTEGRATION OF ADAPTIVE ASSESSMENT SYSTEMS WITH LEARNING ENVIRONMENTS.
DOI: 10.5220/0003344001170122
In *Proceedings of the 3rd International Conference on Computer Supported Education* (CSEDU-2011), pages 117-122
ISBN: 978-989-8425-50-8

lies in combining the strengths of both systems. LEs provide administration, learning and support capabilities, while AASs provide personalized assessment to students.

The focus of this paper is to analyze, how current open standards and specifications can be used to achieve integration of LEs and AASs. The work presented in this paper is part of an overall project aiming at implementing a new AAS (Saul et al., 2010).

The remainder of the paper is organized as follows: The second and third chapters describe LEs and their requirements, which an AAS should meet as well as AASs and their information requirements. The fourth chapter deals with current standards and specifications, which can be used to integrate LEs and AASs. Chapter five proposes solutions, which are discussed in chapter six. Concluding remarks and references complete the paper.

2 LEARNING ENVIRONMENTS

LEs such as moodle, ILIAS, OLAT and dotLRN are designed to support and enhance learning and training in educational settings.

Generally, many LEs intersperse content with assessments. The questions and tests, which make up the assessments, are mostly created by tools provided by the specific LE. The functionalities provided by these tools encompass simple monitoring mechanisms often used to unlock further content. However, they do not provide sophisticated control mechanisms, able to realize any kind of individualization or personalization. In addition, the lack of standardization on how these simple control mechanisms are represented faces the authors of questions and tests with the decision whether they define control rules for a specific LE or not. However, LEs generally offer extensive management facilities to the authors such as student and class management, course assembly and publishing as well as student tracking across courses. Therefore, LEs can offer student management facilities to the AASs. In order to enable a successfully integration of LEs with AASs, the LEs require control information as well as assessment information from the AASs.

2.1 Control Information

If a student executes a test, the LE needs to be informed when the student has finished the test. These control information can be used by the LE to determine the state in the learning process, for example, to lock further content or to start the test environment again upon logging into the LE for the next time. This will happen until the LE has received the information from the AAS that the student has taken the test completely. In addition, the AAS should provide mechanisms to resume the assessment in case the connection is interrupted or lost.

2.2 Assessment Information

The LE needs to be informed about the results achieved after the student has finished answering the questions. This information encompass the question that was asked, the final answer, the reached scores for the answer and attempts the student took in getting the final answer. The assessment information can be used by the LE to report and compute overall test scores.

3 ADAPTIVE ASSESSMENT SYSTEMS

AASs such as SIETTE (Conejo et al., 2004), PASS (Gouli et al., 2002), CosyQTI (Lalos et al., 2005) and iAdaptTest (Lazarinis et al., 2009) take into account the students' individual context, prior knowledge and preferences in order to personalize the assessment. Although these systems adapt the assessment process of each student resulting in presenting different questions, they still enable a better comparability between different individuals. Moreover, they reveal the current areas of strength and weakness of the students more precisely.

In order to seamlessly launch the AAS by any LE, the AAS require student as well as assessment information from the LE.

3.1 Student Information

Due to the fact that AASs personalize the assessment, they need to be informed about learning aspects of the student, such as prior knowledge, learning preferences, etc. This, in turn, causes several problems. The first is how to present the information in a way that all LEs and AASs are able to understand. Another problem arises, if the LEs are not able to provide learning information about the student to the AASs. In such cases, the AAS is responsible to pre-test the student to obtain this information. When the student launches the AAS again, the AAS uses the previously determined information and a new pre-test can be avoided. Moreover, the information gathered during the assessment process can also be used to determine and refine the student information. A further requirement of the AAS is to uniquely identify students.

3.2 Assessment Information

When the LE asks the AAS to launch a particular test, it needs to uniquely identify that test. The required information encompasses the location of the questions, the test identification and possibly the version of the test.

4 STANDARDS AND SPECIFICATIONS

The following standards and specifications can be used to enable the integration of AASs with LEs.

4.1 Student -Modeling

The IMS Learner Information Package (LIP) specification (2005) addresses interoperability between internet-based LEs. It holds, maintains and manages student information in XML documents. Moreover the specifications records metadata such as timestamp, source and privacy information.

The IEEE LTSC Public and Private Information (PAPI) specification (2002) is a standard to exchange student data between different systems. It represents the students' knowledge by specifying the student-model.

As there are disjoint attributes in both specifications, for example, privacy and security issues, there are often combinations of elements of both specifications used (Lalos et al., 2005; Lazarinis et al., 2006).

4.2 Domain-Modeling

The XML Topic Maps (XTM) specification (2001) defines a grammar for representing the structure of information resources. XTM describe relations as associations, mentioned in addressable information resources (occurrences).

4.3 Questions and Tests

The IMS Question & Test Interoperability (QTI) specification (2006) describes a data model for representing question and test data and their corresponding results supporting the exchange of this material between authoring and delivery systems. It structures material into assessments, sections, and items.

4.4 Interworking

RQP (Remote Question Protocol) (Delius, 2005) is a Web Service protocol based on SOAP (2007) and has been developed by the Serving Mathematics project (http://maths.york.ac.uk/serving_maths) aiming at developing assessment tools in mathematics education. Although RQP looked very promising, the effort ran out of resources and the protocol has never been finished. Conceptually similar to RQP and in a working state is OPAQUE.

OPAQUE (Open Protocol for Accessing QUestion Engines) (Hunt, 2008) is also based on SOAP and allows LEs to delegate the presentation of questions, the scoring of responses and the generation of feedback to a remote question engine. However, the LE takes full responsibility for authenticating students and asks an appropriate question engine to render each question. The question engine will then process the request and pass a response back to the calling LE. Although OPAQUE has been implemented into the LE moodle as well as into the question engines OpenMark (https://openmark.dev.java.net) and STACK 2.0 (http://stack.bham.ac.uk), it is designed to allow interoperability between arbitrary different types of LEs and question engines.

5 PROPOSED SOLUTION

In this chapter a data structure and an interworking of open standards and specifications is proposed to achieve a seamless communication between LEs and AASs. The proposed solution facilitates the communication of control, student and assessment information and thus enables personalized assessment. The interworking includes the following actions:

- Launching the test
- Communicating between AAS and LE
- Completing the test

5.1 Launching the Test

When the test is launched from the LE, it needs to initialize its connection with the AAS. OPAQUE defines the following Web Service message to initiate a connection request between a LE and a question engine. Please notice that AASs can simply be referred to as question engines, although they do much more than rendering questions.

```
<wsdl:message name="startRequest">
   <wsdl:part name="questionID"
   type="soapenc:string"/>
   <wsdl:part name="questionVersion"
   type="soapenc:string"/>
   <wsdl:part name="questionBaseURL"
   type="soapenc:string"/>
   <wsdl:part name="initialParamNames
   type="impl:ArrayOf_soapenc_string"/>
   <wsdl:part name="initialParamValues"
   type="impl:ArrayOf_soapenc_string"/>
   ...
</wsdl:message>
```

The message parts *questionID* and *questionVersion* can be used to identify the adaptive test, whereas *questionBaseURL* can be used to define where the adaptive test resides. As mentioned earlier, AASs not only require assessment information, but also need student information. This information can be submitted from the LE to the AAS using IEEE LTSC PAPI/IMS LIP within the message parts *initialParamNames* and *initialParamValues*. After the AAS has received the connection request, it will fetch the test, establish a test session and return the corresponding session identifier.

5.2 Communicating between AAS and LE

When the LE has established a connection with the AAS and has got its session identifier, the exchange of data can be started. Independent of the internal representation of the questions and tests and processes taking place, each AAS has to render and return questions in a format, which is generally known by the LE. Due to the fact that almost all LEs are web-based, the AAS should return data conforming to established web standards. In OPAQUE, the Web Service message, containing a question, is defined as follows:

```
<wsdl:message name="processResponse">
   <wsdl:part name="XHTML"
   type="soapenc:string"/>
   <wsdl:part name="CSS"
   type="soapenc:string"/>
   <wsdl:part name="resources"
   type="impl:ArrayOfResource"/>
   <wsdl:part name="progressInfo"
   type="soapenc:string"/>
   <wsdl:part name="questionEnd"
   type=" xsd:boolean"/>
   <wsdl:part name="results"      type="
impl:Results"/>
</wsdl:message>
```

The message parts *XHTML* and *CSS* are predestined to accommodate the HTML and the CSS representation of the question, respectively. Further needed resources like JavaScript libraries or the like can be included using the *resource*s message part. Now, the LE can compile the question using the several message parts and present it to the student. After the student has answered the question, the LE forwards the answer(s) to the AAS. In the following, the Web Service message used to return the answer(s) to the AAS for further processing are described: The message part *questionSession* is purposed to accommodate the session identifier of the assessment. The proper answer(s) can be included in the second and third message part. Afterwards, the AAS has to compare the received answer(s) with the correct answer(s) and decide how to proceed. In case of a correct answer, the AAS has to return the next question included in the test. In case of an incorrect answer, however, the AAS has to start its working. For example, personalized feedback could be used to guide the student to the correct solution or a slightly easier question addressing the same topic.

5.3 Completing the Test

Finally, the student answers all questions of a test and the test will be finished. Now, the AAS has to inform the LE about the results achieved by the student. This information is used by the LE for reporting and computing overall test scores. In order to inform the LE that the test is completely answered, t

he AAS can use the *questionEnd* message part included in every *processResponse* message (see 5.2). In this case the AAS simply has to set the value to true and the LE knows that there are no questions left and that the results of the test are included into the *results* message part dedicated to hold this information. Developed by the authors, a concluding sequence diagram of the interworking of student, LE and AAS is presented in Figure 1.

6 DISCUSSION

In the previous chapter several open standards and specifications and their interworking were presented aiming at realizing integration of AASs in LEs. By analyzing LEs and AASs important requirements could be extracted, which had great influence on the design of the interworking.

LEs require control information to determine the state of the student in the learning process. Thus, the interworking needs to be based on a session management, which enables pausing and resuming in case the student interrupts the assessment session or lost the connection to the LE. OPAQUE uses a session identifier to identify the requesting LE and provides control messages to start, process, and stop the assessment process. Among others, these are reasons why we propose to use OPAQUE as general interworking protocol. Moreover, although the LE passes the assessment responsibilities to the AAS, it needs to be informed how the student has performed in the assessments completed. Based on this information, the LE decides how to proceed in the learning process. Although OPAQUE only presents a structure dedicated to hold this information for one question, it is flexible enough to accommodate the student's result of an entire test. IMS QTI was also investigated to communicate of student results. It not only defines a format for representing and exchanging assessment content like questions and tests, but also for assessment results (IMS QTI Results Reporting package). The underlying data structure is similar to the one provided by OPAQUE, but the uncertainty as to whether an LE is able to process this response could become a problem.

AASs require student information from the LEs. This includes learning aspects, such as prior knowledge, learning preferences, learning styles or lifelong learning goals (i.e. career path). In chapter 4.1 candidate specifications including IEEE LTSC PAPI and IMS LIP were presented, which aims at presenting student information in a way that all LEs and AASs can understand. Although there are some standards for presenting this information, there is no common vocabulary, which enables a common understanding. Almost all AASs have their own mechanisms and vocabularies for storing student information. CosyQTI (Lazarinis et al., 2006), for example, uses the IEEE/ACM vocabulary to facilitate a common understanding. Such a vocabulary combined with a unified domain-modeling (see chapter 4.2) enables the AAS to derive information about the current level of knowledge of the student with respect to a specific

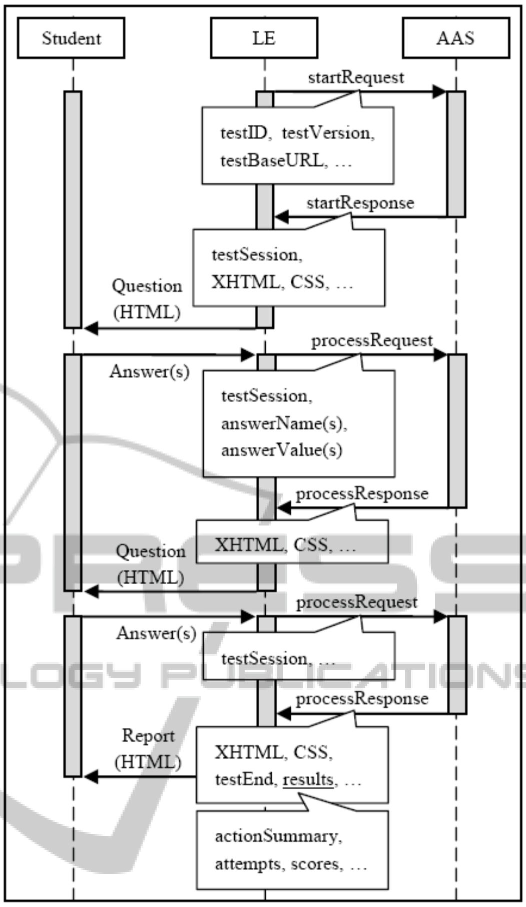

Figure 1: Sequence Diagram of Student, LE and AAS Interaction.

topic. Based on this understanding, efficient personalization by the AAS is possible. Moreover, after the test is taken, the results can be returned to the student-model by updating the competencies included. As presented in chapter 5.1, OPAQUE defines a message part to exchange student information between LEs and AASs. The returning session identifier can be regarded as key for further communication and identifies the assessment session. Another requirement is the uniform identification of the student. OPAQUE does not explicitly define a message part to identify the specific student. In the IMS LIP specification it is argued, that the source of the information record is responsible for the uniqueness of the student identifier. That would mean that the LE is responsible to uniquely identify the student, in order to track the learning process of the student. But, the AAS has also an interest in identifying the individuals, because it has to adapt the assessment process accordingly. For that reason, the LE is required to identify the student and to pass the student information to the AAS (see chapter 5.1). In addition, the AASs require assessment information from the LEs in order to identify the specific test. This includes the location of the questions, the test identification and possibly the version of the test. As presented in chapter 5.1, OPAQUE defines a Web Service message to initiate a connection request between LEs and AASs. Usually, questions are defined according to the IMS QTI specification and deposited in question banks.

7 CONCLUSIONS AND FUTURE WORK

The objective of this paper was to analyze how current open standards and specifications can be used to achieve integration between LEs and AASs seamlessly so that they can profit from each other. The analysis was caused by an understanding of the need of assessment adapted to the students' individual context, prior knowledge and preferences as well as the understanding that current LEs lack possibilities to provide personalized assessment to students. For that reason, both LEs and AASs were subject of a requirement analysis. The results of the analysis pointed out that LEs require control and assessment information from the AASs, which in turn require student and assessment information from the LEs. After having analyzed the requirements of both sides, related standards and specifications were studied in detail and matched against the requirements. As a result, an interworking of several standards and specifications were proposed, which could well be used to achieve integration between LEs and AASs.

Future work of the institution of the main author will implement a new AAS providing personalized assessment. The system not only selects and presents questions individually, but also takes sophisticated feedback techniques and methods resulting in providing feedback that is appropriate for the students' context, knowledge level, individual characteristics and preferences into account. The work provided in this paper helps integrating this system in established LEs, which, in turn, can contribute to a prompt and widespread adoption.

ACKNOWLEDGEMENTS

This work in this paper has partly been supported by the European Project Nº 216746 PReservation Organizations using Tools in Agent Environments (PROTAGE) and by the Fraunhofer Institute for Digital Media Technology, Ilmenau, Germany.

REFERENCES

Conejo , R., Guzman, E., Millan, E., Trella, M., Perez-de-la-Cruz, J., Rios, A., 2004. SIETTE: A Web-Based Tool for Adaptive Testing, In *International Journal of Artificial Intelligence in Education, 14*, 29-61.

Delius, G., 2005. RQP – Remote Question Protocol, http://maths.york.ac.uk/yorkmoodle/course/view.php?id=68

Gouli, E., Papanikolaou, K., Grigoriadou, M., 2002. Personalizing assessment in adaptive educational hypermedia systems. In *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, 153-163.

Hunt, T., 2008. OPAQUE – Open Protocol for Accessing Question Engines. http://docs.moodle.org/en/Development:Opaque

IMS LIP, 2005. Learner Information Package, http://www.imsglobal.org/profiles

IEEE PAPI, 2002. Public and Private Information, http://www.cen-ltso.net/Main.aspx?put=230

IMS QTI, 2006. Question and Test Interoperability, http://www.imsglobal.org/question

Lalos, P., Retalis, S., Psaromiligkos, Y., 2005. Creating personalised quizzes both to the learner and to the access device characteristics: the Case of CosyQTI. In *Proceeding of Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia*, 1-7.

Lazarinis, F., Retalis, S., 2006. Engineering an interoperable adaptive hypermedia testing tool supporting user adaptable strategies. In *Artificial Intelligence*, *204*, 393-400.

Lazarinis, F., Green, S., Pearson, E., 2009. Focusing on content reusability and interoperability in a personalized hypermedia assessment tool. In *Multimedia Tools and Applications, 47*, 257-278.

Saul, C., Runardotter, M., Wuttke, H.-D., 2010. Towards Feedback Personalisation in Adaptive Assessment, In *Proceedings of the Sixth EDEN Research Workshop*.

SOAP, 2007, Simple Object Access Protocol. http://www.w3.org/TR/soap/

XTM, 2001. XML Topic Maps, http://topicmaps.org/xtm/

Wuttke, H.-D., Ubar, R., Henke, K., Jutman, A., 2008. The synthesis level in Blooms Taxonomy a nightmare for an LMS. In *Proceedings of the 19th EAEEIE Annual Conference*, 199-204.