TECHNOLOGY-ENHANCED ASSESSMENT OF THINKING SKILLS IN ENGINEERING SCIENCES

A dissertation submitted to the
Fakultät für Informatik und Automatisierung
Technische Universität Ilmenau

for the degree of
Doktor-Ingenieur (Dr.-Ing.)
presented by

Dipl.-Inf. Christian Saul
born August 13, 1983
in Leinefelde, Germany

REVIEWER:
Prof. Dr.-Ing. habil. Andreas Mitschele-Thiel
Prof. Dr.-Ing. Dr. rer. nat. h.c. mult. Karlheinz Brandenburg
Univ.Doz. Dipl.-Ing. Dr. techn. Christian Gütl

DATE OF SUBMITTANCE: 2014/10/30
DATE OF DEFENSE: 2015/10/23

urn:nbn:de:gbv:ilm1-2015000483

ABSTRACT

Assessment is generally recognized as one of the most important elements of an educational experience. Since digital technologies found their way into assessment processes (referred to as technology-enhanced assessment or e-assessment), new possibilities for more personalized, immediate and engaging assessment experiences were opened up. However, especially in current times when sophisticated digital learning environments, mostly enriched by multimedia, virtual/augmented reality technologies, change the way what can be learned, when and how, the methods of assessing students' learning that have so far been developed are surprisingly limited. This can be demonstrated by the fact that current e-assessment practices simply imitate or replicate traditional pen-and-paper assessments. Consequently, new solutions are needed to identify, gather, analyze and interpret information about students' learning, especially considering the requirements of the 21st century.

In recognizing this need, this thesis proposes a novel architectural model for personalized and interactive e-assessment systems and tools. It allows integrating and using interactive and immersive tools (e.g., simulations or animations) into questions and tests, and enables tailoring them to students' individual characteristics (e.g., prior knowledge, context and preferences). While the former key feature (didactic interactivity) takes into account the assumption that learning is the result of interaction and the active engagement with the subject matter, respectively, the latter one (personalization) tackles the one-size-fits-all approach mostly applied in traditional e-assessment settings. Furthermore, the thesis describes the structure and the constituent components of the architectural model. A consistent user model, a generic domain model and a flexible adaptation model build up the central part of the overall model and represent the fundamental basis for the adaptive behavior. Each model is managed by an own component and has well-defined interfaces to each other. Additionally, the architectural model is complemented by a question modeling component responsible for representing (interactive) questions, responses, etc. and finally, an adaptive testing engine component that performs the actual adaptations. Moreover, this thesis presents the implementation of the architectural model by the web-based e-assessment system *askMe!*. It also describes how this system was trialed and evaluated from a pedagogical (learning support) and technical (usability and user experience) point of view.

The research and development performed in this thesis open up new opportunities for advanced e-assessment systems, which are able to consider the needs and characteristics of students and allow for more creativity in answering by interacting with digital tools in a variety of ways.

Keywords: E-Assessment, Personalization, Didactic Interactivity, Adaptive Assessment System, Rule-based Reasoning

ZUSAMMENFASSUNG

Assessment gilt allgemein als eines der wichtigsten Elemente in der Aus- und Weiterbildung. Mit dem Einzug digitaler Technologien in Assessment-Prozesse (auch als E-Assessment bezeichnet), wurden neue Möglichkeiten für personalisierte, unmittelbare und eindrucksvolle Erfahrungen beim Assessment eröffnet. In Zeiten, in denen hochentwickelte, digitale Lernplattformen die Art und Weise verändern, was, wann und wie gelernt werden kann, ist es umso verwunderlicher, wie eingeschränkt die vorhandenen Methoden für die technologie-gestützte Bewertung des Lernens sind. Deutlich wird dies durch die Tatsache, dass aktuelle E-Assessment-Systeme sich größtenteils auf das Replizieren von traditionellen Tests mit Stift und Papier beschränken. Folglich bedarf es neuer Lösungen für die Identifikation, Sammlung, Analyse und Interpretation von Informationen über das individuelle Lernen. Die Berücksichtigung der Anforderungen an das Lernen im 21. Jahrhundert spielt dabei eine entscheidende Rolle.

In Erkenntnis dieser Notwendigkeit präsentiert diese Arbeit ein neuartiges Architekturmodell für personalisierte und interaktive E-Assessment-Systeme und -Werkzeuge. Es erlaubt die Integration und Nutzung von interaktiven und immersiven Werkzeugen (z.B. Simulationen oder Animationen) innerhalb von Fragen und Tests, und ermöglicht diesen, sich an die individuellen Charakteristiken der Prüflinge (z.B. Vorwissen, Kontext und Vorlieben) anzupassen. Während das erste Hauptmerkmal (Didaktische Interaktivität) der Annahme gerecht wird, dass Lernen ein Ergebnis von Interaktionen sowie der aktiven Auseinandersetzung mit der jeweiligen Thematik ist, adressiert das zweite Hauptmerkmal (Personalisierung) die bei vielen E-Assessment-Systemen vorherrschende *one-size-fits-all* Strategie. Die Arbeit beschreibt die Struktur der grundlegenden Komponenten des Architekturmodells. Ein konsistentes Nutzermodell, ein generisches Domänenmodell sowie ein flexibles Adaptionsmodell bilden den zentralen Kern des Gesamtmodells und repräsentieren die Basis für das adaptive Verhalten. Komplettiert wird das Architekturmodell durch eine Komponente für die Modellierung von Fragen sowie einer Komponente für die Durchführung der spezifizierten Adaptionen. Darüber hinaus präsentiert die Arbeit die Implementierung des Architekturmodells durch das webbasierte E-Assessment-System *askMe!* sowie deren Erprobung und Evaluation nach pädagogischen (Lernunterstützung) sowie technischen (Gebrauchstauglichkeit und Nutzungserlebnis) Gesichtspunkten.

Die Ergebnisse dieser Arbeit eröffnen neue Möglichkeiten für zukunftsweisende (E-Assessment-)Systeme, welche in der Lage sind, die Bedürfnisse und Charakteristiken Einzelner zu berücksichtigen sowie mehr Kreativität bei der Beantwortung durch Interaktion mit digitalen Werkzeugen ermöglichen.

Schlagworte: E-Assessment, Personalisierung, Didaktische Interaktivität, Adaptive Assessment Systeme, Regelbasiertes Schließen

PUBLICATIONS

[SRW10] C. Saul, M. Runardotter, and H.-D. Wuttke. Towards Feedback Personalisation in Adaptive Assessment. In *Book of Abstracts & Workshop Programme Sixth EDEN Research Workshop*, pages 142–143, Budapest, October 2010.

[SDW11] C. Saul, F. Dürrwald, and H.-D. Wuttke. Towards a Generic Integration of Adaptive Assessment Systems with Learning Environments. In *Proceedings of the 3rd International Conference on Computer Supported Education (CSEDU 2011)*, pages 117–122, Noordwijkerhout, May 2011.

[SW11a] C. Saul and H.-D. Wuttke. Feedback Personalization as Prerequisite for Assessing Higher-order Thinking Skills. *European Journal of Open, Distance and E-learning (EURODL) - Special Issue: Best of EDEN*, pages 84–103, 2011.

[SW11b] C. Saul and H.-D. Wuttke. Personalized Assessment of Higher-order Thinking Skills. In *Proceedings of the CSEDU Special Session on Assessment Tools and Techniques for e-Learning (ATTeL 2011)*, pages 425–430, Noordwijkerhout, May 2011.

[SHLP11] C. Saul, P. Hofmann, M. Lucht, and P. Pharow. Competency-based Approach to support Learning Objectives in Learning, Education and Training. In *Proceedings of the 9. e-Learning Fachtagung Informatik der Gesellschaft für Informatik (DeLFI 2011)*, pages 21–30, Dresden, September 2011.

[HSW12] A. Heimbuch, C. Saul, and H.-D. Wuttke. Generic User Modeling for Adaptive Assessment Systems. In *Proceedings of the 4th International Conference on Computer Supported Education (CSEDU 2012)*, pages 187–192, Porto, April 2012.

[SW12] C. Saul and H.-D. Wuttke. Towards a high-level Integration of interactive Tools with E-Assessments. In *Proceedings of the 12th IEEE International Conference on Advanced Learning Technologies (ICALT 2012)*, pages 597–598, Rome, July 2012.

[SS12] A. Schulz and C. Saul. Adaptive E-Assessment Application Scenario: A Framework and Concept for the Integration of askme! in Higher Education. In *Proceedings of the 4th International Conference on Computer Supported Education (CSEDU 2012)*, pages 132–137, Porto, April 2012.

[SW13a] C. Saul and H.-D. Wuttke. E-Assessment meets Personalization. In *Proceedings of the IEEE Global Engineering Education Conference (EDUCON 2013)*, pages 200–206, Berlin, March 2013.

[SW13b] C. Saul and H.-D. Wuttke. An Adaptation Model for Personalized E-Assessments. *International Journal of Emerging Technologies in Learning (iJET) - Special Issue: EDUCON2013*, 8(2):5–12, 2013.

[SW13c] C. Saul and H.-D. Wuttke. Assessment 3.0 meets Engineering Sciences. In *Proceedings of the 16th International Conference on Interactive Collaborative Learning (ICL 2013)*, pages 623–630, Kazan, September 2013.

[HSL14] A. Heimbuch, C. Saul, and M. Lucht. Volition-driven Adaption - Following the individual Goals of Engineers in E-Learning Processes. In *Proceedings of the IEEE Global Engineering Education Conference (EDUCON 2014)*, pages 1032–1037, Istanbul, April 2014.

[SW14] C. Saul and H.-D. Wuttke. Turning Learners into effective better Learners: The Use of the askMe! System for Learning Analytics. In *Extended Proceedings of the 22th International Conference on User Modeling, Adaptation and Personalization (UMAP 2014)*, pages 57–60, Aalborg, July 2014.

Tell me and I forget,
teach me and I may remember,
involve me and I learn.

Benjamin Franklin

ACKNOWLEDGMENTS

There are a large number of people who have influenced my life and the path I have chosen. They all have, in their own way, made this thesis and work described herein possible. I am grateful for all their assistance and encouragement.

First of all, I want to thank Dr.-Ing. Heinz-Dietrich Wuttke for being my supervisor and mentor during my Ph.D. studies. I have been very fortunate to have the opportunity to learn from and work together with him. His astute insights, valued comments and suggestions have been of such great benefit for improving my work and in writing this thesis. Thank you so much for supporting me at all times in my studies.

I am also grateful to Prof. Dr.-Ing. Dr. rer. nat. h.c. mult. Karlheinz Brandenburg for giving me the opportunity to be a Ph.D. student at the Fraunhofer Institute for Digital Media Technology (IDMT), being my co-supervisor and providing me with an excellent research environment.

Furthermore, I would like to express my gratitude to my friends and colleagues (past and present) in the Data Representation and Interfaces Group at the IDMT. Their interdisciplinary research has had a significant effect on my studies. I also would like to especially thank Peter Pharow for supporting me all the time and for making the implementation of my research possible. Furthermore, I want to thank my master and bachelor students who implemented parts of the proposed model as well as all test persons who participated in our studies.

I am especially grateful to my family for their loving assistance, unconditional support and continued belief throughout all stages of my life.

Finally, I would like to express my deepest thanks to Franzi, my wife, for her endless support and encouragement, for taking care of me in busy times and for helping me going through the ups and downs during my Ph.D. studies. Without her love and patience I would never have finished this work.

Thank you all.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF LISTINGS

ACRONYMS

AAS	Adaptive Assessment System
ADL	Advanced Distributed Learning
AE	Adaptation Engine
AEHS	Adaptive Educational Hypermedia System
AHAM	Adaptive Hypermedia Application Model
AHS	Adaptive Hypermedia System
AI	Artificial Intelligence
AJAX	Asynchronous JavaScript and XML
AM	Adaptation Model
API	Application Programming Interface
CAA	Computer-assisted Assessment
CAT	Computerized Adaptive Testing
CBA	Computer-based Assessment
CM	Communication Mechanism
CS	Communication Schema
CSS	Cascading Style Sheets
DAML	Darpa Agent Markup Language
DIS	Distributed Interactive Simulation
DM	Domain Model
DOM	Document Object Model
FSM	Finite State Machine
GUI	Graphical User Interface
HLA	High Level Architecture
HOTS	Higher-Order Thinking Skills
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
GM	Goal & Constraints Model
IEEE	Institute of Electrical and Electronics Engineers
ICO	Interactive Content Object
ICT	Information and Communications Technology
IR	Information Retrieval
IRT	Item Response Theory
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
LE	Learning Environment
LCMS	Learning Content Management System
LIP	Learner Information Package
LMS	Learning Management System
LOTS	Lower-Order Thinking Skills
LTI	Learning Tools Interoperability
MRM	Munich Reference Model
MVC	Model-View-Controller

OCL	Object Constraint Language
OPAQUE	Open Protocol for Accessing QUestion Engines
OWL	Web Ontology Language
PAPI	Public and Private Information
PHP	PHP: Hypertext Preprocessor
PM	Presentation Model
QM	Question Model
QML	Question Markup Language
QTI	Question & Test Interoperability
RDFS	Resource Description Framework Schema
RQP	Remote Question Protocol
RTE	Run-Time Environment
SCO	Sharable Content Object
SCORM	Shareable Content Object Reference Model
SKOS	Simple Knowledge Organization System
SoC	Separation of Concerns
SVG	Scalable Vector Graphics
TENA	Test and Training Enabling Architecture
TM	Teaching Model
UEQ	User Experience Questionnaire
UM	User Model
UI	User Interface
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WWW	World Wide Web
XML	Extensible Markup Language

1 INTRODUCTION

This chapter is focused on presenting the general research agenda. First of all, Section 1.1 describes the motivation of the research. Then, Section 1.2 defines the research question and infers research goals for this thesis. Following up on this, Section 1.3 defines the scope of the work carried out and finally, Section 1.4 presents an outline of this thesis.

1.1 MOTIVATION

Today, learning occurs in a variety of places, not only within a teacher-student relationship (*formal learning*), but also at home, work and through daily interactions with today's society (*informal learning*). Whatever the environment of learning and method of delivery, it is crucial to obtain evidence about students' learning. It enables evaluating instructional materials and methods and thus helps improving teaching effectiveness. The measurement of learning outcomes is addressed through *assessment*.

In educational settings, assessment aims at assisting students' learning and enables identifying students' strengths and weaknesses. Furthermore, assessment also provides data that can support in making decisions [KK99]. Certainly, each of us has taken a large number of assessments at school. No matter whether they were examinations, tests or essays, they were mostly *pen-and-paper*-based and performed at a specific time and place (classroom). Moreover, they were characterized by a high formalization in terms of organization and administration and highly controlled in terms of content and marking. This kind of assessment practice performed from the beginning of the 20th century until now is referred to as *Assessment 1.0* [Ell07]. *Assessment 1.0*

Computers have found their way into assessment processes quite a long time ago. The term *e-assessment* has become widely used to describe the application of computers for assessment tasks. This includes the development and presentation of assessments as well as the recording of students' responses [QCA07, DW08]. Although e-assessments result in reduced economical costs (e.g., by cost savings in room and staff necessary for supporting and correcting, time savings in correcting the results and material savings through digitalization), they have often been criticized to simply imitate or replicate traditional *pen-and-paper* assessments [Ree00, BTP$^+$12]. Typically, e-assessment systems and tools provide a limited number of question types, which mostly require students to select an answer from a list of choices. But this kind of assessment (referred to as *Assessment 1.5* [Ell07]) does not allow students to be creative in answering and does not require or encourage them to actively thinking or problem-solving [Tho01]. Instead, it encourages surface learning and *"teaching to the test"* [Ell07]. Summarized it can be stated *Assessment 1.5* that current e-assessment systems and tools are good for what they were designed for: evaluating the acquisition of declarative knowledge.

But, learning in the 21st century aims at integrating and using knowledge and not just acquiring facts and procedures [FHP07]. Hence, e-assessment systems need to evaluate not just students' declarative (i.e., knowing *that* even numbers end with digits 0, 2, 4, 6 and 8), but also their procedural knowledge (i.e., knowing *how* to add two numbers), which is currently left to oral examinations or project work [Tha07, WH08]. This is a particularly serious problem in the case of engineering sciences, where students should be able to solve technical problems. For that, they have to analyze problems, evaluate solutions and often create a new whole [WUHJ08]. This coincides with the German Central Association for Electrical Engineering and Electronics Industry (ZVEI), who states that each graduate of an engineering study course should be able to identify, formulate, analyze and solve engineering problems [Die04]. These kinds of skills and capabilities students have to master cannot be measured using *pen-and-paper*-based assessments. What is needed are *sophisticated e-assessment tasks* [Boy05], which provide students the opportunity to demonstrate procedural knowledge by interacting with media-rich stimulus material in a variety of ways. Such stimulus materials are digital tools (e.g., simulations or animations in the form of HTML5, Java or Flash applets) that students can use to generate responses or analyze data. For that reason, assessments that make use of interactive and immersive tools, hereafter called as Interactive Content Objects (ICOs), are referred to as *interactive e-assessments* [Cri06]. In general, this kind of tool-assisted assessment practice is referred to as *Assessment 2.0* [Ell08].

Assessment 2.0

Assessment 2.0 describes e-assessments that are aligned with one of the characteristics of the Web 2.0 (i.e., *interaction*). It offers new opportunities for immersion and interactivity, which do not restrict students to be passive recipients. This allows students to actively discover concepts or subjects, to manipulate data, to examine the consequences and their responses and to make decisions about possible solutions. Strzebkowski and Kleeberg [SK02] have defined this kind of interactivity between a student and a computer as *didactic interactivity*. Besides, interactive e-assessments provide evidence for the development of students' advanced thinking skills as described by the higher levels of Bloom's taxonomy [BEF+56] or the SOLO taxonomy [BC82]. Over the time, a lot of ICOs in different disciplines have been developed suitable for sophisticated e-assessment tasks.

With the advent of the Web 3.0, another characteristic is becoming more and more interesting for e-assessments. It is the character *P* in Mitra's formula [Mit07] [1] for the future of the Web: Web 3.0 = 4C + P + VS (Content, Community, Commerce, Context + Personalization + Vertical Search). The Web 3.0 focuses on the individual and provides personalization through the use of *Semantic Web* and in general Artificial Intelligence (AI) technologies. Thus, the author of this thesis refers to the incorporation of personalization

1 http://www.sramanamitra.com/2007/02/14/web-30-4c-p-vs/
[last visited: May 21, 2014]

aspects into interactive e-assessments as *Assessment 3.0*. This new kind of assessment practice enables e-assessments that are tailored to students' individual knowledge and skills, and particularly engages them by considering individual aspects (e.g., context or preferences). The most common way to realize personalization in web-based systems is the use of adaptation methods and techniques [Bru96]. As the Web 3.0 is built on the data created by the Web 2.0, *Assessment 3.0* is also built on *Assessment 2.0*. That means, *Assessment 3.0* systems and tools enable both personalization and (didactic) interactivity. Following this naming, e-assessment systems and tools that only concentrate on personalization could be referred to as *Assessment 2.5* (cf. Figure 1.1).

Assessment 3.0

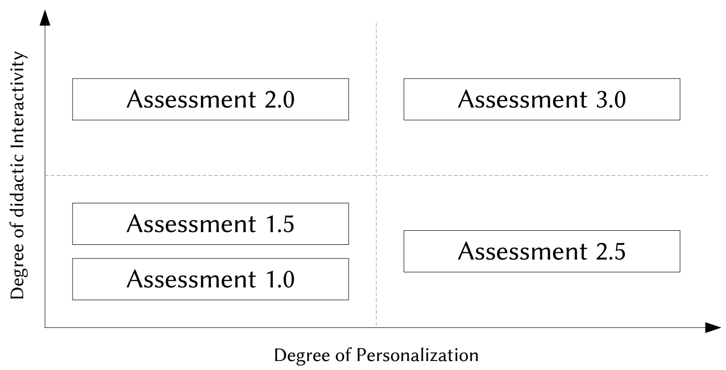


Figure 1.1: Comparison of the different assessment practices

Therefore, the challenge is to provide an architectural model for e-assessment systems and tools that enables integrating ICOs and allows complementing these by personalization aspects. The use of adaptation techniques will help students to develop advanced thinking skills by supporting at the beginning and then gradually turning over responsibility to the students to operate on their own [KW02b]. On the other hand, the use of ICOs within personalized e-assessments will provide valuable assessment findings that can be used to refine adaptivity decisions to be made. Integration in this respect not only means allowing ICOs to exist within a system or tool, but also to allow communication at a much deeper level to enable more efficient and effective personalized assessments of students thinking skills [SW11b]. However, a generic integration will not be simple. There are many different types of simulations or animations based upon different and often proprietary technologies, embedded in different types of Information and Communications Technology (ICT) infrastructure and implemented according to different pedagogical models.

1.2 RESEARCH QUESTION AND GOALS

Based on the benefits and limitations of today's e-assessment systems, the research question posed in this Ph.D. project is as follows:

RESEARCH QUESTION: *How can interactive e-assessment be enhanced with personalization aspects?*

In answering this research question, the primary objective of this Ph.D. project is to provide an architectural model that enables creating new types of questions, which enable integrating ICOs into the assessment process and consider students' individual aspects (e.g., prior knowledge, context and preferences). Hence, the outcome of this Ph.D. project provides a sound basis for the development of new e-assessment systems and tools that are compliant with *Assessment 2.0* and *3.0*. Therefore, the research goals are to:

G1 Analyze how existing e-assessment concepts, methods and systems can support the design of such an architectural model.

G2 Design of an architectural model that integrates personalization and didactic interactivity.

G3 Apply the architectural model by implementing it in an exemplary scenario.

G4 Perform an evaluation to validate the educational benefits of the architectural model and its implementation.

The realization of these activities involves the design and prototyping of a generic, flexible and extensible personalized e-assessment system. The approach seeks to enable the author/teacher to implement various methods of adaptivity as well as diverse pedagogical approaches (e.g., constructivism).

Currently, Learning Management Systems (LMSs) and Learning Content Management Systems (LCMSs) such as Moodle, ILIAS and OLAT are 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 by these systems. Hence, the integration and communication with established systems and tools is a prerequisite for a prompt and widespread adoption of the architectural model and its implementation. For this purpose, a further research goal is to:

G5 Examine how integration with established LMSs/LCMSs can be achieved.

1.3 SCOPE

This thesis deals with technology-enhanced assessment in the frame of educating engineers and natural scientists. In doing so, it solely focuses on *individuals* (e.g., teachers, authors, students) when defining or explaining concepts, methods or systems and does NOT deal with the assessment of *organizations* (e.g., the accreditation of colleges or universities). Moreover, the main focus is on assessing knowledge and its application in the domain of engineering sciences. These thinking skills are often referred to as *hard skills*. In contrast, the assessments of soft skills (e.g., communicative strength, the ability to work in a team, creativity) as it is often done in assessment centers is also NOT dealt with in this thesis.

However, this does not preclude the ability to apply the technologies, methods or concepts developed in other application domains nor does to use them to assess other skills, but it is not further discussed in this thesis.

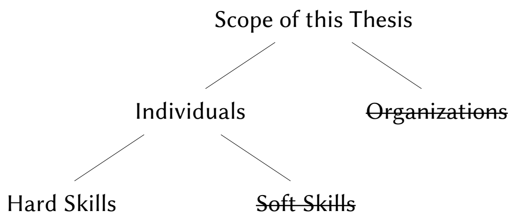


Figure 1.2: Scope of this thesis

1.4 OUTLINE OF THIS THESIS

This thesis is divided into seven chapters as follows:

CHAPTER 2: ASSESSMENT AND RELATED TERMS This chapter outlines the background of the research described in this thesis. The definitions lay the foundation for a common understanding of the theories, concepts and methods described and analyzed in the following chapter.

CHAPTER 3: STATE-OF-THE-ART CONCEPTS, METHODS AND SYSTEMS This chapter focuses on reviewing state-of-the-art concepts, methods and systems, which are relevant for answering the research question and achieving the research goals.

CHAPTER 4: DESIGN This chapter presents the novel architectural model for personalized and interactive e-assessment systems and tools developed in this thesis.

CHAPTER 5: IMPLEMENTATION This chapter describes the implementation of the architectural model by the web-based e-assessment system *askMe!*.

CHAPTER 6: EVALUATION This chapter focuses on evaluating the usability, user experience and educational benefits of the *askMe!* system.

CHAPTER 7: CONCLUSION This chapter gives a summary of the results of this thesis and clearly states their contributions. Furthermore, it discusses possible directions for future work.

2 ASSESSMENT AND RELATED TERMS

2.1 INTRODUCTION

This chapter focuses on presenting the background of the research described in this thesis. The definitions lay the foundation for a common understanding of assessment theories, concepts and methods. First of all, Section 2.2 defines the term *assessment*, points out the different roles and actors involved in the assessment process, explains the main principles, types and methods of assessment as well as presents learning objectives for assessment. Afterward, the terms *e-assessment* and *evaluation* are discussed (Section 2.3 and Section 2.4) and finally, Section 2.5 summarizes the main results.

2.2 ASSESSMENT

2.2.1 *Definition*

In general, assessment is defined by Erwin [Erw91] as *"systematic basis for making inferences about the learning and development of students"*. More specifically, he further defines assessment as the process of *"defining, selecting, designing, collecting, analyzing, interpreting, and using information"* in order to increase students' learning and development. This definition goes along with Astin [Ast93], who defines assessment as *"gathering of information [...] to facilitate student learning and development, to advance the frontiers of knowledge, and to contribute to the community, and the society"*. According to Angelo [Ang95], assessment involves *"making our expectations explicit and public; setting appropriate criteria and high standards for learning quality; systematically gathering, analyzing, and interpreting evidence to determine how well performance matches those expectations and standards; and using the resulting information to document, explain, and improve performance"*. Furthermore, Shepherd and Godwin [SG04] define assessment as *"systematic method of obtaining evidence from posing questions to draw inferences about the knowledge, skills, attitudes and other characteristics of people for a specific purpose"*. In accordance with the NSW Department of Education and Training [NSW08], assessment is the process of *"identifying, gathering and interpreting information about students' learning. The central purpose of assessment is to provide information on student achievement and progress and set the direction for ongoing teaching and learning."*. As shown, several authors dealt with the definition of assessment. Although they described the term in a number of ways, they often shared similar concepts and opinions. The key concepts derived from the definitions are summarized in the following definition, which will be used in this thesis.

DEFINITION: *Assessment is a systematic method comprising the process of identifying, gathering, analyzing and interpreting information about students' knowledge, skills, attitudes and other characteristics aiming at drawing inferences about their achievements and progresses as well as improving their learning and development performance.*

Even though this definition is very general, this thesis only focuses on the assessment of knowledge (cf. Section 1.3).

2.2.2 *Roles and Actors*

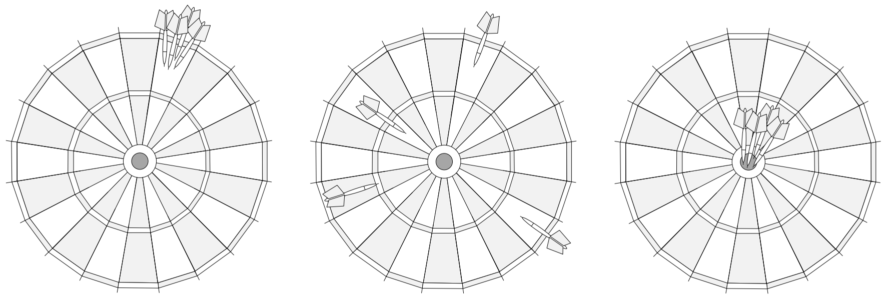
There are different roles and actors involved in the assessment process depending on the objective of the assessment. Sometimes, the terms are used interchangeably, but at least for this thesis, they have distinct meanings. A distinction is made between *authors*, *examiners* and *teachers* as well as between *learners*, *examinees* and *students*.

- *Author*: Someone who creates questions and tests. They have a look at the tests before students take them.
- *Examiner*: Someone who administers the test in order to determine students' knowledge. They have a look at the results during and after students took the tests.
- *Teacher*: Generic term for both an examiner and an author.
- *Learner*: Someone who takes the tests for self-assessment.
- *Examinee*: Someone who takes the tests for grading.
- *Student*: Generic term for both an examinee and a learner.

2.2.3 *Principles*

Reliability and validity are known as principles of assessment because they define the overall quality of assessment. The term reliability refers to the extent to which assessments are consistent. An assessment is reliable, when it measures the same thing consistently, that means it gives the same results under identical circumstances. In contrast, the term validity refers to the accuracy of an assessment. An assessment is considered as valid, if it measures what it is intended to measure [ACW95, McA02, SG04].

Reliability is a necessary, but not a sufficient condition for validity. That means, reliability is possible without validity, but validity is impossible without reliability [SG04, Woo09]. Figure 2.1 shows how reliability and validity are related to each other. The left dartboard (2.1a) shows that all the darts are stuck in the same field. With regard to a specific assessment, that means that the results are reliable and consistent, but not valid. On the dartboard in the middle (2.1b), the darts are distributed all around. This assessment is neither reliable nor valid because it is not consistent. The right dartboard (2.1c) shows an assessment that is both reliable and valid because all the darts are stuck together in the bull's eye.



(a) Reliable (consistent) but not valid (b) Not reliable (consistent) and therefore not valid (c) Reliable and valid

Figure 2.1: Relationship between reliability and validity (adapted from [SG04])

2.2.4 *Types*

There are many terms, which are used to describe different types of assessment. According to McAlpine [McA02], assessment can be categorized as *diagnostic*, *formative* and *summative*, *formal* and *informal*, *final* and *continuous*, *process* and *product*, *divergent* and *convergent* as well as *group-*, *peer-* and *self-assessment*. The categories that are important for this thesis are described in the following.

2.2.4.1 *Diagnostic, Formative and Summative Assessment*

Diagnostic assessment, also known as *pre-assessment*, is used prior to a learning activity to ascertain students' knowledge, skills and attitudes for the purpose of determining needs. The results can be used by teachers to assist them in developing learning activities and providing differentiated materials to meet students' needs. Diagnostic assessments can also be regarded as summative assessments of the previous learning activity. Examples of diagnostic assessments are the Diagnostic Online Reading Assessment (DORA[1]) and the Diagnostic Online Mathematics Assessment (DOMA[2]).

Formative assessment is used during the learning activity for providing feedback in order to inform the students of their current knowledge. The results can be used to discover what students have learned and where they still have weaknesses. Based on this information, future performance can be improved. Formative assessment is also called as assessment *for* learning. Examples of formative assessments include computer-based tests, which provide feedback on areas of strength and weakness as well as essays, which are annotated with teachers' comments.

1 `http://www.letsgolearn.com/lglsite/`DORA [last visited: May 21, 2014]
2 `http://www.letsgolearn.com/lglsite/`DOMA [last visited: May 21, 2014]

Summative assessment, also known as *post-assessment*, is used after the learning activity to judge the students' overall performance by measuring the level of success or proficiency and by comparing it against an expected standard. The results can be used for documenting and communicating students' knowledge, skills and attitudes. Summative assessment is also called as assessment *of* learning. Examples of summative assessments include IQ tests, traditional examinations and driver's tests.

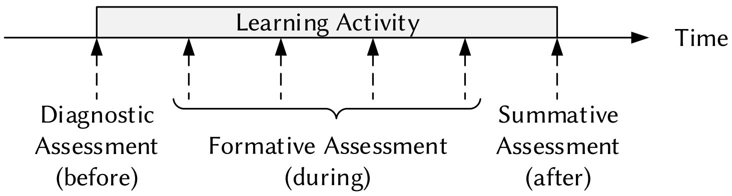


Figure 2.2: Diagnostic, formative and summative assessment

2.2.4.2 *Group-, Peer- and Self-Assessment*

Group-assessment focuses on evaluating one product, which is collaboratively produced by a group of students. Due to the fact that each student of the group is given a common mark, the marking burden for the teachers can be significantly reduced. The major problem of group-assessment is that students may not contribute equally. But, there are several strategies to address this problem [Rus01].

Peer-assessment involves students to evaluate the performance of other students, often using a predetermined list of criteria. This type of assessment is appropriate when assessing group work and is particularly valuable if both products and processes are assessed. Peer-assessment also reduces the marking burden for teachers and can help students in learning to evaluate their own learning and in interpreting assessment criteria [Rus01].

Self-assessment involves students to evaluate their own performance. It encourages students to reflect on their understanding of the subject matter, skills and processes, and can increase their motivation [Bou95].

2.2.5 *Methods*

As defined above, the term assessments in this thesis is used generic to describe any method of identifying, gathering, analyzing and interpreting information about students' knowledge, skills, attitudes and other characteristics (cf. Section 2.2.1). Such methods for assessing students' knowledge are *exams*, *tests*, *quizzes* and *surveys*. Although the terms are often used interchangeably, they differ significantly in terms of the purpose of measurement and the scope of content covered [Dav93].

- *Exams*: This assessment method is the most comprehensive method and typically given at the end of a learning activity (summative assessments, cf. Section 2.2.4.1). It measures knowledge for the purpose of *documenting* students' current level of knowledge [SG04].

- *Tests*: This assessment method is more limited in scope compared to exams and focused on a particular aspect of the learning material. It measures knowledge for the purpose of *informing* students about their current level of knowledge (diagnostic assessments, cf. Section 2.2.4.1) [Dav93, SG04].

- *Quizzes*: This assessment method is even more limited in scope than tests and covering only a small aspect of the learning material. It measures knowledge for the purpose of *providing feedback to inform* students about their current level of knowledge (formative assessments, cf. Section 2.2.4.1) [JC92, SG04].

- *Surveys*: This assessment method is diagnostically (cf. Section 2.2.4.1) and measures knowledge or opinions of a group of students for the purpose of *determining* needs that are required to fulfill a specific purpose, the effectiveness of learning material and any suggestions for potential change [SG04].

The models, methods and techniques to be developed in this thesis aim at supporting any of these methods.

2.2.6 *Learning Objectives*

As stated above, assessment aims at providing evidence about the achievement of *learning objectives*. A learning objective is an outcome statement that captures what knowledge, skills and attitudes students should be able to exhibit [Mag84]. Moreover, they also refer to *competencies* students can achieve through learning. Over the time, many researchers have attempted to identify and classify learning objectives and outcomes across the following three *learning domains*:

- *Cognitive*: This domain is concerned with knowledge.

- *Psychomotor*: This domain is concerned with the performance of skills.

- *Affective*: This domain is concerned with attitudes, feelings and emotions.

The classification (*taxonomy*) used as basis in this thesis is Bloom's taxonomy revised by Anderson and Krathwohl.

2.2.6.1 *Bloom's Taxonomy*

The most common and earliest taxonomy was provided by Bloom et al. [BEF+56]. In 1956, Bloom et al. created a hierarchy of intellectual skills (*thinking skills*) in the cognitive domain, known today as *Bloom's taxonomy*.

LEVEL	SAMPLE VERBS
Knowledge	Recognize, recall, identify, label, list, name, select, state
Comprehension	Interpret, match, estimate, explain, generalize, summarize
Application	Demonstrate, apply, modify, prepare, relate, show, solve, use
Analysis	Analyze, compare, infer, differentiate, distinguish, separate
Synthesis	Categorize, combine, create, construct, reconstruct, modify
Evaluation	Evaluate, appraise, judge, conclude, criticize, discriminate

Table 2.1: Indicating verbs for Bloom's taxonomy

Although Bloom's taxonomy also covers the psychomotor and affective domain, the cognitive domain has received most attention. The cognitive domain of Bloom's taxonomy distinguishes between six different levels namely:

1. *Knowledge*: This level requires students to recall and recognize terms and facts and their place in a particular domain.
2. *Comprehension*: This level requires students to inherit information from these terms by summarizing or interpreting.
3. *Application*: This level requires students to apply this information in new situations.
4. *Analysis*: This level requires students to separate parts of a whole and to understand the relationships in between.
5. *Synthesis*: This level requires students to combine parts to create a new whole.
6. *Evaluation*: This level requires students to make judgments based on criteria or standards through checking and critiquing.

The taxonomy is hierarchically structured, that means that learning at the lower levels (e.g., *knowledge*) must be achieved in order to master the higher levels (e.g., *evaluation*). Table 2.1 provides some verbs that indicate the different levels. An example how these verbs can be used to support the definition of learning objectives in learning, education and training can be found in Saul et al. [SHLP11].

More than 50 years later, Bloom's taxonomy of the cognitive domain was revised by Anderson and Krathwohl [AKA^{+}01]. Differences are the rewording of the levels from nouns to verbs, the renaming of some of the levels and the repositioning of the last two levels. But, the major differences are the addition of the type of knowledge being learned:

- *Factual*: This type of knowledge, also called as *declarative* or *descriptive* knowledge, is knowledge that is essential to specific disciplines.

- *Conceptual*: This type of knowledge is knowledge about the interrelationships among the basic elements within a larger structure that enable them to function together.
- *Procedural*: This type of knowledge is knowledge that helps students to do something.
- *Meta-cognitive*: This type of knowledge is knowledge of cognition in general as well as awareness of one's own cognition

In comparison to Bloom et al., Anderson and Krathwohl's revised taxonomy uses recent advancements in psychological and educational research (e.g., *constructivism*, *meta-cognition* and *self-regulated learning*) and thus is more generally applicable for specifying learning objectives and assessments [Ame06].

Two alternatives to the above mentioned taxonomies are the *SOLO Taxonomy* developed by Biggs and Collis [BC82] as well as the taxonomy of *learning outcomes* provided by Gagne [Gag85].

2.2.6.2 *Higher-order Thinking Skills*

In 1984, Imrie [Imr84, Imr95] firstly proposed dividing thinking skills into two tiers. The lower tier of his *RECAP model* covers all abilities of the lower three levels of Bloom's taxonomy for the cognitive domain (*knowledge*, *comprehension* and *application*), while the upper tier groups the upper three levels (*analysis*, *synthesis* and *evaluation*) together as *problem-solving skills*. While the former one focuses on thinking skills, that provide the basis for higher levels of learning such as *discriminations*, *cognitive strategies*, *comprehension* and (simple) *application* (cf. Section 2.2.6.1), and linked to prior knowledge of subject matter content, focuses the latter one on more advanced thinking skills such as *analysis*, *synthesis* and *evaluation* (cf. Section 2.2.6.1) [KGR98]. Consequently, the basic skills are referred to as Lower-Order Thinking Skills (LOTS), while the more advanced skills are referred to as Higher-Order Thinking Skills (HOTS) (cf. Figure 2.3). HOTS are crucial for advanced *cognitive processes* such as *problem-solving* [Duf91], *decision-making* [KT65], *critical thinking* [Lev97] and *creative thinking* [FM93].

Summarized it can be stated that learning (objective) taxonomies not only provide a good structure to assist teachers in writing learning objectives, but also in creating and categorizing assessment questions.

2.3 E-ASSESSMENT

In general, the term e-assessment is defined as the use of Information and Communications Technology (ICT) for any assessment-related activity. This includes the development and presentation of assessments as well as the recording of responses [QCA07, DW08]. Due to the fact that computers play an important role in ICT, the term e-assessment is becoming widely used to describe the application of computers within the assessment process.

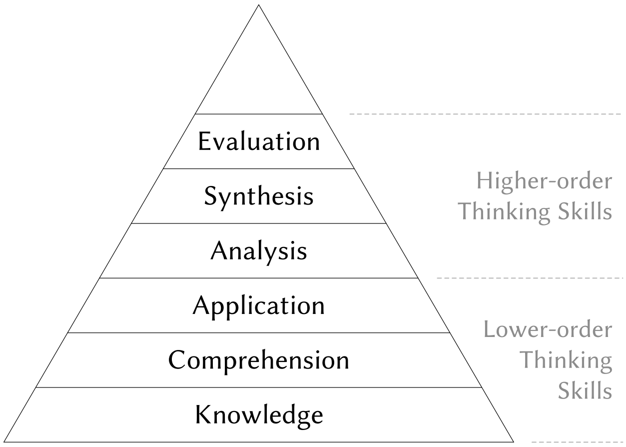


Figure 2.3: Lower-order and higher-order thinking skills in Bloom's taxonomy of the cognitive domain

However, the common term for the use of computers for the assessment of students is Computer-assisted Assessment (CAA) [BM03]. This includes collecting e-portfolios, constructing concept maps and Computer-based Assessment (CBA). In CAA, computers mainly facilitate the assessment process by, for example, recording and transferring responses between the students and the teachers. In CBA and in contrast to CAA, computers are not only used for recording and transfer, but also for marking responses. CBAs can be subdivided into stand-alone applications that only require a single computer, applications that work on private networks and those that are designed to be delivered across the web. The later one is known as online-assessments [CW05]. The different types of e-assessment are illustrated in Figure 2.4.

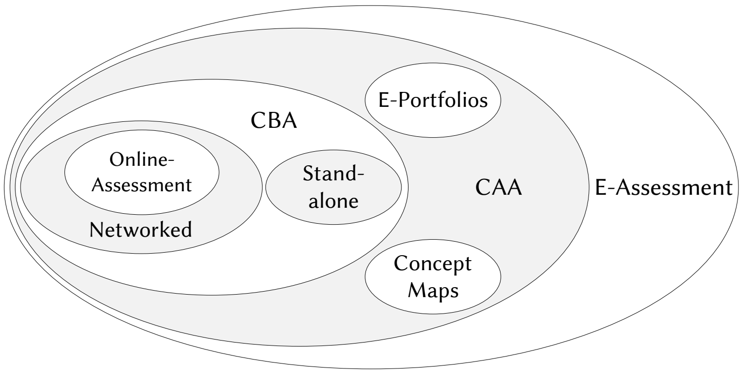


Figure 2.4: Different types of e-assessment (adapted from [CW05])

2.4 EVALUATION

The terms *assessment* and *evaluation* are often used interchangeably, but their meanings are different. While assessment is defined as identifying, gathering, analyzing and interpreting of information concerning students' learning (cf. Section 2.2.1), evaluation is defined as making judgments about students' learning. While assessment provides feedback on performance, strengths, areas for improvement and insights, evaluation only determines whether the predefined objectives were achieved or not. Assessment aims at improving the quality of future performances, whereas evaluation aims at determining the quality of the present performance [PFB+01].

2.5 SUMMARY

This chapter focused on developing a common understanding of the theories, concepts and methods described, analyzed and developed in this thesis. This included, of course, the definition of *assessment* as a systematic method of identifying, gathering, analyzing and interpreting information about students' knowledge. In order to assess their knowledge, different methods were distinguished such as *exams*, *tests*, *quizzes* and *surveys*. Moreover, in terms of the roles of the people involved in the assessment process, two main actors were identified: the *teacher* as the generic term for both an examiner and an author and the *student* as the generic term for both a learner and an examinee. Furthermore, the main principles (i.e., *reliability* and *validity*) and different types of assessment were briefly described. Finally, the term *evaluation* was differentiated from the term assessment.

3 STATE-OF-THE-ART CONCEPTS, METHODS AND SYSTEMS

3.1 INTRODUCTION

This chapter focuses on reviewing the state-of-the-art, which is relevant for answering the research question and achieving the research goals (cf. Section 1.2). First of all, Section 3.2 provides a general overview about the use of interactivity in e-learning settings. Following up on this, Section 3.3 investigates how interactivity is realized or supported by established e-assessment systems and tools as well as standards and specifications. The next two sections deal with personalization aspects. Due to the fact that the consideration of individual aspects in educational hypermedia is already far advanced, Section 3.4 gives an insight into established concepts, methods and methods. Based on this, the realization of personalization in e-assessment settings is analyzed and related to each other (Section 3.5). Finally, Section 3.6 summarizes the main results.

3.2 INTERACTIVITY IN E-LEARNING SETTINGS

3.2.1 *Introduction*

Interactivity is often being discussed as *"a good thing"* [Bat90] in e-learning settings because it allows *"learning-by-doing"*, arouses interests and generates motivation [Tho01, WH08]. Although its meaning has never been clearly defined [Bat90], it is widely agreed that students should not be passive recipients, but must respond in some way to the learning content in order to demonstrate that they have understood. Besides, feedback is also considered as an important component of interaction. It provides students with knowledge about the correctness of their responses and an indication how well they have learned. Thus, an interaction consists of a sequence of actions between two parties. Section 3.2.2 points out different types of interaction. The best way to provide complex and meaningful interactivity is to use *simulations*, which are addressed in Section 3.2.3. Afterward, three exemplary interactive and simulative systems and tools are presented and compared to each other (Section 3.2.4), and finally, several established standards and specifications in this field are discussed (Section 3.2.5).

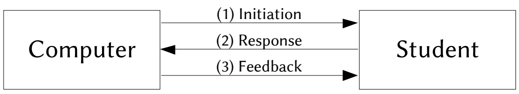


Figure 3.1: Three-way model of interactivity (adopted from [ES03])

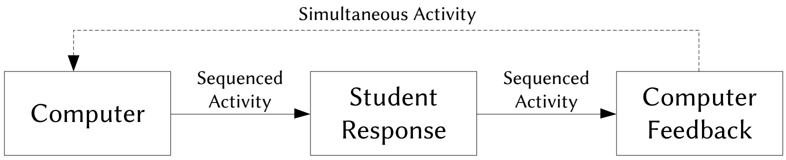


Figure 3.2: Interactivity cycle (adopted from [ES03])

3.2.2 *Types of Interaction*

In order to overcome the misunderstandings between teachers who use different media and to ensure maximum effectiveness, Moore [Mo089] provides three types of interactions:

- *Learner-content interaction*: This type of interaction is interaction between the student and the content or subject of study.
- *Learner-instructor interaction*: This type of interaction is interaction between the student and the expert who prepared the subject material.
- *Learner-learner interaction*: This type of interaction is interaction between the learner and other learners.

Each of these types can make use of computers to mediate the communication between the parties involved (e.g., through chat rooms or video conferencing software) or to access learning content (e.g., through a Learning Management System (LMS)). Moreover, Schar and Krueger [SK00] further subdivide the interaction between a student and the content into:

- *Student-initiated interaction*: In this type of interaction, the students require some information from the content and controls the sequence and speed.
- *Computer-initiated interaction*: In this type of interaction, the computer initiates some actions and requires some input from the students.

Based on the latter interaction mode, Evans and Sabry [ES03] propose a three-way model of interactivity consisting of a sequence of three actions namely *initiation*, *response* and *feedback* (cf. Figure 3.1). Each action involves a one-way flow of information between a computer and a student. The three actions can form an iterative interactivity cycle as shown in Figure 3.2. The computer gives feedback, but also simultaneously initiates another interaction.

Strzebkowski and Kleeberg [SK02] analyzed different LMSs and their possibilities of providing interactivity. As a result, they identified two groups of interactions:

- *Control interactions*: In this kind of interaction, the student passively interacts with the system (e.g., by selecting an audio file for playing).
- *Didactic interactions*: In this kind of interaction, the student actively interacts with the system (e.g., by controlling an interactive tool).

They define didactic interactions as the core of multimedia-based learning environments because they allow students to actively discover concepts or subjects. In e-learning settings, the best way to provide complex and meaningful (didactic) interactivity is to use (computer-based) *simulations* [Tho01].

3.2.3 *Simulations*

Simulations are programs that allow students to observe a real-world experience and to interact with it. They support the intuitive understanding of complex phenomena and are useful for simulating laboratories that are impractical, expensive or too dangerous to run [Sah06]. Thomas [Tho03] has defined two key features of simulations:

1. There is a computer model of a real or theoretical system that contains information on how the system behaves.
2. Experimentation can take place, that means changing the input to the model affects the output.

Having this key features, it is possible to distinguish what is and is not a simulation. Terms that are often associated with simulations are *animations* as well as *virtual* and *augmented reality*:

- *Animations*: An animation provide a series of images or a dynamic visualization. It only responds to preset values and does not behave according to a model. Thus, animations do not correspond to the definition of a simulation.
- *Virtual reality*: Virtual reality provides a real or imagined environment that can be experienced visually in the three dimensions. It can range from simple 3D images to complex flight simulators. Even though the former one does not include behavior, the latter one correspond to the definition of a simulation.
- *Augmented reality*: Augmented reality provides a scene that combines a real scene and a virtual scene generated by the computer and augmented with additional information. In simplified terms, it is a sophisticated data display (only the user's view changes) and thus does not correspond to the definition of a simulation.

Today, simulations are used for a variety of applications ranging from *research*, *design* and *analysis* to *education*, *training* and *entertainment* [Tho03]:

- *Research*: Simulations are used to establish trends, to demonstrate relationships between system parameters or to make predictions about the future.
- *Design*: Simulations are used to characterize or visualize systems, which do not yet exist.
- *Analysis*: Simulations are used to determine the behavior or capability of a system currently in operation or to verify its correctness.
- *Education*: Simulations are used to represent an exploratory world where students can conduct experimentation, create and test hypotheses and construct their own understanding of a system.
- *Training*: Simulations are used to allow trainees to practice a sequence of actions or to learn the correct response to an event.
- *Entertainment*: Simulations are used to create a consistent model of an imaginary world.

The application of simulations in education and training owes much to aviation training, which has half a century of experience and lessons learned with simulators and simulations [KW09]. The educational benefits of using simulations as supplementary materials to traditional learning content come from the ability of *"learning-by-doing"* and exploration [Tho03]. They allow students to manipulate input variables, to change the system behavior and to view the results. In this way, they can construct and test hypotheses and receive feedback as a result of their actions [Tho01]. Furthermore, simulations can contribute to conceptual change and provide tools for scientific inquiry and problem-solving experiences [Sah06].

3.2.4 *Systems and Tools*

This section briefly sets out three exemplary interactive and simulative systems and tools and compares them to each other.

3.2.4.1 *IrYdium Chemistry Lab*

The IrYdium project[1] at the Carnegie Mellon University has developed a teaching and learning tool that allows selecting from a variety of standard reagents and manipulate them in a way that is very much akin to that of a real laboratory. Moreover, it allows designing and performing diverse experiments in acid-base chemistry, thermo-chemistry and solubility.

1 `http://ir.chem.cmu.edu/` [last visited: May 21, 2014]

The chemistry lab[2] depicted in Figure 3.3 is divided into three panels. The panel on the left shows the stockroom explorer containing a list of solutions categorized into different categories. The panel in the middle shows the workbench of the lab, in which different glassware (e.g., flasks, beakers or pipettes) and tools (e.g., Bunsen burner, weighing boats or scales) are arranged. By simply dragging and dropping the different items on top of each others, several actions are performed (e.g., heating with a Bunsen burner or pouring a solution). The panel on the right shows various information (e.g., name, volume, concentration, etc.) about the solution currently selected.

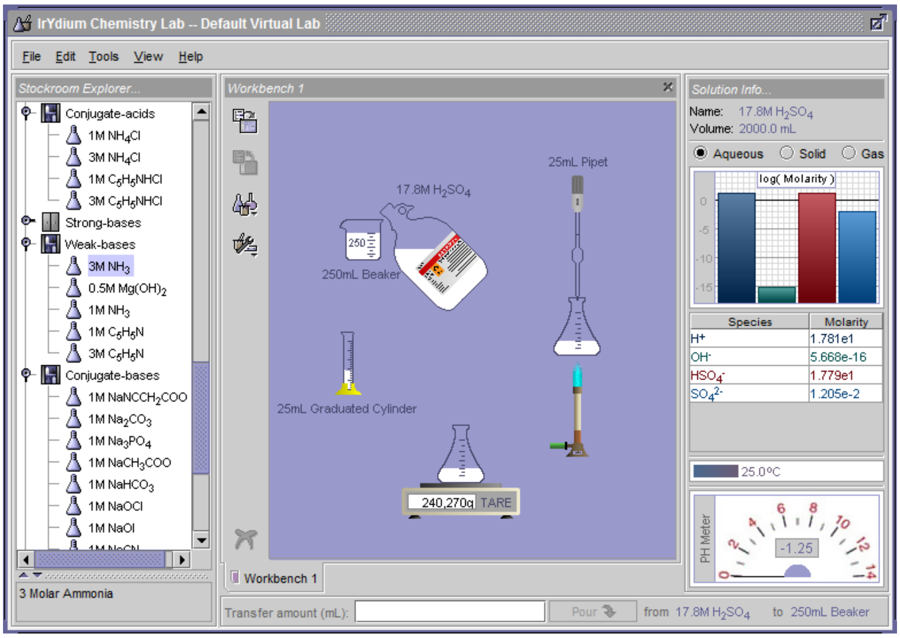


Figure 3.3: IrYdium chemistry lab

3.2.4.2 *Remote Engineering and Application Laboratory*

The Integrated Hard- and Software group at the Ilmenau University of Technology has developed a Remote Engineering and Application Laboratory[3], in which students can design, verify and implement digital circuits and control systems [WH08]. The lab consists of different tools that provide real-time experiments with hardware equipment or simulations to students. All of these models empower students to solve complex design tasks.

The elevator model depicted in Figure 3.4 combines both virtual and remote laboratory technologies and thus enables performing both real experiments and simulations to students. The model enables the controlling the elevator by entering control equations. Before using the simulation, the student is required to design a Finite State Machine (FSM) and to derive state-transition

2 `http://www.chemcollective.org/vlab/vlab.php` [last visited: May 21, 2014]
3 `http://ih7.theoinf.tu-ilmenau.de/applets/index.htm` [last visited: May 21, 2014]

and output functions out of it. These functions are entered in the four tabs on the left. The first tab lists and describes all input variables of the model. The second tab contains all output variables and provides input fields to define output functions for each variable. The third tab enables defining state-transition functions and initial states. The fourth tab enables defining banned configurations, which are not allowed and should never be reached. After entering an own or predefined (example) configuration, students choose whether they want to run the configuration live on the hardware equipment observable via the web cam or as a simulation within the applet without any connection to the remote lab. Using the simulation mode, students can additionally control the speed of the simulation by setting the clocks per second or the sensor distance. The model is controlled by the lift control on the right. It enables driving the elevator up- and downwards. The current driving direction and position of the elevator as well as whether it is overloaded are displayed under the lift control panel. Furthermore, the status bar indicates any error occurring during the input of the configuration.

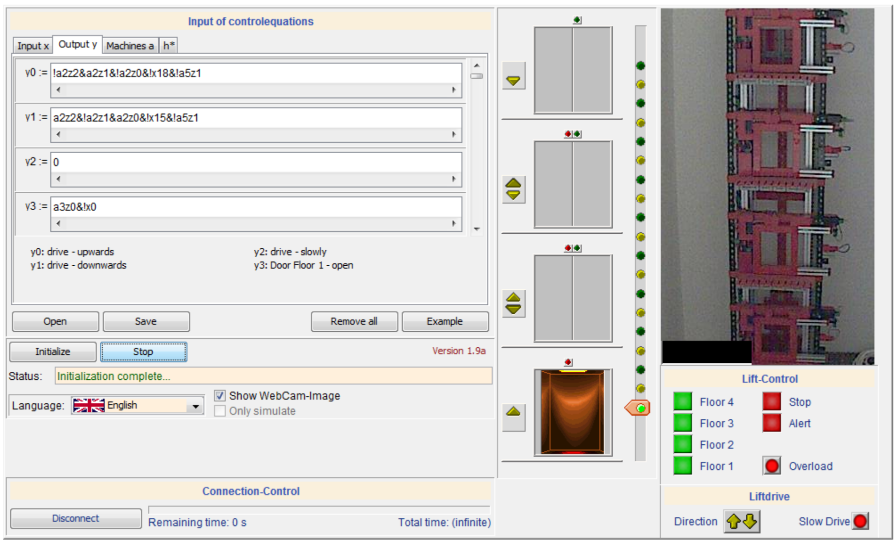


Figure 3.4: Remote engineering and application laboratory (elevator model)

3.2.4.3 *Basic Coordinates and Seasons Lab*

The Nebraska astronomy applet project at the University of Nebraska-Lincoln has developed numerous astronomy simulations and animations for introductory level courses in astronomy. They are grouped into several laboratories and complemented with background information or demonstration guides for students. The Basic Coordinates and Seasons Lab[4] is such a laboratory. It covers the areas of terrestrial and celestial equatorial coordinates, but also seasons and ecliptic.

4 http://astro.unl.edu/naap/motion1/motion1.html [last visited: May 21, 2014]

The seasons and ecliptic simulator[5] depicted in Figure 3.5 shows the geometry of the earth as it goes around the sun and demonstrates why seasons occur. The large panel on the left shows the relative positions of the earth and the sun. The orbit view shows the position of the earth in space as it revolves around the sun. The celestial sphere view shows what the same arrangement looks like to an earth observer, where the sun is projected onto the sphere of the sky. The panel in the upper right shows how the rays of the sun project onto the globe and the view from side shows the direction of the sun's rays relative to the earth. The lower right panel indicates the direction (sunlight angle) or intensity (sunbeam spread) of the sun's rays for an observer at a given latitude on earth. The latitude is selected in the upper right panel by dragging the stick figure or red latitude circle. The day of year for the simulator is indicated in the time line panel at bottom. The day can be changed by dragging the cursor or pressing the start animation button. The sub-solar point option places a marker on the globe indicating where the sun would appear directly overhead.

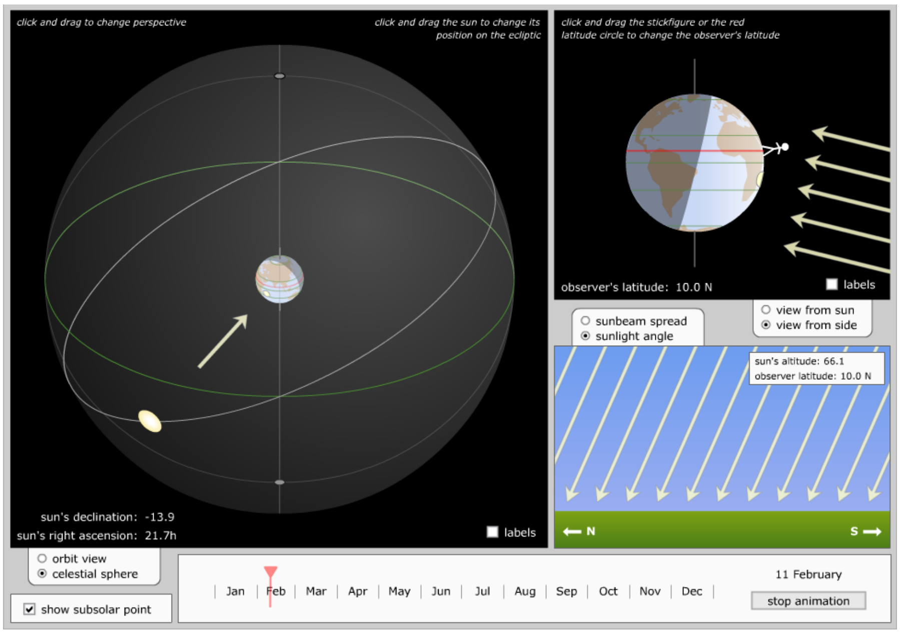


Figure 3.5: Basic coordinates and seasons lab (seasons and ecliptic simulator)

3.2.4.4 *Comparison*

In the previous sections, several interactive and simulative systems and tools were described and analyzed. Table 3.1 compares the different systems and tools to each other. The first criterion for the comparison concerns the type of the system or tool and distinguishes between simulation, animation, virtual reality and augmented reality as explained in Section 3.2.3. The next five criteria are inspired by Parshall et al. [PDP00] and whose categorization for

5 `http://astro.unl.edu/naap/motion1/animations/seasons_ecliptic.html` [last visited: May 21, 2014]

innovative question types. They propose a framework to arrange them along five dimensions:

1. *Item format*: This category refers to whether the student is required to select or construct a response.
2. *Response action*: This category refers to the physical action that a student makes to respond (e.g., by entering text using a keyboard or by clicking on a mouse).
3. *Media inclusion*: This category refers to the use of elements such as audio or video.
4. *Level of interactivity*: This category refers to the extent to which items react or respond to student input (i.e., low, medium or high).
5. *Scoring algorithm*: This category refers to how student's response is translated into a quantitative score (i.e., dichotomous or polytomous).

The level of interactivity ranges from low to high. Minimal (low) interactivity is given, for example, when the tool only provides a highlighted or shaded display of the response option selected by the student. At the next level of interactivity (medium), a student acts and the tool then responds with some sort of reaction or informative feedback. The highest level of interactivity is provided when the interactivity occurs in the situated task or within a realistic situation.

In addition to the second category (response action), Sim et al. [SHB04] propose four *response formats* based on the human interaction technique required:

- *Point and click*: This format requires students to select objects on the screen.
- *Move object*: This format requires students to move objects to predetermined positions on the screen.
- *Text entry*: This format requires students to input short answers.
- *Draw object*: This format requires students to draw objects or lines.

As a result, the comparison reveals that the systems and tools considered almost all show a high degree of interactivity caused by a high number of possible (response) actions and system/tool reactions provided. Although they all allow students actively interacting with the system, however, they were not seen in the context of assessment so far. This becomes particularly obvious by the lack of scoring algorithms in these systems and tools.

		Irydium Chemistry Lab	Remote Engineering and Application Laboratory	Basic Coordinates and Seasons Lab
Type		Simulation	Simulation	Animation
Item Format		Construct	Construct	Select
Response Action	Point and Click	x	x	x
	Move Object	x	–	x
	Text Entry	–	x	–
	Draw Object	–	–	–
Media Inclusion	Graphics	x	x	x
	Audio	–	–	–
	Video	–	x	–
	Animation	–	x	x
Level of Interactivity		high	high	medium
Scoring Algorithm		–	–	–

Table 3.1: Comparison of interactive and simulative systems and tools

3.2.5 *Standards and Specifications*

There are several standards and specifications related to simulations. The most widely used are:

- *Distributed Interactive Simulation* (DIS): Network protocol standard for exchanging information among various simulations and allows geographically separated simulations to interact in real-time.
- *High Level Architecture* (HLA): General purpose architecture developed to facilitate the reuse and interoperability of distributed simulations.
- *Test and Training Enabling Architecture* (TENA): General purpose architecture developed to enable interoperability among range systems, facilities, simulations, etc. and to foster reuse for range assets.

All of these standards and specifications share the fact that they were originally designed for military applications. This becomes particularly obvious when looking at them more closely. For example, the DIS standard specifies data messages that are exchanged between simulation applications. They are arranged into different protocol families such as warfare, radio communications and minefield. Even though these specifications are strongly shaped by their application domain, De Penning et al. [DBK08] stated that they could also be used in e-learning settings.

3.2.6 *Summary*

This section has shown that an interaction in e-learning settings consists of a sequence of actions between a student and a computer or another student. It requires students to respond in some way in order to demonstrate what they have learned. This results in feedback and can simultaneously initiate another interaction. On the contrary, a system or tool that allows students to simply navigate through the content cannot be referred to as interactive. Over the time, a large number of interactive and simulative systems and tools were developed covering a wide range of application domains. Three exemplary systems and tools were given and compared to each other. Finally, a brief look at standards and specifications for interactive and simulative systems was taken.

CONCLUSION: Relevant for this thesis is the fact that the interactive and simulative systems and tools available almost all provide a high degree of interactivity, but they were not seen in the context of assessment so far. This once again confirms the need for a general way to integrate and use these systems and tools for assessment. In addition, the standards and specifications available are not suitable for their use as basis because they were originally designed for military applications and thus strongly shaped by this domain. This makes their use in e-assessment settings difficult, not to say impossible.

3.3 INTERACTIVITY IN E-ASSESSMENT SETTINGS

3.3.1 *Introduction*

Following up the general insight into interactivity in e-learning settings, this section investigates how interactivity is provided or supported by established e-assessment systems and tools (Section 3.3.3) as well as how it is supported by established standards and specifications (Section 3.3.4). But first of all, the term *interactive e-assessment* is discussed (Section 3.3.2).

3.3.2 *Interactive E-Assessments*

The demand for more interactivity in assessments reaches back more than 20 years. As early as 1988, Bunderson et al. [BIO88] suggested that sophisticated assessment methods such as simulations and case studies will become common in the future. Ten years later, Bennett [Ben98] took up these ideas as his vision of the future of assessment. He pointed out that e-assessment has not yet achieved its full potential and predicted a dramatically improvement in using simulations and virtual reality while assessment. In 2005, Mackenzie [Mac05] stated that the so-called *sophisticated e-assessment tasks* go beyond simple multiple-choice or multiple-response question types into the area of complex question types, scenarios and simulations of real life situations or problems. Even though it was generally agreed that sophisticated e-assessment tasks *"make use of the characteristics of the new digital educational environment, namely interactivity"* [Cri10], a common definition is unlikely to be found. However, Boyle [Boy05] identified two core features of sophisticated e-assessment tasks:

Sophisticated E-Assessment Tasks

- They contain media-rich stimulus material
- They require student to interact with the stimulus material in a variety of ways

This kind of stimulus material that enables students to interact is hereafter called as Interactive Content Object (ICO). It is defined as follows:

Interactive Content Object

DEFINITION: *An ICO is an interactive and digital tool that enables students to actively discover concepts or subjects, to conduct experiments by manipulating data and observing the effects of change and to put in and test hypotheses.*

Not only simulations (cf. Section 3.2.3) are falling into this category, but also animations and virtual reality. All the systems and tools mentioned in Section 3.2.4 can be referred to as ICOs. The use of these tools in e-assessment settings, also referred to as *interactive e-assessments* [Cri06], offers students more creativity in answering by interacting with the tool (e.g., by generating and not only by selecting answers). This allows the assessment of advanced thinking skills more effectively than traditional methods [Boy05]. Advanced

Interactive E-Assessments

thinking skills refer to the higher levels of Bloom's taxonomy (cf. Section 2.2.6.1). The limitations of traditional assessment methods for assessing these skills are well documented [WH08]. In contrast, interactive e-assessments allow students to manipulate data, to examine the consequences of their responses and to make informed decisions about potential solutions. This does not only requires, but also encourages students to use their advanced thinking skills.

Constructivism

Besides, this kind of assessment practice coincides with the *constructivist learning theory* [Fos05]. The theory argues that learning involves own knowledge and own experiences and promotes students' free exploration within a given structure to solve realistic problems. Students are expected to make decisions and reflect on the consequences of those.

3.3.3 *Systems and Tools*

This section briefly sets out six established e-assessment systems and tools and compares them to each other. The selection was based on the level of popularity and innovation.

3.3.3.1 *Questionmark Perception*

Questionmark Perception[6] is an assessment management system that enables authoring questions and organizing them into surveys, quizzes, tests and exams. Questions can be authored through a Microsoft Windows application or a web interface and delivered online through a web browser, offline on a CD or any other medium or on paper. The first version of Questionmark Perception was demonstrated in 1995. The following explanations are based on version 5.4.

Question Types

For creating assessments, Questionmark Perception allows using several assessment options such as open access, time limit or anonymous results. Moreover, there are four predefined assessment types namely *exams*, *tests*, *quizzes* and *surveys*, which preselect a subset of these options. Each of them differs in a variety of restrictions for the students (cf. Section 2.2.5). Currently, the system allows creating 22 question types ranging from drag and drop to Adobe Captivate. In order to control how a student proceeds through an assessment, Questionmark Perception allows defining *jump blocks*. This allows question blocks within an assessment to be skipped or retaken. A jump can be unconditional or conditional based on outcomes, the score in the previous block or the score in the assessment. The system not only allows defining feedback for the correct and incorrect answer, but also for each answer option. Feedback can not only include text, but also images, sound, equations, videos and Adobe Flash applets. Furthermore, it can be defined to which questions feedback should be displayed (e.g., to all, to wrong only or to wrong and unanswered). Moreover, it can also be defined, when

Feedback

6 `http://www.questionmark.com/us/perception/`
[last visited: May 21, 2014]

feedback should be given, either after each question block or at the end of the assessment. Questionmark Perception is compliant with the IMS Question & Test Interoperability (QTI) v1.2 specification. That means, questions created with the authoring tool can be exported from and imported to IMS QTI files. Moreover, the system's own format called Question Markup Language (QML)[7] can also be used for importing and exporting questions. Besides, questions can also be imported as ASCII files using specific ASCII import definitions. For importing and exporting assessments, the system is limited to its own *Qpack* format.

Interoperability

In conclusion, particularly noteworthy on Questionmark Perception is the fact that it allows embedding Adobe Flash applets and Adobe Captivate simulations when creating questions. But, in order to do that, it requires to use the QML format. Unfortunately, no other e-assessment system or tool makes use of or enables importing this markup language. Moreover, the system only allows the Flash applet or Captivate simulation to communicate whether the student has passed or failed the question or task. Thus, intermediate results or partial solutions cannot be communicated. Also worth mentioning on Questionmark Perception is the ability to adapt the sequence of the questions by using *jump blocks*. Although jump blocks allow question blocks to be skipped or retaken, they are limited to adapt to question/test results and can not consider the characteristics of individual students.

3.3.3.2 *Respondus*

Respondus[8] is an assessment management system that enables creating questions and managing assessments. Questions can be authored through a Microsoft Windows application and directly delivered to different LMS and Learning Content Management System (LCMS) (e.g., Blackboard, Moodle or Desire2Learn) or on paper. In 2000, Respondus, Inc. released Respondus Lite 1.0, which enabled teachers creating assessments offline and transferring them into WebCT courses. In 2001, the lite version was replaced by a full-featured version. The following explanations are based on version 4.0.

For creating assessments, Respondus provides 15 question types and several assessment options such as multiple attempts and the ability to make the assessment accessible only for a fixed group of students. The system allows defining feedback for all question types. Feedback can not only include text, but also images, animations, audio and video files, and equations. In general, Respondus distinguishes between *general* and *specific feedback*. While general feedback does not depend on the students' response, specific feedback is linked to a particular answer option and will appear next to that option. The system allows publishing and retrieving exams and assessments and their associated media files directly to and from various LMS/LCMS server (e.g., Blackboard, Moodle or Desire2Learn). It also allows importing questions in plain text, rich text, word, tab/comma delimited and StudyMate

Question Types

Feedback

Interoperability

7 `http://www.questionmark.com/qml/` [last visited: May 21, 2014]
8 `http://www.respondus.com/` [last visited: May 21, 2014]

Class format. Furthermore, Respondus also allows importing and exporting questions according to the IMS QTI v1.2 specification.

In conclusion, noteworthy on Respondus is the fact that it not only enables creating questions and managing assessments, but also allows publishing and retrieving exams and assessments and their associated media files directly to and from various LCMSs and LMSs (e.g., Blackboard, Moodle or Desire2Learn). This is advantageous if, for example, the LCMS or LMS is not compliant to specific standards or specifications. However, the system does not provide any question type that allows actively interacting with the system.

3.3.3.3 *TATS*

TATS[9] is an open-source and web-based assessment tool that enables creating questions and compile them into tests. For rendering questions and tests, it uses the R2Q2[10] web service. TATS has been developed by the Centre for Educational Technology, a research unit within the Institute of Informatics at the Tallinn University. The first version (v0.1) of TATS was release in 2007. The following explanations are based on version 0.8.

Question Types

TATS allows defining question using 11 different question types such as multiple-choice or hottext. Furthermore, each question can be enriched with meta-data (e.g., language, difficulty, subject and license). Questions are organizing into tests, which can also be enriched with meta-data. Moreover, TATS allows grouping students in order to send tests to groups. The results can be viewed later on students as well as on group level. In terms of feedback, TATS only allows defining textual feedback for multiple-choice questions. In doing so, feedback can be defined for the correct and incorrect answer, but also for each answer option. With respect to interoperability, all questions and tests created in the system can be exported according to the IMS QTI v2.1 specification.

Feedback

Interoperability

In conclusion, particularly worth to mention on TATS is the fact that it allows questions and tests to be enriched with meta-data such as the language, difficulty, subject or license. However, the meta-data is not used to adapt the tests to the individual student. In addition as just like Respondus, TATS do not provide any (didactic) interactive question types.

3.3.3.4 *ILIAS*

ILIAS[11] is an open source web-based and Shareable Content Object Reference Model (SCORM)-compliant LMS/LCMS. It not only supports learning content management, but also provides tools for collaboration, communication, evaluation and assessment. ILIAS has been developed by the University of Cologne. The first prototype of ILIAS was released at the end of 1997. In 1998, the first official version of ILIAS was published and offered for learning at the

9 `http://ait.opetaja.ee/tats/` [last visited: May 21, 2014]
10 `http://www.r2q2.ecs.soton.ac.uk/` [last visited: May 21, 2014]
11 `http://www.ilias.de/` [last visited: May 21, 2014]

Cologne faculty of business administration, economics and social sciences. The following explanations are based on version 4.3.

For creating assessments, ILIAS provides several assessment options such as anonymity, maximum processing time, kiosk mode, etc. With the use of these options, a variety of assessments can be defined such as self-assessment, anonymized and randomized tests. ILIAS allows creating 12 question types such as multiple-choice and ordering, but also Java and Flash. In terms of feedback, ILIAS not only allows defining feedback for the correct answer, but also for the case when at least one answer was is not correct. Moreover, feedback for each answer option can also be defined. Feedback is not limited to text, but can also include images, equations and Hypertext Markup Language (HTML) code. All the questions and tests created in the system can be exported according to the IMS QTI v1.2 specification. In addition, an export as Microsoft Excel file is also provided, but it is only a simple overview about all questions in the question pool. ILIAS supports the import of IMS QTI files, which were prior exported from ILIAS. A generic IMS QTI import is not integrated in the current version.

Question Types

Feedback

Interoperability

In conclusion, worth mentioning on ILIAS is the fact that it allows embedding Adobe Flash and Java applets. The applet offers the question and the logic to communicate with ILIAS. Compared to Questionmark Perception, the applet is able to communicate several parameters (e.g., user solution or the reached points for the given solution) back to ILIAS. But, this information is only used by ILIAS for grading purposes and not, for example, to give formative feedback or to adapt subsequent questions.

3.3.3.5 *Moodle*

Moodle[12] is an open source web-based and SCORM-compliant LMS/LCMS that can be used in many types of environments such as in education, training and development as well as in business settings. Moodle was originally developed by Martin Dougiamas with the aim of helping teachers creating online-courses. The first version (Moodle 1.0) was released in 2002. Major improvements in accessibility and display flexibility were developed in the meantime. The following explanations are based on version 2.5.

For creating assessments, Moodle also provides several assessment options such as time limits, restricting the number of passes for an assessment, etc. Using these options, Moodle provides two main forms of assessment namely *assignments* and *quizzes*. Assignments allows teachers to collect work from students, review it and provide feedback including grades, whereas quizzes allows teachers to design and build assessments consisting of a set of questions. Currently, Moodle allows creating a variety of different question types to set up assessments. In general, it distinguishes between *standard question types* such as multiple-choice, numerical and matching, and *third-party question types* such as drag and drop, molecular editor and OPAQUE. The most interesting one is OPAQUE. It is less a question type than a communication protocol that allows LCMSs/LMSs to delegate the presentation

Question Types

12 `http://www.moodle.org/` [last visited: May 21, 2014]

of questions, the scoring of responses and the generation of feedback to a remote testing/question engine such as STACK[13] and OpenMark[14]. With regard to feedback, Moodle can display feedback at different times during the assessment. Feedback can not only include text, but also images, animations, audio and video files, and equations. In general, the system distinguishes between *general*, *overall*, *specific* and *combined feedback*. For each of the first three feedback types, the time can be defined, when the students will see this feedback. Furthermore, Moodle allows defining hints, which can clear incorrect responses or show the number of correct responses. These hints can be used, for example, in the adaptive mode before asking the student to try again. Until Moodle 2.1, questions created could be exported according to the IMS QTI 2.0 specification. This export format is no longer available since Moodle 2.2. But, the export in a Moodle-specific text or XML format is provided. For importing questions, Moodle has a variety of file formats that can be used. This includes some LCMS-specific question formats (e.g., the Blackboard V6+ format, the Examview format or the WebCT format) as well as the Moodle-specific text and XML format.

Feedback

Interoperability

In conclusion, noteworthy on Moodle is the fact that it integrates several third-party question types contributed by the Moodle community. Worth mentioning is the molecular editor, which allows designing molecular structures. However, this editor is not open-source and needs to be installed separately. Furthermore, the possibility of Moodle in providing feedback is outstanding in comparison with the other systems or tools. Moodle distinguishes between *general*, *overall*, *specific* and *combined* feedback. Feedback can not only include text, but also images, animations, audio and video files, and equations.

3.3.3.6 *OLAT*

OLAT[15] is an open source web-based and SCORM-compliant LMS. It focuses on presenting learning and training content and does not provide extensive authoring tools for creating or editing learning content. However, externally created learning content can be imported into OLAT using standardized interfaces. The development of OLAT started in 1999 at the University of Zurich. In 2004, version 3 was completely rebuild in Java due to performance and scalability issues of the previous PHP implementations. The following explanations are based on version 7.7.

For creating assessments, OLAT provides three predefined assessment types namely *test*, *self-test* and *questionnaire*. They differ in terms of the number of attempts and the recording of the results. OLAT allows creating five question types namely *single-choice*, *multiple-choice*, *kprim*, *cloze* and *free-text*. In terms of feedback, the system allows defining feedback for all correct, but also for wrong answers. Moreover, it allows defining feedback for each answer option. Additionally, feedback is not limited to text, but

Question Types

Feedback

13 http://stack.bham.ac.uk/ [last visited: May 21, 2014]
14 https://java.net/projects/openmark [last visited: May 21, 2014]
15 http://www.olat.org/ [last visited: May 21, 2014]

can also include images, animations, audio and video files, and equations. Regarding the interoperability of questions and tests, OLAT enables importing and exporting (self-)tests and questionnaires according to the IMS QTI v1.2 specification. Moreover, IMS QTI v2.1 is also supported by using the ONYX[16] plug-in.

Interoperability

In conclusion, particular noteworthy on OLAT is the fact that it provides standardized interfaces for integrating authoring tools and importing learning content into the system. An example of a plug-in that makes use of these interfaces is ONYX. But, with respect to didactic interactivity, OLAT is limited to traditional question types.

3.3.3.7 *Comparison*

In the previous sections, several established e-assessment systems and tools were described and analyzed according to the assessment functionality (i.e., question types, feedback and interoperability) provided. Figure 3.6 now classifies the question types provided by the different systems and tools according to the levels in Bloom's taxonomy (of the cognitive domain) (cf. Section 2.2.6.1) they are able to assess. A description about the different question types can be found in Appendix A. As it can be seen in the figure, the majority of question types are located at the lower two levels and only require students to recall content in the exact form it was presented (knowledge) or to recognize previously unseen content (comprehension). Simply put, the students do not need to write any words or sentences, they just select their answer from a list of choices. These question types are referred to as *constrained*. In contrast, question types whose response is open and needs to be created by the student is referred to as *constructed* [AGK+09]. All question types in the figure above the second level can be assigned to this group. As shown in the figure, a small number of question types is assigned to the fourth level in Bloom's taxonomy (which also means that they are able to assess the levels below too). This means that they are (theoretically) able to assess the application of knowledge, but only by simply checking the answer to a question/task filled in an input field (blank). The preceding elaboration processes that led to this answer took place outside the system or tool and cannot be recognized and used for evaluating the answer. But, this is needed in order to make reliable statements about the competencies achieved. In conclusion, these question types do not really assess the levels they intend to do so. This is very similar to the question type essay and its derivatives (i.e., extended text, free text, text-subset and spoken response) located at the highest level. They are about the same in that the student also only enters the answer (even if it is a more comprehensive text), but with the difference that the answer is mostly not evaluated by the system itself. Instead, the answer is forwarded to the teacher for evaluating. File uploads are handled similarly. In this case, the computer moves into the background. However, the situation is quite different with the other question types on the upper level.

16 `http://onyx.bps-system.de/` [last visited: May 21, 2014]

The figure shows that both Questionmark Perception and ILIAS provide question types that enable embedding external tools such as Adobe Flash or Java applets. But, these opportunities are limited to the respective system or tool. As an example, in addition to the Adobe Flash file, Questionmark Perception needs a QML file for integrating Flash applets in tests. In contrast, ILIAS goes without such an additional file for describing the external object. In addition, both systems do not provide an interface to enable communication at run-time, for example, to provide hints to guide the student towards the correct solution. Furthermore, the information that is returned from external tool is very limited and not intended to be further-processed (e.g., in order to decide what question should be presented next) by the systems. In this way, the teacher is also not informed about how the student reached this answer or solution. When taking a closer look at the functionality provided by the different e-assessment system and tools, it becomes clear that a significant gap in assessing Bloom's upper levels exists, which needs to be closed by new concepts and approaches.

Bloom's Taxonomy (Cognitive Domain)

Bloom's Taxonomy (Cognitive Domain)	Questionmark Perception	Respondus	TATS	ILIAS	Moodle	OLAT
Evaluation Evaluate or make judgments on the worth of something **Synthesis** Combine parts to create a new whole	Adobe Flash ~~File upload~~ Adobe Captivate ~~Essay~~ ~~Spoken response~~	~~Essay~~	~~Extended text~~	Adobe Flash ~~Essay~~ Java Applet ~~Text-subset~~ ~~File upload~~	Molecular editor (OPAQUE) ~~Essay~~	~~Free text~~
Analysis Break something down and analyse how the parts are related to each other **Application** Apply something learned to solve a problem	~~Fill in blanks~~ ~~Numeric~~ ~~Text match~~	~~Fill in blanks~~ ~~Short answer~~ ~~Long answer~~ ~~Arithmetic~~ ~~Embedded answer~~ ~~Fill in multiple blanks~~	~~Text-entry~~	~~Numeric~~	~~Short-answer~~ ~~Numerical~~ ~~Random sa. matching~~ ~~Regex short answer~~ ~~Calculated~~	
Comprehension Describe something in own works or recognize previously unseen **Knowledge** Recall content in the exact form it was presented	True/false Multiple-choice Multiple-response Yes/no Drag and drop Likert scale Hotspot Survey matrix Matching Pull-down list Knowledge matrix Ranking Select a blank (Explanation)	True/false Multiple-choice Ordering Multiple-select Drop-down list Matching Numerical answer (Description)	Multiple-choice Associate Multiple-response Match Inline-choice Gap-match Slider Hottext Order	Multiple-choice Hotspot Error text Matching Ordering Cloze	True/false Matching Drag and drop Cloze Multiple-choice (Description)	Single-choice Kprim Multiple-choice Cloze

Figure 3.6: Classification of question types according to Bloom

3.3.4 *Standards and Specifications*

This section describes several established e-assessment standards and specifications and compares them to each other. The main focus is on the question types provided and whether and how they enable (didactic) interactivity. In order to enable comparing the different standards and specifications, the Unified Modeling Language (UML)[17] and in particular the class diagram is used. It abstracts from secondary aspects such as syntax and concrete Extensible Markup Language (XML) bindings and supports highlighting the major differences.

3.3.4.1 *Moodle XML*

Assessment Content Structure

Moodle XML[18] is a Moodle specification for importing and exporting questions to be used in Moodle quizzes (cf. Section 3.3.3.5). The structure of the Moodle XML specification is depicted in Figure 3.7. It shows that each question is associated to a question type such as true/false, multiple-choice or cloze (cf. Section 3.3.3.5). In addition, each question has in turn specific attributes and sub-elements such as *feedback*, *defaultgrade* and *penalty*).

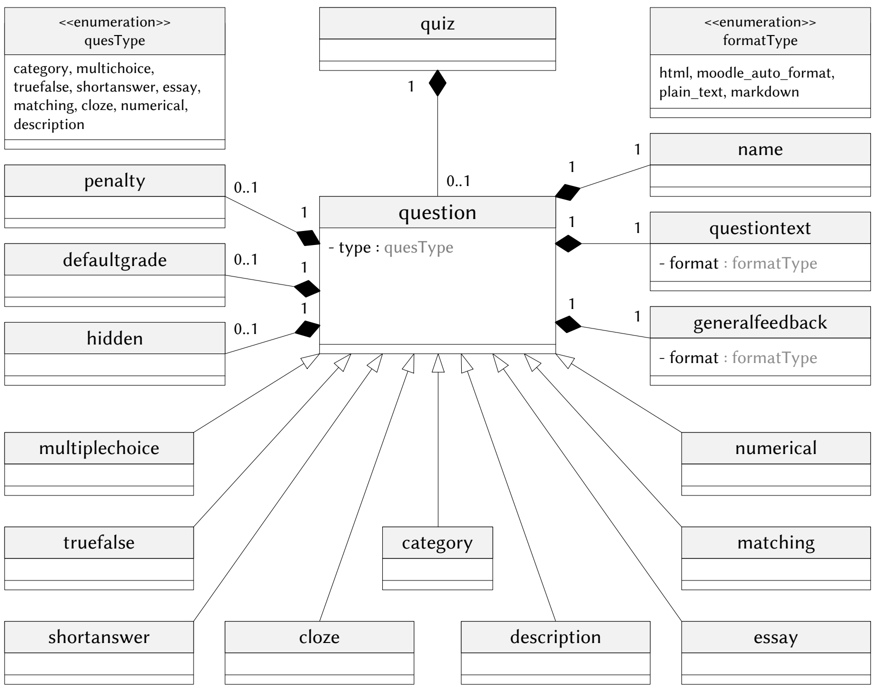


Figure 3.7: Structure of a Moodle XML question

In conclusion, particularly noteworthy on Moodle XML is the fact that it covers the extensive range of feedback Moodle is able to define. However, according to the question types specified, it is much as limited as Moodle

17 `http://www.uml.org/` [last visited: May 21, 2014]
18 `http://docs.moodle.org/25/en/Moodle_XML_format` [last visited: May 21, 2014]

itself. The very promising question types mentioned in the Moodle section (e.g., molecular editor) are not based on Moodle XML, but are implemented individually.

3.3.4.2 *ASD/AIA/ATA S1000D Learning Assessment*

S1000D[19] is an international specification for authoring technical data that supports authoring, storing and publishing of maintenance and operational information. Although S1000D was initially developed for the procurement and production of technical publications, the specification can also be used for non-technical publications. The initial issue of the S1000D specification (AECMA Spec1000D) was firstly released in 1989 for use with military aircrafts and later extended to include land, sea and commercial equipment. The following explanations are based on version 4.0.

In order to support the reuse of technical data, S1000D requires data to be created and maintained in *data modules*. This enables reusing and redistributing the same data module in many other publications and updating a data module will automatically effect updating the dependent publications. S1000D defines 13 content types, also called as *data module types*. The data module types range from descriptive and procedural information to maintenance checklists and inspections and provide the capability to capture operator, maintenance and training content. Particularly notable amongst the various types is the *learning* data module type. It supports the development of technical training content using the data module concepts of S1000D. The learning data module captures five different learning information types namely:

- *Learning overview*: This type is used to drive course strategy and development.
- *Learning plan*: This type is used to introduce the course to students.
- *Learning content*: This type is used to hold the learning content itself.
- *Learning summary*: This type is used to review learning objectives and learning activities and can also discuss future learning requirements.
- *Learning assessment*: This type is used to test students' learning.

Assessment Content Structure

In S1000D, a learning assessment consists of a set of attributes and a sections such as *title*, introduction *lcIntro* and interactions *lcInteraction* (cf. Figure 3.8). The specification provides six different interactions that can be used to create assessment items namely *true/false*, *single-select*, *multiple-select*, *sequencing*, *matching* and *hotspot*. In addition, each interaction is subdivided in attributes and sub-sections that allow specifying details about the interaction. In terms of feedback, S1000D enables providing feedback to the students for a correct and incorrect response in an assessment interaction and during an interaction for each answer option. The latter one is limited to true/false, single-select and multiple-select interactions.

19 http://public.s1000d.org/ [last visited: May 21, 2014]

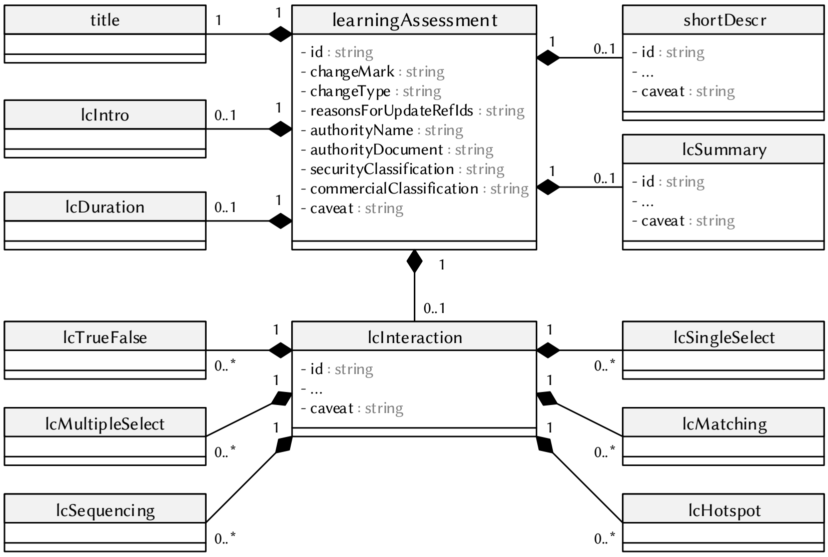


Figure 3.8: Structure of the S1000D learning assessment

In conclusion, ASD/AIA/ATA S1000D Learning Assessment is similar to Moodle XML as measured by the interaction/question types provided. Hence, the specification also does not enable (didactic) interactivity during answering. Furthermore, the extent of providing feedback is also very limited and only represents a subset of the functionality Moodle offers.

3.3.4.3 *IMS QTI*

IMS QTI[20] is an international specification that describes a *data model* for representing assessment content and its corresponding results, and an XML *data binding* that defines a language for interchanging these materials. It allows assessment content to be authored, delivered and exchanged between authoring and delivery systems, repositories and other LMSs. The first version (v1.0) was released in 2000. The following explanations are based on version 2.1.

Assessment Content Structure

The IMS QTI specification structures assessment content into *assessment items*, *assessment sections*, *test parts* and *assessment tests*. An assessment item (*assessmentItem*) is the smallest object that can be exchanged using the IMS QTI specification. Although it can be considered as a question, it contains more than a question text, choices and a correct answer. Figure 3.9 shows the internal structure of an assessment item. In order to characterize an assessment item, IMS QTI defines several *attributes*, some of which are mandatory (e.g., *identifier* and *title*) and others are optional (e.g., *label* and *language*). Many of the functionality of IMS QTI relies on *item variables*. These variables are declared by *variable declarations* and used to hold information during

20 `http://www.imsglobal.org/question/` [last visited: May 21, 2014]

the run-time processing of an assessment item. There are three types of assessment item variables:

- *Response variables*: These variables are used to hold the answers of the students. They are assigned during item body interactions and referenced inside the response processing.
- *Outcome variables*: These variables are used to hold a score gained by a student. They are assigned during response processing and referenced inside modal and integrated feedback.
- *Template variables*: These variables are used to generate different assessment items by cloning. They are assigned during template processing and referenced inside the item body and the response processing

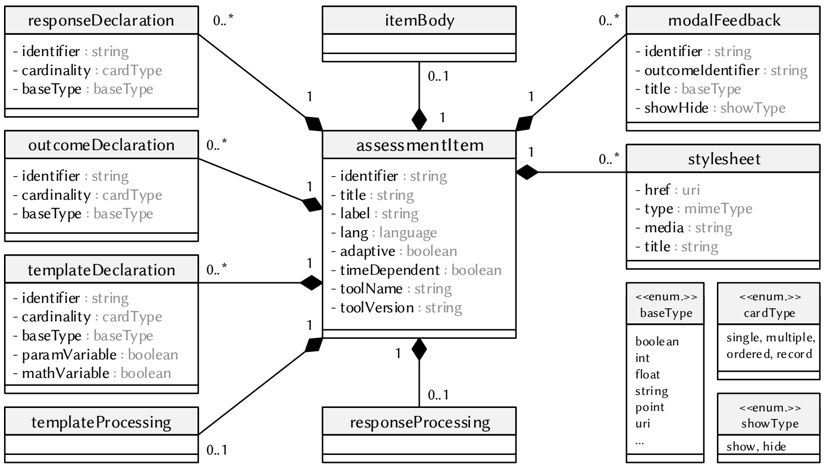


Figure 3.9: Structure of an IMS QTI v2.1 assessment item

Each assessment item variable type is declared within a particular declaration section (*responseDeclarations*, *outcomeDeclaration* and *templateDeclaration*). *Template processing* specifies one or more template rules, which are evaluated by delivery systems in order to assign values to the template variables. The *item body* specifies the assessment item as seen by the student and interactions between the student and the assessment item. *Interactions* allow to interact with the assessment item and are associated with at least one response variable. The student's responses are stored in these response variables and passed to the response processing. IMS QTI splits the term interaction into several sub-interactions from which 21 actual item interactions/types are derived. These assessment item interactions are categorized into four main types namely *simple*, *text-based*, *graphical* and *miscellaneous* (cf. Figure 3.10). *Question Types*

Particularly notable amongst the various interactions is *customInteraction*. It enables extending the QTI specification to include new interaction types,

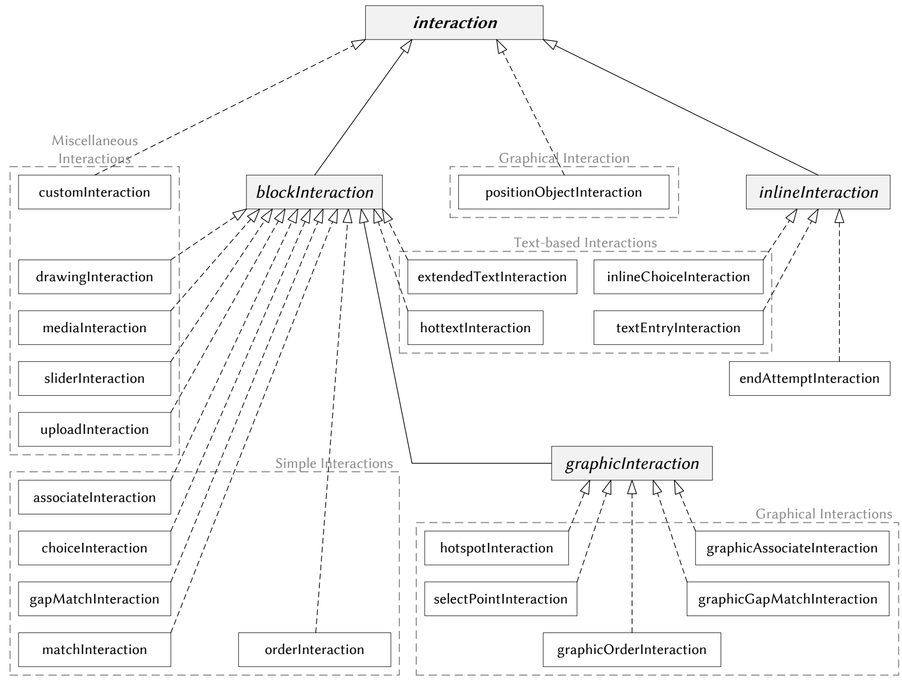


Figure 3.10: Hierarchy of IMS QTI v2.1 interactions

which are not covered by existing assessment item interactions. The *responseProcessing* describes the process of judging response variables and setting outcome variables, which can be used, for example, to provide feedback to the students. Response processing instructions are expressed using different control structures such as *responseIf*, *responseElse* or *setOutcomeValue*.

Feedback The IMS QTI specification enables authors to specify feedback at different points during the interaction with the assessment item. Feedback is presented to the students based on the values of outcome variables. IMS QTI defines three types of feedback:

- *Integrated feedback*: This type of feedback is integrated into the item body. Students are free to update their responses while viewing integrated feedback.
- *Modal feedback*: This type of feedback is not integrated into the item body and defined in its own section.
- *Test feedback*: This type of feedback is presented to the students based on the values of test outcomes.

Hence, modal and integrated feedback is defined by assessment items and test feedback is defined by assessment tests. The visibility of both modal and integrated feedback can dynamically be adapted using *identifier* and *showHide* attributes and outcome variables.

An assessment test (*assessmentTest*) is an organized collection of items. It can contain zero, one or more *outcome declarations*. Each declaration declares one *outcome variable*, which stores a value relevant to the student's performance in the test. In addition, each assessment test is divided into one or more *test parts*, which in turn are divided into *assessment sections* and assessment sub-sections, etc. Particular noteworthy on test parts are the elements *preCondition*, *branchRule*. These are simple expressions, which are evaluated *during* and *after* a test part and allow adapting the *assessment sections* accordingly.

In conclusion, IMS QTI is the most comprehensive specification for representing assessment content and its corresponding results. Although the specification does not provide didactic interactive question types, it offers well-defined extension mechanisms such as the *customInteraction* and the *customOperator* element. Particularly worth mentioning is the former one. It allows including new interaction types, which are not covered by existing assessment item interactions. Additionally, also noteworthy on IMS QTI are the branching and sequencing feature introduced in version 2.1. They allow items to be scored individually over a sequence of attempts. With respect to feedback, the specifications enables a wide range of feedback possibilities by the use of response and outcome processing. Disadvantageous on IMS QTI is that it implies a great difficulty of implementing it (completely). However, there are several open-source QTI engines (e.g., QTIEngine[21] or NewAPIS[22]) that can be used with the own system or tool to ensure compliance with this specification.

3.3.4.4 *Comparison*

In the previous sections, several established e-assessment standards and specifications were described and analyzed. Although they all define data models for representing assessment content and/or its responses, however, they clearly differ in terms of question types and feedback option supported. Moodle XML is a Moodle-specific format and thus mainly covers the assessment functionality provided by Moodle. In contrast, IMS QTI is not limited to any specific LMS/LCMS and provides a more finer subdivision of the assessment content, a variety of interaction types and several other assessment options. ASD/AIA/ATA S1000D learning assessment is also not limited to any LMS/LCMS, but only uses a subset of the functionality provided by the IMS QTI specification. In this respect, it can be compared to Moodle XML, even though it provides much less feedback options. Also worthy of mention on IMS QTI is the fact that it not only provides a model for representing content, but also for representing and processing the responses resulting from the interactions with the questions (i.e., outcome and response processing). These functionalities are prerequisites to respond to students' inputs appropriately, for example, by giving directed feedback or by adapting the subsequent (sequence of) questions. In contrast, Moodle XML and S1000D learning assessment only allow presenting the same questions and the same feedback to each student. In addition, also noteworthy on IMS QTI in contrast

21 http://sourceforge.net/projects/qtitools/files/QTIEngine/ [last visited: May 21, 2014]
22 http://sourceforge.net/projects/newapis/ [last visited: May 21, 2014]

to the other specifications are the well-defined extension mechanisms. The *customInteraction* element, for example, allows implementing and integrating interaction types currently not included by the specification.

3.3.5 *Summary*

This section has investigated how interactivity is provided or supported by established e-assessment systems and tools. But first, the term ICO has been defined. It not only refers to simulations, but also animations and virtual reality fall into this category. Then, six established e-assessment systems and tools were described and analyzed, and finally, classified according to the levels in Bloom's taxonomy they are able to assess. Following on that, several established e-assessment standards and specifications were given and compared to each other.

CONCLUSION: Relevant for this thesis is the fact that there is a significant gap in assessing Higher-Order Thinking Skills (HOTS) by established e-assessment systems and tools. This can be highlighted by the insufficient support for interactive e-assessments, both from the student as well as from the teacher's point of view. This confirms the need for a new architectural model that enables integrating and using ICOs for assessment. Although there is no e-assessment standard or specification that specifies how to integrate and interact with ICOs, however, the IMS QTI specification has been turned out to be well-suited to built on. Not only due to its well-defined extension mechanisms, but also because of its branching and sequencing feature.

3.4 PERSONALIZATION IN EDUCATIONAL HYPERMEDIA

3.4.1 *Introduction*

This section analyzes how personalization is realized in educational hypermedia. It starts looking at adaptive hypermedia, its key features and subtypes 3.4.2. Afterward, Section 3.4.3 describes a number of reference models for implementing as well as comparing such systems.

3.4.2 *Adaptive Hypermedia*

A *hypermedia system* is defined as a computer-based system that provides access to texts, graphics, audio, videos, etc. related to a particular subject. All of these elements are intertwined by hyperlinks to create a generally non-linear medium of information [Nel65]. The currently largest hypermedia system is the World Wide Web (WWW). The growing success of the Internet has resulted in a huge amount of content in nearly all hypermedia systems. But, one limitation of traditional hypermedia systems is that they provide the same content to all users. This *one-size-fits-all* approach has led to navigation problems in the *hyperspace* and resulted in users becoming disoriented. This problem has been addressed by Adaptive Hypermedia Systems (AHSs). AHSs aim at increasing the functionality of hypermedia systems by adapting the content to the individual needs of the user [Bru96, Bru01]. In general, Brusilovsky1996 [Bru96] identified three key features of an AHS namely:

1. It is a hypertext or hypermedia system
2. It maintains a user model that reflects some features about the user
3. It is able to adapt the hypermedia using this model

Due to the fact that AHSs can be useful anywhere where hypertext and hypermedia are used, several fields of application for AHSs emerged over the time. In 2001, Brusilovsky [Bru01] identified three main kinds of AHSs namely:

- *Adaptive Educational Hypermedia Systems (AEHSs)*: This kind of AHS enables adapting learning content to the students' individual context, prior knowledge in the subject and preferences. Examples of such systems are AHA! [DC98], ELM-ART [WB01] and INSPIRE [GPKM01].
- *Adaptive online information systems*: This kind of AHS enables adapting the access to online information to the users' individual context, prior knowledge in the subject and preferences. Examples of such systems are SWAN [GIK99], ADAPTS [BC99] and HIPS [OS99].
- *Information Retrieval (IR) hypermedia systems*: This kind of AHS enables browsing the hyperspace of documents using similarity links between documents. Examples of such systems are WebWatcher [JFM97], PEA [MGH97] and ELFI [SPK00].

In order to classify all the methods and techniques used in the different AHSs to provide adaptation, Brusilovsky has defined a taxonomy of adaptive hypermedia technologies [Bru96]. This taxonomy was updated in 2001 [Bru01], in order to accommodate new methods and techniques (cf. Figure 3.11). In general, the taxonomy is subdivided into:

- *Adaptive presentation*: This group of adaptive hypermedia technologies adapts the content of a hypermedia page to the current knowledge, goals and other characteristics of the individual user.
- *Adaptive navigation support*: This group of adaptive hypermedia technologies helps users in navigating through the hyperspace by adapting the way of presenting links to the current knowledge, goals and other characteristics of the individual user.

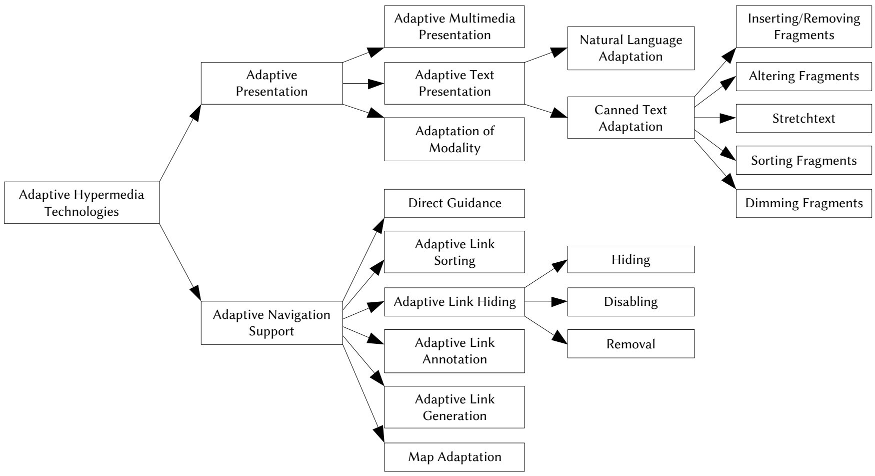


Figure 3.11: Taxonomy of adaptive hypermedia technologies (adopted from [Bru01])

3.4.3 *Reference Models*

Over the last few years, several reference models in the field of adaptive hypermedia were developed. They not only serve as a starting point for the design of new AHSs, but also enable comparing the characteristics and functionality of different AHSs. Most of them are based on the Dexter hypertext reference model [HS94]. The Dexter reference model was the first attempt to formally specify the important abstractions found in existing hypertext systems. It divides a hypermedia system into three layers (cf. Figure 3.12a):

- *Run-time layer*: This layer deals with the presentation of hypertext and user interactions.
- *Storage layer*: This layer describes how the hypertext components and links are connected to each other.

- *Within-component layer*: This layer is concerned with the content and structure within the components of the hypertext network.

The interface between the run-time and the storage layer is accomplished by a mechanism called *presentation specifications*. It specifies how a component is to be presented to the user and how this kind of information is encoded into the storage layer. The interface between the storage and the within-component layer is accomplished by a mechanism called *anchoring*. It refers to locations or items within the content of an individual component. Over time, several reference models have been developed, which extends the original Dexter model to provide adaptation based on a User Model (UM). Examples of such models are AHAM, MRM and LAOS.

AHAM

The Adaptive Hypermedia Application Model (AHAM) [DHW99] preserves the three-layer structure of the Dexter model, but subdivides the *storage layer* to include user modeling and adaptation aspects (cf. Figure 3.12b):

- *Domain Model* (DM): This model represents the author's view on the application domain expressed in *concepts* and *concept relationships*.
- *User Model* (UM): This model represents the relationship between the user and the Domain Model (DM).
- *Teaching Model* (TM): This model consists of *pedagogical rules*, which specify how the UM is updated and the adaptation is done by using information from the UM and the DM.

MRM

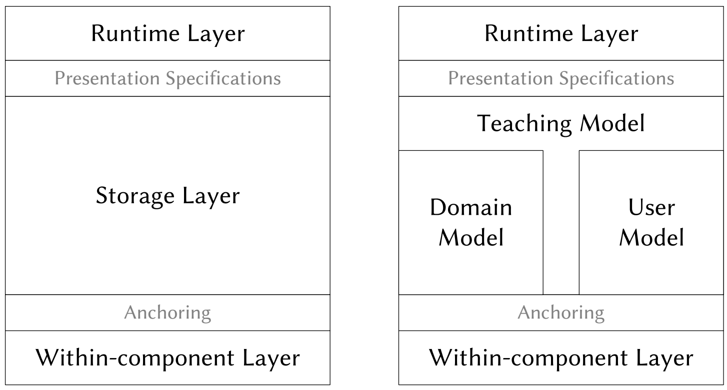
LAOS

AHAM not only focuses on the *storage layer* of the original Dexter model, but also on the *anchoring* and the *presentation specification*. An Adaptation Engine (AE) uses the rules provided by the teaching model to manipulate link anchors (*anchoring*) and to generate *presentation specification*. The Munich Reference Model (MRM) [KW02a] is similar to AHAM in dividing the *storage layer* into three models to support adaptation, but it is formally defined from an object-oriented software engineering point of view. MRM is visually represented using UML notation and formally specified in Object Constraint Language (OCL). UML enables showing how the relevant concepts are organized and how they are related to each other, whereas OCL enables specifying invariants for the model elements and pre- and post-conditions on operations describing the adaptive functionality. The LAOS model [CdM03] is a layered framework for describing AHSs. It is built upon AHAM and additionally allows the inclusion of goals and constraints. The LAOS model consists of five layers:

- *Domain Model* (DM): This layer defines the domains of the content, the composing elements and the relations between these elements.
- *Goal & Constraints Model* (GM): This layer filters useful domain concepts and groups them together according to a predefined pedagogical goal.

- *User Model* (UM): This layer stores user specific variables (e.g, demographics, prior knowledge and preferences).
- *Adaptation Model* (AM): This layer defines a set of rules, which specify how the content is adapted to the user needs.
- *Presentation Model* (PM): This layer deals with the presentation of the content.

Furthermore, Cristea et al. [CSD05] propose an adaptation format that enables representing the Goal & Constraints Model (GM) and all the DMs using a simple XML format and an adaptation language [CC03] that enables specifying more complex adaptation strategies.



(a) Layers of the Dexter reference model (adopted from [HS94])

(b) Layers of the AHAM reference model (adopted from [DHW99])

Figure 3.12: Comparison between Dexter and AHAM reference model

3.4.4 *Summary*

This section has shown that the most common way to realize personalization in educational hypermedia are AHSs. Although a variety of such systems have been developed over the time, they all have two features in common: they (1) maintain a user model that reflects some features about the user and (2) adapt various visible aspects of the system to the user. There are a variety of adaptive hypermedia technologies and they can roughly be divided into two groups: *adaptive presentation* and *adaptive navigation support*. Furthermore, several reference models in the field of adaptive hypermedia have been developed over the time.

CONCLUSION: Relevant for achieving the goals of this thesis are the general structure of AHSs consisting of a UM, a DM and an AM. Although the focus of this thesis is on personalizing questions and tests, the different models and how they work together serve as a good starting point for the design of an architectural model that integrates personalization.

3.5 PERSONALIZATION IN E-ASSESSMENT SETTINGS

3.5.1 *Introduction*

This section analyzes how personalization is realized in e-assessment settings. First of all, the term *adaptive assessment* is defined and its techniques are discussed (Section 3.5.2). Afterward, examples of systems and tools that make use of these techniques are given and compared to each other (Section 3.5.3).

3.5.2 *Adaptive Assessment*

In general, *adaptive assessment* is defined as a form of assessment, where the structure and behavior of the test is adapted to students' individual characteristics or to responses to previously answered questions [Sit09]. A system or tool that provides adaptive assessments can be referred to as Adaptive Assessment System (AAS). There are two types of adaptive techniques that are applied in AASs namely *adaptive testing* and *adaptive questions*.

3.5.2.1 *Adaptive Testing*

The technique of adaptive testing iteratively selects a question from a question pool that matches best to the current estimation of the student's knowledge level. According to the answer, the estimation of the student's knowledge is updated. This process is repeated until a certain termination criterion is met [WK84]. The technique of adaptive testing has made an enormous step forward through the development of the Item Response Theory (IRT) in the middle of the last century. The IRT [HSR91, dA08] is based on the idea that the probability of a correct response to a question is a mathematical function of student and item (question) parameters. The student parameter is called as *latent trait* and represents the knowledge level of the student. The item parameters include *discrimination*, *difficulty* and *guessing factor*. The mathematical function is called as *item characteristic curve* or *item response function* and represents the conditional probabilities of the successful answer to the question by a student with a certain latent trait. The combination of IRT, adaptive testing and the administration of tests by computers is known as Computerized Adaptive Testing (CAT) [WK84].

Several systems and tools exploit the technique of adaptive testing and IRT such as SIETTE (cf. Section 3.5.3.1) and PASS (cf. Section 3.5.3.2).

3.5.2.2 *Adaptive Questions*

The technique of adaptive questions specifies a dynamic sequence of questions based on rules. These rules are linked, for example, to the response of the students and allow selecting appropriate or skipping unrelated questions at run-time [GPG02, TRP04]. The basic principles of adaptive questions go back to computer-assisted surveys. Pitkow and Recker [PR95] showed that web-based adaptive questionnaires can reduce the number and complexity of questions presented to each student. In addition, in CATES [Choo0], adaptive questionnaires are used to assess students' attitudes.

Several systems and tools exploit the technique of adaptive questions such as CosyQTI (cf. Section 3.5.3.3) and iAdaptTest (cf. Section 3.5.3.4).

3.5.2.3 *Comparison*

Adaptive testing is an iterative procedure that dynamically selects questions to ask next based on the current estimation of student's knowledge level. The overall goal of this technique is to obtain accurate student knowledge estimations and to minimize the number of questions required for that purpose. That means, testing is continued only as long as necessary for each student. An advantage of adaptive testing is that questions, which are too difficult or too easy are removed. Thus, the technique ensures that students only see questions, which are very close to his or her level of knowledge. Summarized it can be stated that this technique is entirely focused on students' knowledge levels and does not consider any other student characteristic.

In contrast, the adaptive questions technique is not as limited as the adaptive testing technique concerning the information that can be used as basis for adapting the test and its questions. It dynamically adapts the test based on predefined rules (e.g., IF-THEN). The advantage of this technique is that it allows considering a variety of information such as students' interests, context or goals and not only their knowledge. In this way, it gives teachers the flexibility to express their didactic philosophy and methods through the creation of appropriate rules. But students also take advantage of this technique because it allows considering strengths and preferences to compensate individual weaknesses and deficits. According to Lazarinis et al. [LGP10], the adaptive questions technique is especially useful in formative assessments (cf. Section 2.2.4.1).

3.5.3 *Systems and Tools*

This section briefly describes four established AASs and compares them to each other. The first two systems make use of the adaptive testing technique, whereas the last two system make use of the adaptive question technique.

3.5.3.1 *SIETTE*

SIETTE [GC04] is a web-based adaptive assessment system for CAT generation and elicitation. It enables teachers developing adaptive tests by defining subjects, their topics and items, specifying tests and making them available to students. The tests are dynamically selected and presented according to the student's knowledge level estimation.

SIETTE has a *curriculum*-based structure, which means that each test for a particular *subject* is structured into *topics* and questions. A question is associated to one or more topics and belongs to a question type. To this end, the system provides four so-called *internal* question types (i.e., true/false,

Question Types

multiple-choice, multiple-response and open answer) as well as *siettlets*, a *generative* and an *external* question type. Siettlets allow using an embedded program (e.g., a simulation or animation) that captures student's actions, determines whether the answer is correct or not and sends this information back to SIETTE (cf. Figure 3.13). The external question type enables including questions stored on external systems. This type is similar to siettlets because student's responses are corrected externally and sent back to SIETTE. Finally, the generative question type enables generating similar items in real-time.

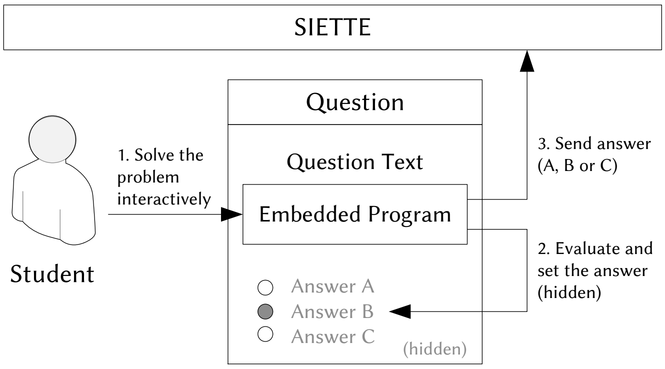


Figure 3.13: SIETTE siettlets (adapted from [GCGH05])

The system allows defining textual feedback for each answer option that will be shown, when the student has selected the respective answer. In order to represent the assessment content and results, SIETTE provides an own specification called *S-QTI*. It consists of a data model that defines the structure of questions, assessments and results together with an XML data binding. S-QTI considers the IMS QTI specification, but extends it with additional elements.

Feedback

Interoperability

In conclusion, noteworthy on SIETTE is the fact that it allows some kind of creativity in answering by providing *siettlets* and *external* question types. However, these advanced question types only imitate multiple-choice items and do not really enable creating interactive e-assessments. But due to the fact that is makes use of the adaptive testing technique, students' knowledge is the only information that is used to provide adaptations. Adaptations are also limited to the selection of questions.

3.5.3.2 *PASS*

PASS (Personalized ASSessment) [GPG02] is an adaptive assessment module within the AEHS INSPIRE [GPKM01]. It aims at estimating student's knowledge level and enables teachers having a detailed overview of the student's performance and progress.

Although the PASS module adopts the adaptive testing technique to estimate student's knowledge level on specific concepts, it additionally uses the adaptive question technique for assessing the students' knowledge on prerequisite concepts. This information is used as entry level or starting point

Question Types

in the adaptive testing procedure. A question in PASS is associated to one or more *concepts*, which are related to other concepts by means of qualitative characterizations [GPKM01]. Furthermore, each question is categorized according to levels of performance:

- *Remember*: A questions on this level requires students to recall content.
- *Use*: A question on this level requires students to apply the provided content in specific problems.
- *Find*: A question on this level requires students to propose and solve original problems.

Feedback

Furthermore and just as SIETTE, PASS allows defining textual feedback for each answer option of the question. This feedback is presented, when the student has selected the respective answer.

In conclusion, particularly noteworthy on PASS is the categorization of questions according to the levels of performance needed to solve the question. However, it does not correspond to any established classification such as Bloom's taxonomy. With respect to the question types provided, PASS is limited to multiple-choice questions. Owed to the fact that it uses the adaptive testing technique, adaptations are limited to the selection of questions and do not affect any other element of the test (e.g., the presentation of questions or the selection of feedback).

3.5.3.3 *CosyQTI*

CosyQTI [LR06] is a web-based adaptive assessment system for authoring and presenting adaptive questions. The architecture of CosyQTI corresponds to whose of AHSs (cf. Section 3.4.3) and consists of the following components:

- *Learner model*: This component contains demographic information, learning goals, learning preferences, knowledge estimations and usage data of each student.
- *Domain model*: This component specifies concepts and their relationships with other concepts.
- *Adaptation model*: This component contains the adaptation decisions in the form of *IF <condition> THEN <action>* rules.
- *Run-time model*: This component applies the rules, collects the usage data and updates the learner profiles.

Question Types

Feedback

In CosyQTI, a question is grouped into sections. Each section is associated with a concept, which in turn is associated with a domain. For creating question, CosyQTI provides five question types namely *true/false*, *multiple-choice*, *fill in the blanks*, *multiple image choice* and *image hotspot*. In terms of feedback, CosyQTI does not allow defining feedback for the students after they submitted an answer. Instead, it allows defining textual hints, which are displayed on request during the test. If the student makes use of hints, a penalty can be set, which will be subtracted from the maximum score. CosyQTI conforms to the IMS Learner Information Package (LIP) and IEEE Public and Private Information (PAPI) specification to exchange student information and to the IMS QTI specification to represent questions and tests.

Interoperability

In conclusion, particularly worth to mention on CosyQTI is the fact that it provides a comprehensive UM (learner model), whose information (e.g., demographic information, learning goals, learning preferences, etc.) can be used to adapt the questions and tests. However, adaptations are limited to move to a certain question and to change the difficulty of the subsequent questions. The presentation or selection of feedback can not be adapted by the system. Also noteworthy is the fact that it allows students to access parts of their learner profile in order to raise their awareness in terms of current knowledge, strengths and weaknesses. With respect to didactic interactivity, CosyQTI is limited to traditional question types.

3.5.3.4 *iAdaptTest*

iAdaptTest [LGP09] is an adaptive assessment system that adapts the testing procedure according to the performance, prior knowledge, goals and preferences of the students. The architecture of iAdaptTest is very similar to those of CosyQTI. They are about the same in that both consist of a set of components that maintain a UM, a DM and an AM. The components of iAdaptTest are as follows:

- *Topics management module*: This component allows teachers developing and displaying topic hierarchies.
- *Learner profile management module*: This component supports the initialization of learner profiles, the import from other sources and the update and display of learner profiles in alternative modes.
- *Testing items and assessment management module*: This component allows teachers creating, importing and modifying test items.
- *Adaptation management module*: This component allows teachers creating and modifying adaptation rules and applying them to specific points in a test.
- *Test execution module*: This component runs the adaptive tests and updates the learner profiles.
- *Data presentation module*: This component views statistics and some conclusions of the testing procedure.

Question Types

In iAdaptTest, a question is associated to one specific topic and belongs to one question type. The system provides six question types namely *true/false*, *single-choice*, *multiple-choice*, *order*, *gap-match* and *associations*. Regarding

Feedback

the provision of feedback, iAdaptTest allows defining textual feedback for both the correct and the incorrect answer. In order to ensure the reusability and interoperability of the content created, iAdaptTest conforms to several established standards and specifications such as IMS QTI, IMS LIP, IEEE PAPI and XML Topic Maps.

Interoperability

In conclusion, worth mentioning on iAdaptTest is the comprehensive UM structured according to the IMS LIP and the IEEE PAPI specification as well as the adaptations supported (e.g., move to or retry a certain question or only show questions of a specific difficulty level). However, feedback is also not considered. With respect to didactic interactivity and just like CosyQTI, iAdaptTest is limited to simple question types and do not provide any (didactic) interactive question types.

3.5.3.5 *Comparison*

In the previous sections, several AASs were described and analyzed according to the (personalized) assessment functionality (i.e., question types, feedback and interoperability) offered. Similar to the comparison of e-assessment systems and tools (cf. Section 3.3.3.7), Figure 3.14 shows the different question types provided classified according to Bloom's taxonomy. It is not surprising that the result is very similar to those of the non-adaptive systems and tools. Only SIETTE provides a question type that allows a little more creativity in answering, but as already stated, the evaluation of the answer is done by the integrated program itself and only the information whether the answer is correct or not will be returned to the system. In this way, the preceding elaboration processes that led to the answer cannot be recognized and thus, reliable statements about the competencies achieved cannot be made.

In terms of the adaptations provided, the different AASs differ significantly from each other (cf. Table 3.2). One reason for this is the use of different adaptive assessment techniques. Due to the fact that SIETTE and PASS make use of the adaptive testing technique, the *adaptation process* is realized by the IRT. This requires the use of the multiple-choice question type and its derivatives. In contrast, CosyQTI and iAdaptTest are based on the adaptive question technique and are less restricted in providing advanced question types. But as described above, they do not exhaust their potential. Similar restrictions exist when looking at the *adaptation information*. The IRT-based systems entirely focus on students' knowledge, in contrast, the rule-based systems allow using much more features. This opens up entirely new opportunities for adaptations such as considering individual preferences, goals or the scores achieved. The technique used is also reflected to the *adaptation means* (i.e., the elements of the test that can be adapted). SIETTE and PASS select questions that match best to the current estimation of the student's knowledge level. In this way, the difficulty of the test will also be adapted. However, the author has no influence on this process. In contrast, CosyQTI and iAdaptTest allow for much more creativity in specifying the adaptive behavior of the test and its questions. For example, CosyQTI allows to move to certain questions or to change the difficulty by removing subsequent

		SIETTE	PASS	COSYQTI	IADAPTTEST
Adaptation Process		IRT	IRT, Rules	Rules	Rules
Adaptation Information	Knowledge	x	x	x	x
	Goals	–	–	–	x
	Preferences	–	–	–	x
	Scores	–	–	x	x
Adaptation Means	Question sequence	–	–	–	x
	Question selection	(x)	(x)	x	x
	Question presentation	–	–	–	–
	Test difficulty	(x)	(x)	x	x
	Feedback selection	–	–	–	–
	Feedback presentation	–	–	–	–

Table 3.2: Comparison of adaptive assessment systems and tools

questions that are too easy or too hard. Moreover, iAdaptTest also allows retrying a question or even the whole test. Although all systems provide some kind of adaptive question selection, however, the adaptive *presentation of questions* is not taken into account at all. But within e-learning settings, students' learning styles (e.g., visual or auditory) already play an important role [Gra07]. The same applies to feedback. The different AASs do not allow adapting the *selection* or *presentation of feedback* to the individual student. Instead, they are limited to provide textual feedback for the different answer options. In this way, the potential that feedback has (e.g., communicating strengths and weaknesses, hints or advices for continuing the assessment) remains unused.

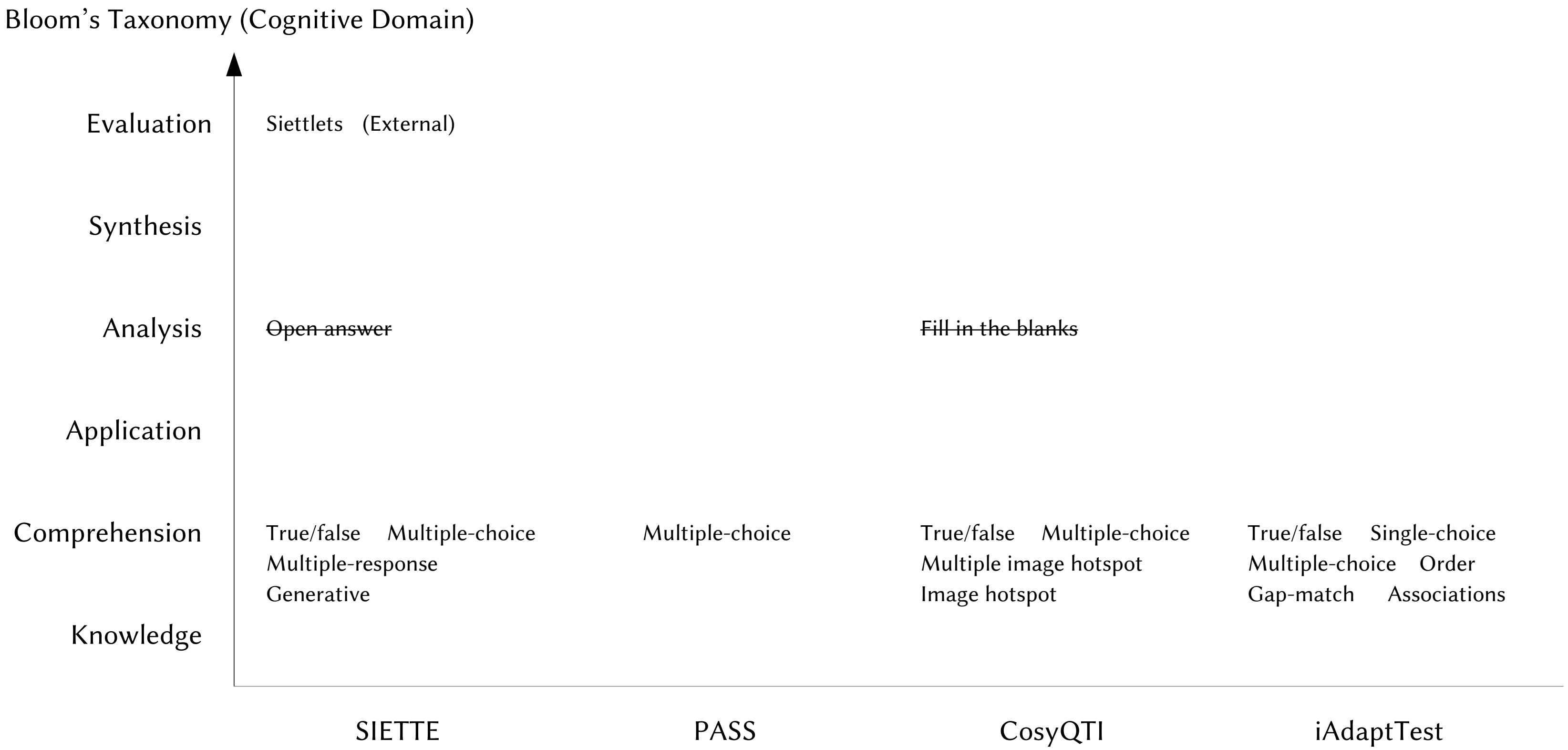


Figure 3.14: Classification of question types according to Bloom (2)

3.5.4 *Summary*

This section has shown that the most common way to realize personalization in e-assessment settings are AASs. They adapt the structure and behavior of the test to students' individual characteristics or responses. There are two types of techniques that can be used to develop AASs namely *adaptive testing* and *adaptive questions*. The adaptive testing technique iteratively selects a question from a pool of questions that matches best to the current estimation of the student's knowledge. The overall goal of this technique is to obtain accurate student knowledge estimations. In contrast, the adaptive questions technique specifies a dynamic sequence of questions based on rules (e.g., IF-THEN). These rules allow selecting appropriate or skipping unrelated questions at run-time. This technique also allows considering a variety of information such as students' interests, context or goals and not only their knowledge. In this way, teachers are able to express their didactic philosophy and methods through the creation of appropriate rules.

CONCLUSION: Relevant for achieving the goals of this thesis is the fact that the technique of adaptive questions provides the necessary flexibility for implementing the adaptive methods envisaged. This does not only concern the information that can be used to provide adequate adaptations, but also the elements that are adapted. Furthermore, this technique is open to advanced questions (using ICOs) and not limited to the multiple-choice question type. However, it has been shown that this potential has not yet been exploited so far. This also concerns feedback and feedback personalization.

3.6 SUMMARY

This chapter focused on reviewing the state-of-the-art for answering the research question and achieving the research goals. Based on a general overview about the use of interactivity in e-learning settings, Section 3.3 investigated how interactivity is realized or supported by established e-assessment systems and tools as well as standards and specifications. This was done in order to get an impression how far the vision of *Assessment 2.0* has progressed. As a result, it is still a vision and far from being reality. Although the development of e-assessment systems and tools has progressed rapidly, creating interactive e-assessments and thus enabling didactic interactivity is mostly neglected or even not addressed by these systems.

Afterward, the realization of personalization aspects moved into focus. Based on a general insight into personalization in educational hypermedia, Section 3.5 reviewed established personalization techniques and systems in e-assessment settings. This was done because the aim of this Ph.D. project is to bring together interactive e-assessments and personalization. As a result, the consideration of personalization aspects in e-assessment settings is realized by AASs. Although they all provide some kind of adaptive question selection, however, the adaptive presentation of questions is not taken into account at all. Moreover, the adaptive selection or presentation of feedback to the individual student is also entirely disregarded by the systems considered. With respect to didactic interactivity, AASs are mainly limited to traditional question types (mostly multiple-choice questions) and do not allow creating interactive e-assessments.

Finally, Figure 3.15 depicts and brings the (personalized) e-assessment systems and tools, which are discussed in this chapter, into relation to the different assessment terms mentioned in the introduction (cf. Section 1.1). It can be seen that *Assessment 3.0* or even *Assessment 2.0* is not really addressed by these state-of-the-art systems and tools.

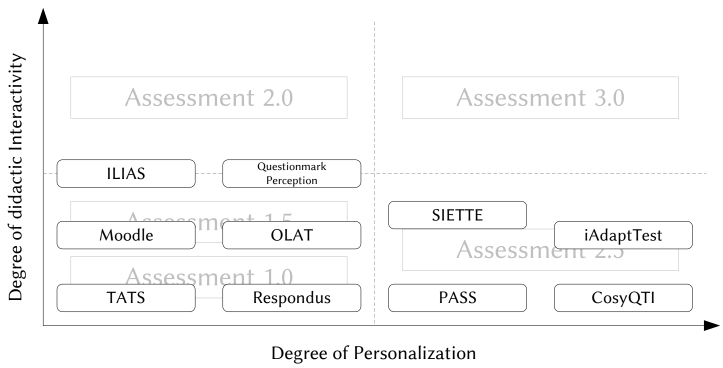


Figure 3.15: Classification of state-of-the-art (personalized) e-assessment systems and tools

4 DESIGN

4.1 INTRODUCTION

This chapter presents the novel architectural model for e-assessment systems and tools developed in this thesis. First of all, Section 4.2 presents an overview of the approach and deals as an anchor for the following sections that explain its concepts in more detail. Then, Section 4.3 presents the design of the Question Model (QM) that enables describing and initializing Interactive Content Objects (ICOs) as well as processing students' responses resulting from interacting with them. Section 4.4 presents the design of the Domain Model (DM) that represents the body of knowledge that makes up a (knowledge) domain and serves as the basis for relating questions and students' features. Section 4.5 presents the design of the User Model (UM) that represents the essential information about each individual user that is needed to provide the adaptation aimed at. Section 4.6 presents the design of the Adaptation Model (AM) that defines what can be adapted, as well as when and how it is to be adapted. The four chapters describe the different models, whereas each of them provides a brief look into the state-of-the-art related to the respective field. The different related work sections differ from the actual state-of-the-art chapter (cf. Chapter 3) in so far as they are only relevant for the design of the respective model. Following this, Section 4.7 gives an insight into the design of the different architectural components based on a set of functional and non-functional requirements. After that, Section 4.8 presents the design of a seamless integration of Learning Environments (LEs) and Adaptive Assessment Systems (AASs) and finally, Section 4.9 summarizes the main results.

4.2 PROPOSED APPROACH

4.2.1 *Introduction*

The primary objective of this Ph.D. project is to propose and evaluate an architectural model for realizing interactive and personalized e-assessments. This section introduces the overall structure of the model developed in this thesis. It serves as an anchor for the following sections that explain the constituent components/sub-models in more detail. First, Section 4.2.2 presents the requirements regarding the design of the overall model. Then, Section 4.2.3 roughly describes the architectural components and depicts a component diagram. Finally, an insight into the architectural/component design principle and pattern is given.

4.2.2 *Requirements*

Considering the limitations of established e-assessment systems and tools, the proposed model has been designed to deliver interactive and personalized questions that:

- Consider students' individual aspects (e.g., knowledge, interests and background)
- Enables didactic interactions by integrating ICOs
- Enable teachers expressing their didactic philosophy and methods
- Are flexible and extensible in the types of adaptivity provided

Towards meeting these benefits, the following technical (non-functional) requirements were considered in the design of the architectural model:

- *Separation*: Architectural components should have exclusivity and singularity of purpose
- *Loose coupling*: Separate the components from each other to reduce the complexity and improve extensibility
- *Standardization*: Utilize established standards and specifications to aid reusability of the content created
- *Interoperability*: Provide well-defined interfaces to achieve interoperability between different systems and tools

4.2.3 *Architectural Model*

User Modeling Component

In general, the architectural model is based on three pillars: a consistent *User Model* (UM), a generic *Domain Model* (DM) and a flexible *Adaptation Model* (AM). Each of them is managed by an own component. The *user modeling component* is responsible for maintaining the UM. A UM is a *"representation of information about an individual user that is essential for an adaptive system to provide the adaptation effect"* [BM07]. Simply put, it is a representation of the properties of an individual user. In contrast, a *user profile* is an instantiation of a UM representing a specific (real) user. The UM is the basis for all adaptive changes to the system's behavior. In order to build up and modify UMs, the component enables collecting and storing data from various sources. This includes both implicit and explicit information. Explicit information (e.g., demographic data) can be gathered by requested direct inputs from the users, whereas implicit information (e.g., knowledge) are gathered while the user interacts with the system. Besides, the user modeling component provides well-defined interfaces to get and set user information.

Domain Modeling Component

The *domain modeling component* is responsible for maintaining the DM. A DM represents the body of knowledge that makes up a domain and decomposes them into a set of domain elements. Although these elements are named differently in different systems (e.g., concepts or knowledge elements), they always denote elementary fragments of domain knowledge. In addition, a DM can also specify the relationships between the different domain elements and can specify their attributes. These links are not only used to reason about the properties and concepts of the domain [OPC+09], but also allow improving the precision of user modeling by, for example, identifying the most likely elements that will remedy the situation, if a user demonstrates a lack of knowledge [BM07]. Whereas in the past DMs were often shown in the form of graphs, nowadays, DMs often use ontologies or topic maps for constituting domain knowledge [ND08]. The domain modeling component provides well-defined interfaces to get domain elements and information about their relationships to each other.

Adaptation Modeling Component

The *adaptation modeling component* is responsible for maintaining the AM. An AM defines *what* can be adapted, as well as *when* and *how* it is to be adapted [Par09]. Adaptation is realized by creating *adaptive tests*. An adaptive tests is a set of questions and *adaptation rules*, which define which and how the questions are to be presented. Due to the fact that the adaptation rules are often refer to or combine information from the user and domain modeling component, the adaptation modeling component enables retrieving and integrating user model as well as domain information from the user and domain modeling component, respectively.

Question Modeling Component

Altogether, these three components build up the central part and the basis of the architectural model. However, there are three further components, which complement the overall model. The *question modeling component* is responsible for authoring questions to be used in adaptive tests. All the questions created by this component are based on a uniform *Question Model* (QM) that determines the structure of questions and how to process students' responses. The component not only allows creating traditional question types (e.g., multiple-choice or true/false), but also enables didactic interactions through the generic integration of ICOs into e-assessment processes.

Adaptive Testing Engine Component

The *adaptive testing engine component* is responsible for performing the actual adaptation by reading the adaptive tests, interpreting the included adaptation rules, presenting the questions to the students and evaluating their responses. It also updates the user profile with the test results, presents statistics about the tests taken and makes assumptions about students' knowledge level in specific domains as well as their strengths and weaknesses.

Settings Component

The *settings component* is responsible for administering general settings that concern more than one component.

A UML component diagram of the overall model is depicted in Figure 4.1. It shows the interaction between the different components and the actors involved. The figure shows three actors namely *student*, *author* and *administrator*. The student takes the adaptive tests and can have a look at its statistics (e.g., correct and incorrect answers or the current level of knowledge). The administrator adjusts all settings, which are of general interests by the other components (e.g., user accounts) and the author manages all the other issues (e.g., creating user and domain models, questions and adaptive tests). However, it is possible and even advisable to subdivide this actor into *domain author*, *question author*, etc.

4.2.4 *Design Principle and Pattern*

The previous section has presented the overall architectural model consisting of a set of loosely coupled components, where each of them has a dedicated task/role. In this way, the architectural model conforms to the design principle of modern software engineering namely Separation of Concerns (SoC). The overall goal of SoC is to *"establish a well-organized system where each part fulfills a meaningful and intuitive role while maximizing its ability to adapt to change"* [Gre08]. SoC is a very important design principle and found its way into a variety of programming languages and design patterns due to its benefits:

- Increased maintainability of the overall system
- Better extensibility
- Adaptability and customization through component exchange
- Reduced complexity

In order to best support the idea of SoC, the Model-View-Controller (MVC) software design pattern [GHJV94] has been chosen for designing and implementing the different components. MVC promotes SoC by the separating the Graphical User Interface (GUI) from the application data and logic. Consequently, the architectural model is characterized by a combined *vertical* and *horizontal separation strategy*. The vertical separation is realized by dividing the system into a set of components that relate to the same feature within the system (i.e., adaptation modeling component, user modeling component, etc.). This clarifies the responsibilities and dependencies of each feature and aids in testing and maintaining the overall system. In contrast, the horizontal separation is realized by dividing the system into layers of functionality that fulfill the same role. The layering is a consequence of the use of the MVC design pattern. That means that each component is horizontal subdivided into a *presentation*, *control* and *data layer*.

The presentation layer consists of a set of *views* and embodies the interface for the interaction with the different components. The data layer consists of a set of *models*, which abstract the data and provide access for reading and manipulating the data and the control layer consists of a set of *controllers*, mediates between the data layer and the presentation layer and makes sure that the right models are called and the right view is rendered. The overall structure of the separation is depicted in Figure 4.2.

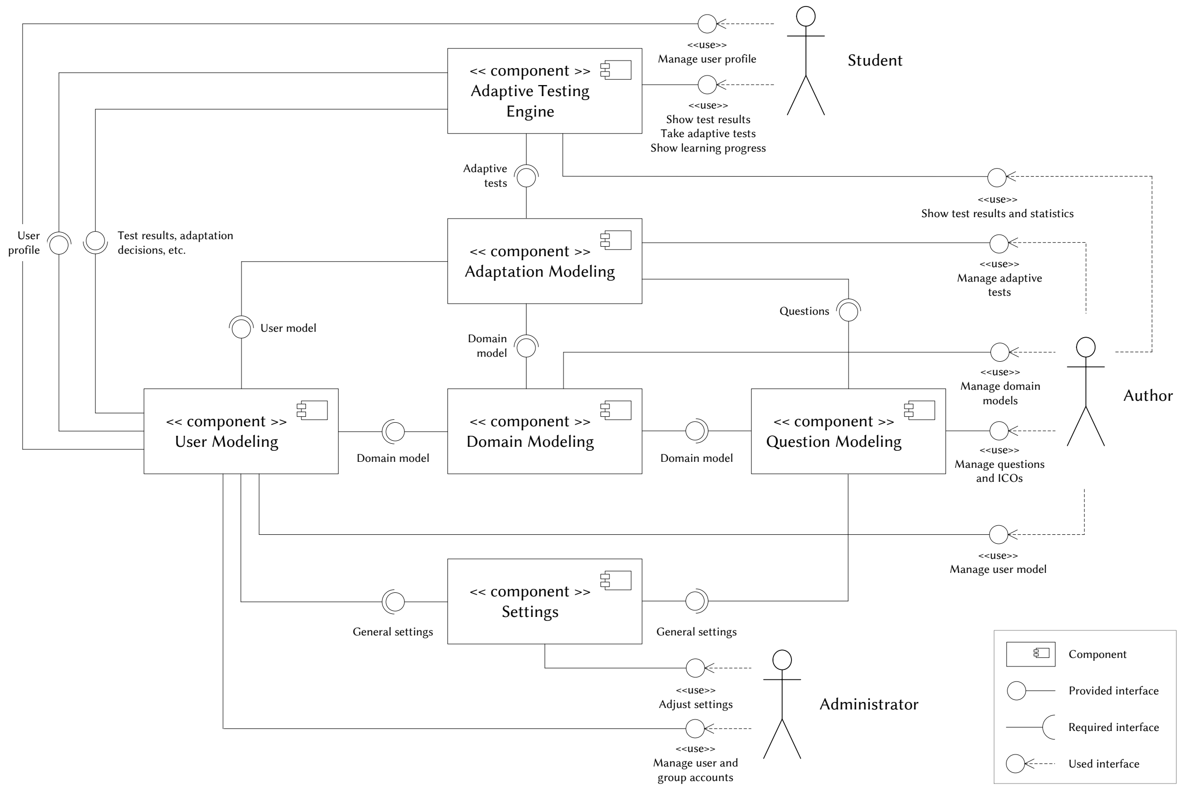


Figure 4.1: Component diagram of the architectural model

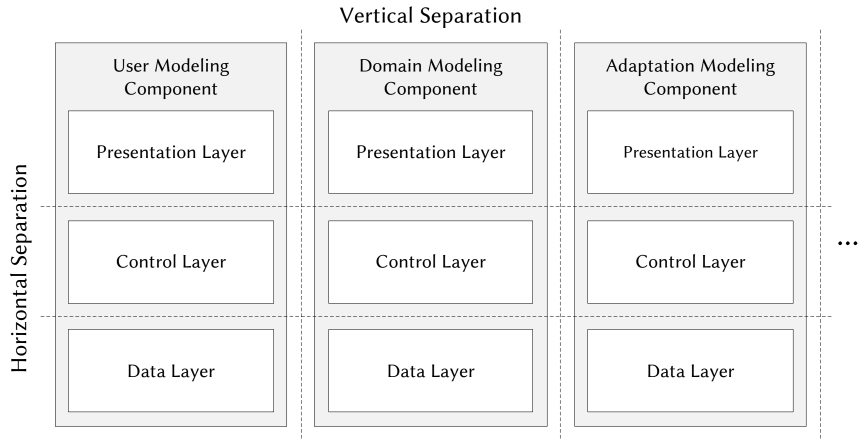


Figure 4.2: Separation structure of the architectural model

4.2.5 *Summary*

This section has presented the overall structure of the architectural model developed in this thesis. It consists of a *user*, *domain* and *adaptation modeling* component, which build up the central part and the basis of the overall model. Additionally, a *question modeling* component is also a fundamental component because it is responsible for authoring questions to be used in adaptive tests. All of these components maintain a specific model, which determine the behavior and functionality of these components. The next four sections are dedicated to describe these models in more detail. In defining a user, domain and adaptation component, the architectural model roughly corresponds to the structure of the most Adaptive Hypermedia Systems (AHSs) (cf. Section 3.4.3). The model is completed by an *adaptive testing engine* component that performs the actual adaptation and a *settings* component, which is used to administer general settings. Finally, the separation strategy of the architectural model has been described.

4.3 QUESTION MODEL

4.3.1 *Introduction*

In general, the QM constitutes a data model to represent tests, questions and actions according to students' responses. It defines the structure of these elements and enables content reuse across different not only e-assessment systems and tools. In this section, the design of the QM is presented, which is used by the question modeling component to create questions. Firstly, Section 4.3.2 presents the requirements regarding the design of the QM. Then, Section 4.3.3 briefly looks into related work in the field of modeling and exchanging assessment and learning content, before the own QM approach is proposed in Section 4.3.4. Parts of this section were already published in [SW12].

4.3.2 *Requirements*

The QM abstractly determines the structure of the question (i.e., question text, answer options, feedback, etc.) and how to respond/react to students' responses (e.g., giving feedback when a certain answer option has been selected). In the following, the term *student's response* is used generic for both student's answer(s) selected or given to a question and a solution given for a task. Due to the fact that the (adaptive) test creation is done by the adaptation modeling component, tests are not considered for this model. The QM should not only allow creating traditional questions such as multiple-choice or true/false, but should also allow creating interactive e-assessments by using the interaction possibilities of ICOs. The kind of didactic interactions provided by the ICOs encourages the motivation of the students by allowing more creativity in answering questions or tasks. The model should not be limited to, for example, Java applets (which are very often used in scientific contexts), but should allow integrating a variety of ICO implementations ranging from Java/Flash applets, VRML/X3D models to Microsoft Silverlight applications. Moreover, the QM should not be restricted to integrate ICOs as simple media object without any communication between the ICO and any e-assessment system, but also should allow two-way communication in order to set up questions/tasks and to get information about students' interactions with the ICO. This conforms to the requirements for a *high-level integration* as stated by Thomas et al. [TAA+05]. A limitation of a very large number of systems and tools is that they are not using any educational technology standard. Standards should play an important role, not only in e-assessment systems and tools, because they enable content reuse across different systems and tools, minimize the effort in implementing and increase the willingness of adopting the approach. That is the reason why this QM approach where possible should be built on established standards and specifications. The requirements for the design of the question modeling approach are summarized as follows:

- *Representativeness*: It should be capable of representing questions.
- *Processability*: It should be capable of processing student's responses resulting from interacting with the questions.
- *Generality*: It should be capable of representing traditional question types (e.g., multiple-choice, true/false and associate).
- *Interactivity*: It should be capable of creating questions using ICOs and their interaction possibilities.
- *Communication*: It should be capable of establishing two-way communication with an ICO (and any other e-assessment system) initially and at run-time (e.g., for setting up tasks or giving feedback).
- *Standardization*: It should be built on established standards and specifications.

4.3.3 *Related Work*

4.3.3.1 *Data Models for Assessment Content Interoperability*

In the last few years, several e-assessment standards and specifications were developed (cf. Section 3.3.4). The most well-known are:

- Moodle XML
- IMS QTI
- ASD/AIA/ATA S1000D Learning Assessment

Although they all clearly differ in the number of elements and how they are related to each other (cf. Section 3.3.4), they all define data structures for representing questions and tests. They include elements such as a question text, answer options and feedback. An example of a question following such a data model is depicted in Figure 4.3. It should be noted that the example does not conform to any standard and specification mentioned above. The question represented as XML file specifies a question title, a question text, three answers, the correct answer, feedback for a correct and an incorrect answer and the score if the student has answered the question correctly. The common procedure of processing such questions is as follows: First of all, (1) the XML file is selected and submitted to a *testing engine*, also referred to as *question engine*, which then interprets this data structure and prepares a graphical representation for the student. In order to be largely independent from the platform used by the students, the output is mostly HTML-based to be displayed by any web browser. Then, (2) the question is presented to the student. As long as the student does not click on the submit button, no information will be sent back to the testing engine. Only when an answer has been selected and the submit button pressed, (3) the answer will be submitted for validation. After that, (4) the testing engine receives

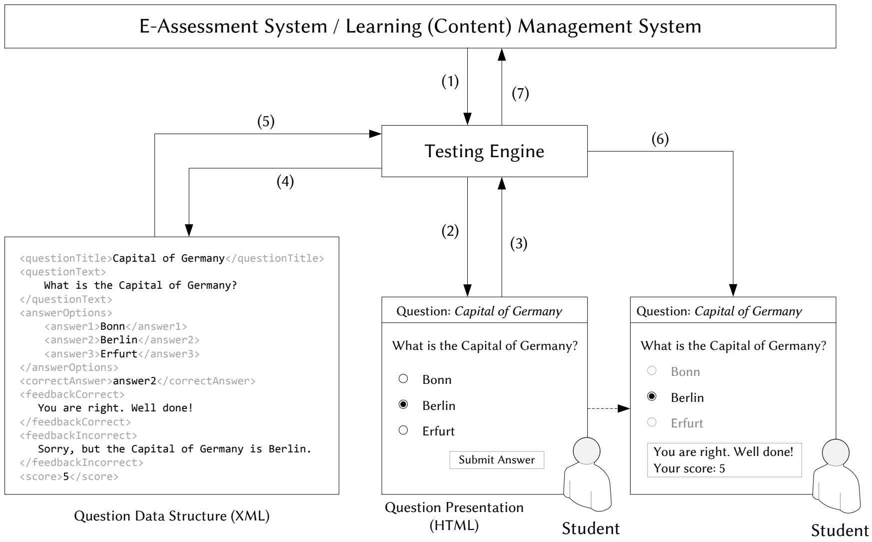


Figure 4.3: Common practice in processing questions and their responses

the answer and checks, whether the answer corresponds with the correct answer specified in the data structure. At the same time, (5) the engine fetches the information about feedback and scores and uses this information to (6) present adequate feedback and/or mark the response(s) of the student. Finally, (7) the results are sent back to an e-assessment system/tool or Learning Management System (LMS)/Learning Content Management System (LCMS). The procedure outlined is the common practice for processing questions in today's e-assessment settings.

So far so good, but when it comes to the integration and use of ICOs into e-assessment processes, the data models provided by the established e-assessment standards and specifications reach their limits. What is missing are data structures that allow for *describing* and *initializing* ICOs and for *processing* the information resulting from interacting with it. None of the above mentioned standards or specifications is able to provide that. However, IMS QTI provides a well-defined extension point (*customInteraction*) for implementing new interaction types. Navarrete et al. [NSHLB11] have used this element to combine IMS QTI with web maps services (Google Maps) in order to enable the computational assessment of geographical skills.

Moreover, the requirement of processing the information resulting from the interactions raises a new requirement to this QM: *communicating* information between ICO and testing engine or any e-assessment system. In traditional e-assessment systems, a testing engine is very closely linked to students' interactions and thus able to respond accordingly. When using ICOs, these objects are mostly located somewhere in the Internet and not locally available such as the question file. For that reason, the testing engine cannot simply get information about students' interactions from these external objects. But, ICOs are characterized by a high level of interactivity. That means, a student is able to interact with the ICO in a variety of ways. This kind of information (e.g., which actions were undertaken by the student in

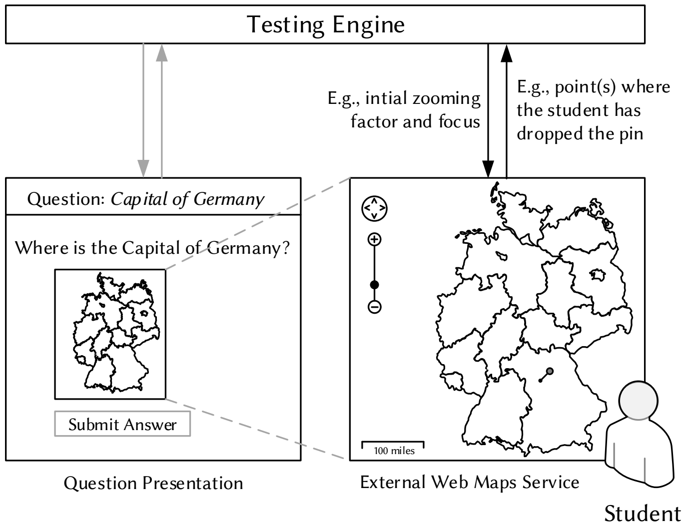


Figure 4.4: Communication with an external object (web maps service)

order to solve the task, cf. Figure 4.4) is particularly important to evaluate students' performance. Otherwise, the ICO is used as simple media object (e.g., compared to an image or an audio file), meaning that the student is mostly not able to freely interact with these objects and only a few or no information (relevant for assessment purposes) are returned. Fontenla et al. [FPC11] use the term *soft integration* for describing this kind of integration. In contrast, *hard integration* provides the system with a comprehensible control over the integrated objects. Moreover, a communication mechanism would not only allow getting information from the ICOs, but also to actively intervene (e.g., by setting or turning on/off specific interaction elements of the ICO) at run-time. This corresponds to the *levels of integration* specified by Thomas et al. [TAA+05]. They distinguish between *low-level* and *high-level* of integration. In a low-level integration, an external object is able to pass information back to the system, whereas in a high level of integration, there is two-way communication between the system and the external object.

Similar to the data model, such a communication mechanism for passing information back and forth between ICOs and testing engines is also not provided in any e-assessment standard or specification. However, in the field of learning content interoperability, there are some standards and specifications for content-to-LMS communication, which are of relevance.

4.3.3.2 *Interface Specifications for Learning Content Interoperability*

The predominant standard/specification in the field of learning content is the Shareable Content Object Reference Model (SCORM)[1]. It encompasses a collection of standards and specifications, which ensure that web-based learning content is created, delivered and represented in a consistent and standardized manner. SCORM defines a communication mechanism for exchanging data

SCORM RTE API

1 http://www.adlnet.gov/scorm [last visited: May 21, 2014]

between SCORM-conforming learning content (so-called Sharable Content Objects (SCOs)) and LMSs. It consists of two parts, a *data model* that specifies which data and a *run-time Application Programming Interface (API)* that specifies how the data is communicated. The data model is specified in Institute of Electrical and Electronics Engineers (IEEE) 1484.11.1-2004 [IEE04] and can be used by LMSs to track items such as status, scores, interactions and objectives. Particularly notable amongst the various elements is the *interactions* data model element. It supports rich interactive content as well as assessments and simulations by communicating the data resulting from interactions with the content [DB06, Ost07]. However, the Run-Time Environment (RTE) API is not appropriate for interfacing ICOs with testing engines because the data model is too minimalistic and does not support new types of interactions that do not currently exist [Joh12]. Furthermore, also the API reaches its limits and needs to be extended when there is a need to communicate with ICOs [DFSP08]. For example, the communication is triggered by the SCO and there is no way for a LMS to initiate calls to functions implemented by a SCO.

Over the last few years, several efforts have been started to use this communication mechanism to interface simulations and learning content [MMHM04, Phi07, DBK08]. However, they mainly focused on integrating DIS or HLA simulations (cf. Section 3.2.5) with SCORM environments. HLA and DIS are undoubtedly the predominant simulation interoperability standards within military simulations, but apart from the military sector, almost none of the interactive tools and simulations available is using it. They have their justifications when realizing real-time war-gaming simulations, but for a simple stand-alone tool aiming at demonstrating any physical or chemical experiment (cf. Section 3.2.4), it does not justify the implementation effort. The limitation of simulations compliant to HLA and DIS as well as the use of the SCORM RTE data model and API, which are mainly targeted on supporting learning/training management, leads to the conclusion that these efforts cannot be simply transferred to realize communication between ICOs and testing engines. In addition, these efforts took a *"narrow, limited approach or could not find enough adopters"*. This was a result of the SISO SCORM - Simulation Interface Standards Study Group [DFSP08], which has investigated these initiatives.

In the meantime, SCORM has gotten a bit long in the tooth because the most widely used version of SCORM (v1.2) is more than 10 years old and there have been a lot of innovations and changes in Internet technology since then. The Advanced Distributed Learning (ADL) initiative has also recognized that SCORM needs get with the times and is currently leading the effort for the next generation of SCORM. Based on feedback from the community of practitioners, *Next Generation SCORM*[2] will address the following requirements [Joh12]:

1. It should be able to handle distributed content
2. It needs to be simpler

2 http://www.adlnet.gov/tla/ [last visited: May 21, 2014]

3. It should handle off-line or long-running content
4. It should provide a way to expose the data it tracks to be simpler
5. It should ensure the sharable content is interoperable and portable
6. It should use current programming standards
7. It should move beyond the single learner approach
8. It should remove sequencing altogether
9. It should have an authentication mechanism, particularly to protect assessment data
10. It needs to track more robust data

Experience API (xAPI)

Requirements 1, 3, 6 and 9 will be met by the Experience API (xAPI), former known as Tin Can API[3]. The xAPI is communication mechanism based on *activity streams* that facilitates and integrates all types of online learning and training. It allows a learning activity provider (e.g., a game) and a learning record store (which can be a component of a LMS) to communicate information about learning experiences. The experiences are expressed as *statements*. Statements are the core of the xAPI. In general, a statement has the following structure:

<actor (student)> <verb> <object> with <result> in <context>

Example statements are *I did this* or *Christian completed 'Final Test' with score 100*. These statements are human readable, meaningful if printed on an interface and also machine readable. The statement definition can be regarded as the data model of xAPI. The following explanations are based on version 1.0.1. The complete structure of a statement is depicted in Figure 4.5. In addition and as the name implies, xAPI has not only a data model that specifies *which* data is communicated, but also an API that specifies *how* the data is communicated. The API is subdivided into four sub-APIs:

- *Statement*: This API is used to track learning records.
- *State*: This API is used to persist the states across devices.
- *Activity profile*: This API is used to store arbitrary key/document pairs related to an activity.
- *Agent profile*: This API is used to store arbitrary key/document pairs related to an agent.

3 `http://www.adlnet.gov/tla/experience-api/`
[last visited: May 21, 2014]

The four APIs are realized by a set of RESTful web service using HTTP methods (i.e., GET, PUT, POST or DELETE). The services use JavaScript Object Notation (JSON) to represent statement objects. This enables out-of-browser content such as mobile apps, virtual worlds, games and simulations using this communication mechanism. Particular noteworthy on xAPI are the well-defined extension points that allow defining new elements as needed. Extensions can be created for *object definitions*, *results* and *contexts* (cf. Figure 4.5).

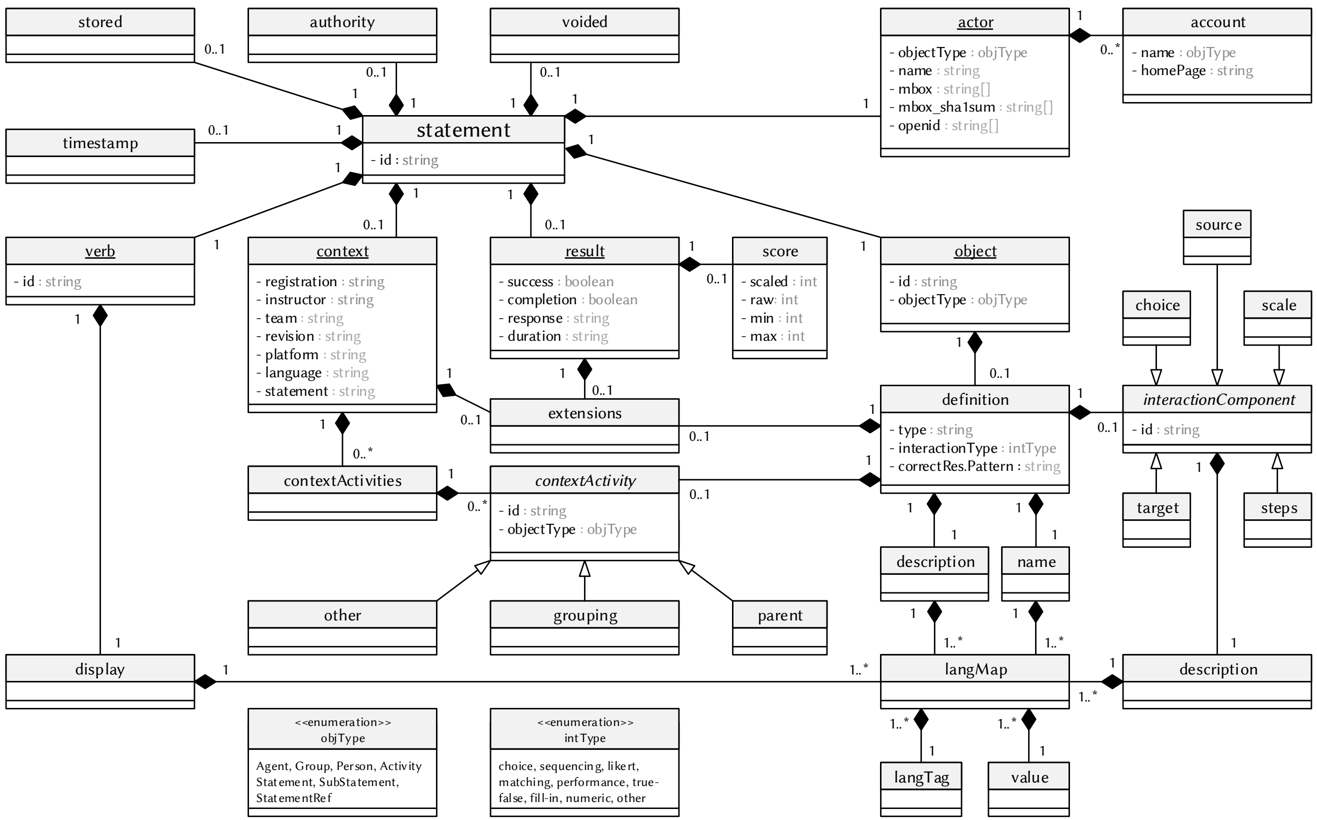


Figure 4.5: xAPI statement structure

Another worth mentioning specification in the field of learning content interoperability is IMS Learning Tools Interoperability (LTI)[4]. It allows learning applications (often remotely hosted and provided through third-party services) to be integrated with platform such as LMSs. Although IMS LTI is an important step in the field of learning content/tools interoperability, however, in its current state (v1.1.1) it is just a launch protocol and not appropriate for a *high* or *hard* level of integration. *IMS LTI*

4.3.4 *Proposed Solution*

Based on the considerations made in the previous sections, it can be stated that there is a need for a QM (data model) that enables *describing* and *initializing* ICOs as well as *processing* students' actions resulting from interacting with them. Additionally, the processing of students' actions in turn requires a communication mechanism for passing information back and forth. In meeting the requirements stated before, a QM, a communication schema and a communication mechanism has been designed. The overall scenario is depicted in Figure 4.6.

Firstly, (1) the e-assessment system delivers an ICO-enhanced question/-task to the testing engine. The question/task includes a reference to the ICO and instructions how it needs to be initialized and launched as well as how students' actions with the ICO will to be processed. The testing engine receives the question/task, detects that an ICO interaction is integrated, interprets the specific elements and attributes attached and (2) generates the statements necessary to set up and initialize the ICO. Then, (3) the initialized ICO is inserted into HTML and (4) presented to the student. Whenever the student interacts with the ICO and makes changes to its elements, the ICO is responsible for (5) informing the testing engine about that. Which ICO elements and/or events are of interest (i.e., whose changes are to be communicated) are specified in the Communication Schema (CS) and the way how the information is passed back and forth are specified by the Communication Mechanism (CM). In case an ICO does not implement the CM, an *interaction mediator* takes on this task by mediating between the testing engine and the ICO. This includes receiving statements for initializing and setting up the ICO, but also sending statements about students' interactions communicated by the ICO. The testing engine evaluates this information and is able to (6) provide feedback and (7) adjust ICO elements accordingly. Finally, (8) the result(s) of the ICO-enhanced question/task is forwarded to the e-assessment system.

In the following, the main components supporting the scenario namely *question model*, *communication schema* and *communication mechanism* are described in detail.

4 `http://www.imsglobal.org/lti/` [last visited: May 21, 2014]

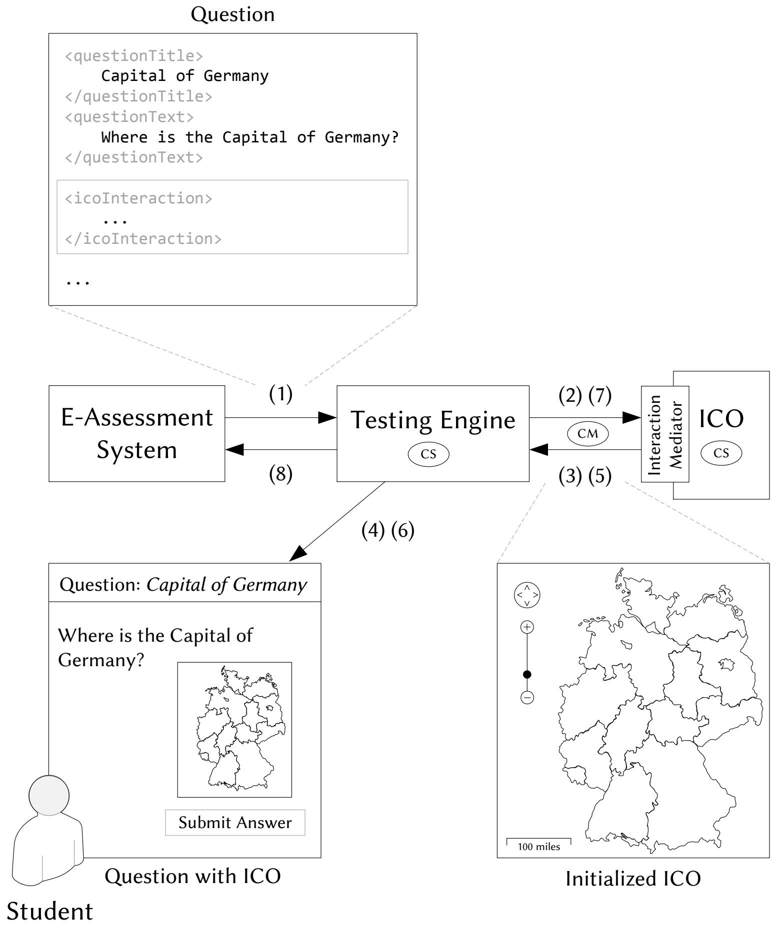


Figure 4.6: Overall communication scenario

4.3.4.1 *Question Model and Communication Schema*

Although the analysis of e-assessment standards and specifications (cf. Section 3.3.4) has pointed out that they do not allow integrating and using ICOs in e-assessment processes, however, they provide an established mean for representing assessment content and responses. This enables content reuse across different systems and tools. For that reason, a valid solution for the integration of ICOs should be based on one of these standards/specifications. In this QM approach, the IMS QTI specification is used as basis on which the own approach has been built on. One reason for that is the fact that the QTI specification is widespread and can be regarded as de-facto standard in e-assessment settings. It is not limited to any specific LMS/LCMS and provides fine-grained subdivision of assessment content, more than 20 interaction types and several other assessment options. But, the decisive factor was the opportunity to extend the specification using the *customInteraction* element. This extension mechanism enables including new types of interactions, which are currently not specified (cf. Section 3.3.4.3).

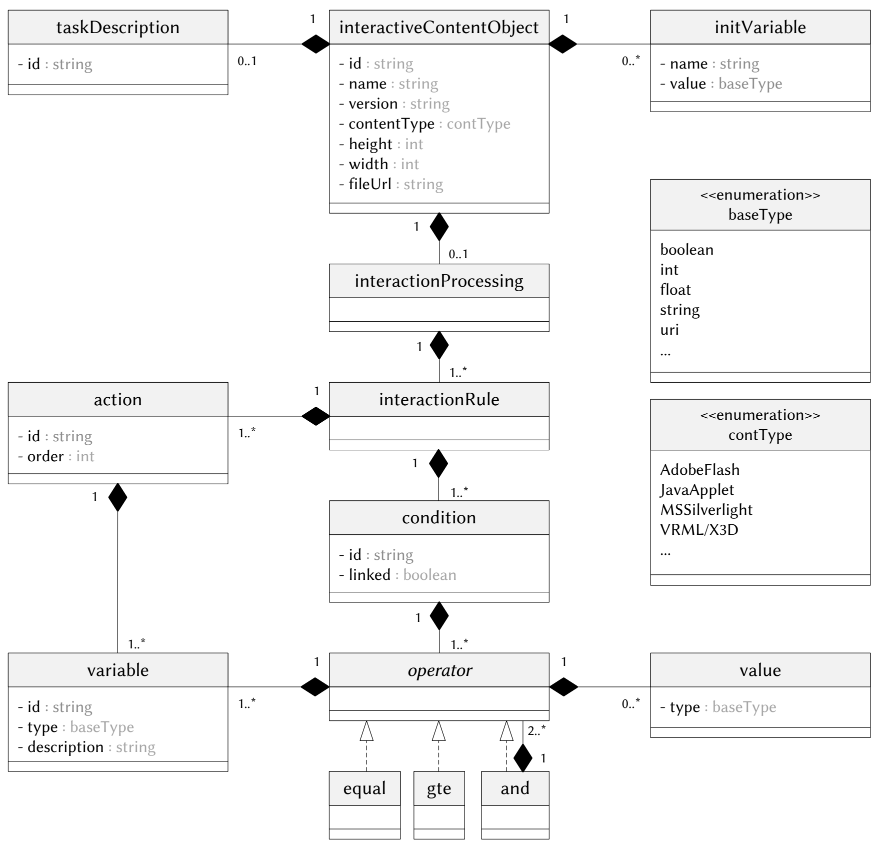


Figure 4.7: Data model for integrating ICOs (MICO)

The own approach attaches right here and extends the established specification with a data structure for the integration of and the interaction with ICOs. That means, the assessment content structure given by QTI is kept and specifically extended. The data structure for this custom interaction (*interactiveContentObject*) is depicted in Figure 4.7.

The data model is called MICO (Model for integrating Interactive Content Objects). It enables the creation of new types of questions, which integrate the interaction possibilities of ICOs. A first version of MICO (v1.0) was already published in [SW12]. This version of MICO (v2.0) has several extensions and improvements. This applies in particular for processing of interactions. MICO consists of several components, which are closely connected to each other: The component *interactiveContentObject* is defined to represent any type of interactive and digital tool that enables students to actively discover concepts or subjects, to conduct experiments by manipulating data and observing the effects of change and to create and test hypotheses (cf. Section 3.3.2). In order to configure how to show the ICO, various attributes have to be defined. This includes a unique identifier (*id*), the *name* of the ICO as well as if it is a Java applet, a VRML/X3D scene (*contentType*) or something else. Other attributes specify the size of the presentation (*height* and *width*) and the path to the archive containing the ICO (*fileUrl*). If the ICO needs additional data for initialization, the component *initVariable* is used to specify key/value

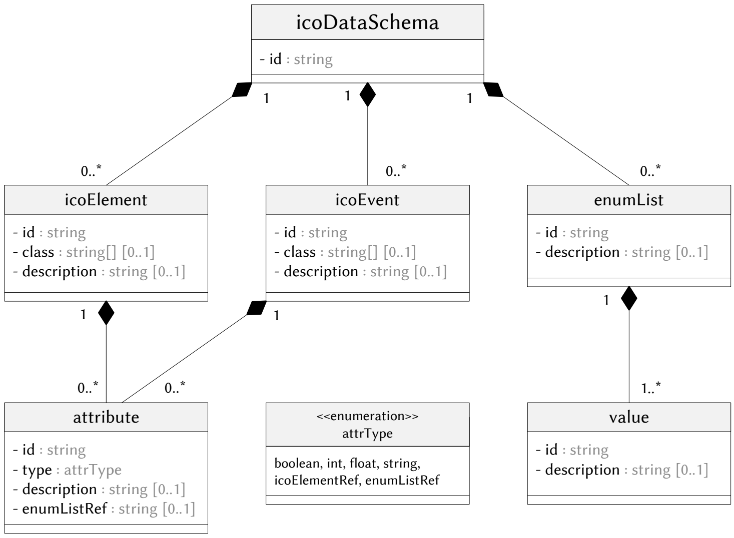


Figure 4.8: MICO communication schema

parameter pairs. The component *taskDescription* is defined to specify the task(s) to be done. It may consist of a short introduction and after that, the actual task is posed to the students.

The components mentioned above allow setting up interactive e-assessments, but they do not yet enable communication nor responding to information resulting from them. As we shall see, the latter one is realized by the *interactionProcessing* component. However, the fundamental basis of a reliable communication and processing of communication data is a consistent *Communication Schema* (CS). The data structure for the MICO CS (*icoDataSchema*) is shown in Figure 4.8.

The component *icoElement* is defined to represent any element provided by ICOs to interact with (e.g., a button). An ICO consists of at least one of such an (interactive) element. By interacting with such an element, events are triggered (e.g., starting a specific procedure). The component *icoEvent* is defined to specify such events. Instances of both components are clearly identified by a unique identifier (*id*), optionally assigned to one or more classes (*class*) and optionally described (*desc*). Additionally, an *icoElement* or *icoEvent* consists of a set of *attributes*. An attribute value is either an integer, string, an element of an enumerated list (*enumList*) or a reference to another ICO element. The latter one allows establishing relationships between different ICO elements.

The *icoDataSchema* can be regarded as an external representation of ICO elements/events. The schema is used by the ICO and the interaction mediator to generate uniform communication messages and by the testing engine to process them accordingly. But, the communication is not limited to one direction. The testing engine is also able to control the ICO remotely by setting/changing element/event attributes arbitrarily. It is important to note

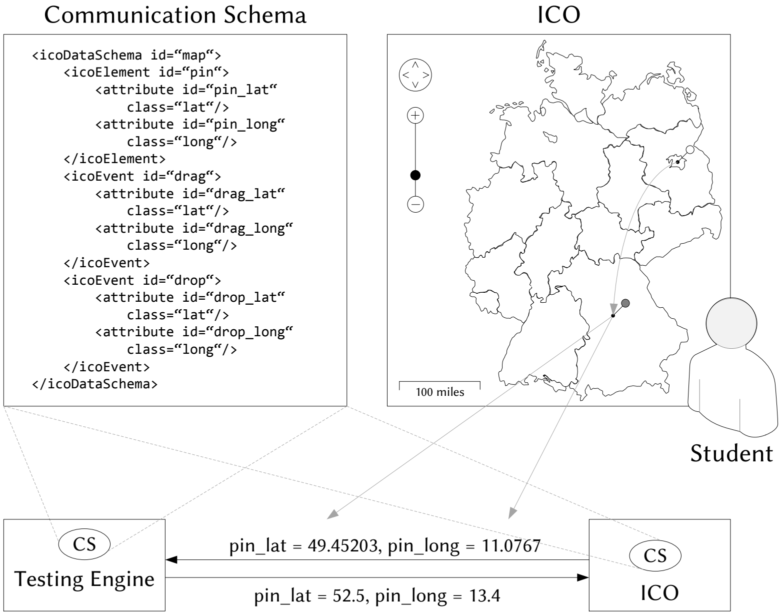


Figure 4.9: Exemplary message exchange based on a predefined CS

that an ICO needs to be slightly updated by the ICO developer in order to be compliant with the specific schema. This includes both the communication of element/event attributes as well reacting on attribute updates.

An exemplary message exchange based on a predefined CS is presented in Figure 4.9. The example shows an *icoDataSchema* that specifies one element (*pin*) and two events (*drag* and *drop*). The pin on the map has two attributes (*pin_lat* and *pin_long*) representing the latitude and longitude of the pin on the map. The two events also have two attributes. Here, the coordinates specify from where the pin is dragged and to where it is dropped. The definition of this schema requires the ICO to inform the testing engine whenever the pin element changes its attributes or one of the two events has been triggered. The example shows a communication message containing pin attribute values. This kind of information can be used by the testing engine to generate results or to provide feedback. In addition to tell the student that the task was not solved successfully, the testing engine is able to show the right solution inside the ICO. This could be done by changing the pin coordinates and submitting them back to the ICO. The ICO detects these attribute updates and changes the pin coordinates accordingly.

It should be noted that ICOs tend to generate a relatively large quantity of data about what the student did during the task. This could easily result in excessive number of communication messages sent out. However, the testing engine filters out the messages, which are of interests (and need to be further processed) and which are not. This is realized using the *interaction-Processing* component. It is light-weight programming language that allows processing students' responses resulting from interacting with ICOs in order to determine whether feedback should be provided or how the responses are converted into results. That means, a significant part of the assessment logic is realized by this component. Control structures are the basis of the *interactionProcessing* component. They allow defining interaction rules *interactionRule* to express, for example, feedback and marking schemes. An interaction rule consists of a *condition* and an *action*. A *condition* consists of at least one *operator*, which usually takes a *variable* and a *value* as input. But, it is also possible to have only variables. Variables either refer to element/event attributes specified in the CS or to variables hold by the testing engine. In contrast, values represent predefined boolean, int, string values. An operator and its variable(s)/value(s) can be regarded as a boolean expression. When an expression proves to be true, the condition has been fulfilled and the corresponding actions will be carried out. A condition can have more than one expression. The attribute *linked* specifies whether all expressions need to be true to satisfy the condition (conjunction) or one true expression is sufficient (disjunction). Furthermore, expressions could be nested arbitrarily using *and* and *or*. The operators provided by MICO are as follows:

- *equal*: This operator allows expressions such as $X == 5, Y ==$ *'string'* or *'X == Y'*, where X and Y are variables.
- *gte*: This operator allows expressions such as $X \geqslant 5$ or *'X ⩾ Y'*, where X and Y are variables.
- *lte*: This operator allows expressions such as $X \leqslant 5$ or *'X ⩽ Y'*, where X and Y are variables.
- *not*: This operator allows expressions such as *'!X' or '!Y'*, where X and Y are variables.

The component *action* is used to define instructions to be triggered by satisfied conditions. An instruction is typically defined by assigning values to variables. Variables refer to outcome variables hold by the testing engine as well as element/event attributes specified in the CS. Due to the fact that an action can consist of more than one instruction, the sequence in which the different instructions are performed is perhaps important. For that reason, the attribute *order* allows specifying a chronological order of actions.

Both, for conditions (getting) and actions (setting), a consistent way of selecting the desired variables/attributes is needed. The way used in this approach is inspired by the way how jQuery matches elements in a HTML document. jQuery[5] is one of the leading JavaScript libraries for document traversing, event handling, animating, etc. In general, the MICO selector allows to:

- Select ICO attribute elements based on a given *id*
- Select a set of ICO attribute elements based on a given *class* name
- Select testing engine variables based on a given *id*

5 `http://www.jquery.com/` [last visited: May 21, 2014]

The general syntax of MICO selectors is depicted in Figure 4.10 and given in Extended Backus-Naur Form (EBNF) in the following:

Selector = Target, '("', (Identifier | Class), '")', '.', Action ;
Target = 'this' | 'ico' ;
Identifier = '#', Characters - '"' ;
Class = '.', Characters - '"' ;
Action = 'getValue ( )' | ('setValue', '(', Characters, ')') ;
Characters = ? All visible characters ? ;

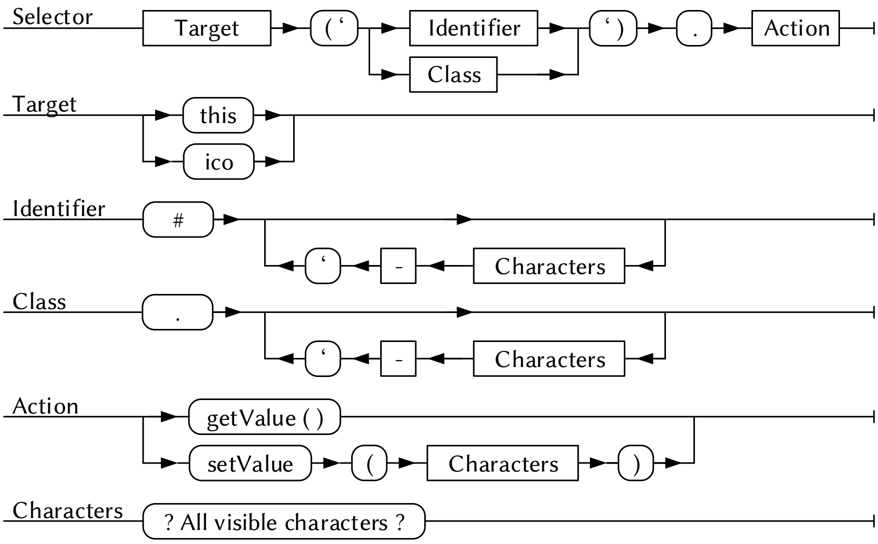


Figure 4.10: Syntax diagram of MICO selectors

Target specifies whether the variable refers to an internal testing engine variable (*this*) or to an external ICO attribute element (*ico*). The next symbols specify the variables/attributes to be selected. In order to select specific variables/attributes, an *Identifier* is used. Due to the fact that an identifier should be unique within the set of variables/attributes, a single, unique variable/attribute is expected to be selected. In contrast, if more than one variable/attribute need to be selected, a *Class* needs to be defined. An *Action* specifies the action to be performed on the variables/attributes selected (*getValue*/*setValue*). Exemplary selectors are listed below:

```
ico ( "#pin_long" ) . getValue ( )
ico ( "#pin_lat" ) . setValue ( 52.5 )
ico ( "#drag_lat" ) . getValue ( )
ico ( ".lat") . getValue ( )
this ( "#var_1" ) . getValue ( )
this ( "#SCORE" ) . setValue ( 5 )
this ( ".title" ) . setValue ( "Map" )
```

These selectors are used by the testing engine to bind the abstract variable declarations with concrete values. Then, the expressions containing the selectors will be evaluated and if the respective condition is satisfied, the related actions will be carried out.

4.3.4.2 *Communication Mechanism*

As for the question data model, a valid solution for the exchange of communication messages (according to a predefined CS) should be based on an established standard/specification. In this QM approach, the Experience API (xAPI) specification is used as basis for the own approach. A reason for that is the general applicability, the high level of flexibility and its extensibility. Due to the fact that the xAPI specification is the basis on which the next generation of SCORM will be built on, it can be expected that it will become the de-facto standard communication mechanism in e-learning settings in the future. This also makes this approach very future-oriented. The main structure of the specification was kept and specifically extended using the well-defined *extensions*. Extensions are available as part of the *object*, *result* and *context* component (cf. Figure 4.5). An extension is defined by a *map*, which consists of a set of key/value pairs. The keys of that *map* must be Uniform Resource Identifiers (URIs) and values can be arbitrary data structures.

In general, the run-time communication between an ICO and the testing engine is based on a *hand-shake* between both parties. That means that when an ICO is initialized and the student interacts with it, the ICO reports element/event changes to the testing engine, which processes this information. If there is a need to communicate information back to the ICO (e.g., element updates), this will be done immediately afterward. For reporting element/event changes to the testing engine, the *statement API* is used. It is the basic communication mechanism of the xAPI and allows storing a statement or a set of statements. Hence, all information reported by the ICO is encapsulated in such statements, using the predefined elements as well as element extensions. As mentioned above, the main structure of a statement is as follows:

<actor> <verb> <object> with <result> in <context>

In this approach, the *actor* is a *person* (student) interacting with an ICO. It is uniquely identified by a *name* and a corresponding e-mail address (*mbox*). The *verb* specifies the action between the actor and the object. The xAPI specification does not dictate any particular verbs, but recommends a set of *core verbs* such as *experienced*, *completed* or *interacted* that can be used to specify actions. The *object* of the statement is the ICO element or event the actor is interacting with. It is uniquely identified by an *id* and described by a *name* and a *description*. The id of the object corresponds to the *id* of the icoElement/icoEvent specified in the icoDataSchema (cf. Figure 4.8).

In this approach, the verb *interacted* has been used to declare students' interactions with ICO elements/events. The verb component does not only contain a human readable representation of the verb definition (*display*), but also an URI that corresponds to that definition. The *result* is used to hold the resulting outcome of students' interactions. The predefined elements are not sufficient for representing this kind of information. Consequently, the result component was extended accordingly. The extension holds the attribute values of the ICO element/event, which are referenced by the statement object and the *context* allows adding contextual information. Here, the ICO specifies whose element/event this statement is reporting about. Finally, the statement is expressed in JSON and sent to the testing engine using the HTTP POST method (cf. Figure 4.11).

Usually, the opposite site (i.e., the testing engine) acknowledges the receiving of this statement with a *status code*. For each HTTP method, xAPI defines specific codes that are returned for indicating errors or acknowledgments. This code is hold back as long as the testing engine has not processed the statement completely. This includes the processing of the information contained in the statement according to the *interactionProcessing* rules. It is possible that certain ICO elements need to be changed. To do so, the testing engine takes the shared CS as template and changes element attribute values accordingly. These changes are made available by the use of the *state API*. Generally, this API allows activity providers to persist state across devices. Here, it is used to deposit element attributes and their values to be fetched by the ICO itself.

After the testing engine processes the statement completely, it sends the status code *"200 OK"*, which in turn induces the ICO to request the state API whether attribute changes are available. The request is done using the HTTP GET method and the parameters include the identifier of the ICO (*activityId*) as well as the actor associated with statement submitted before. In order to ensure that only new attribute changes are returned, the *since* element holds the timestamp of the last statement post. In response to the request, the testing engine returns the relevant attributes and their associated values. Once the ICO has processed the returned information completely, the hand-shake is finished. An exemplary hand-shake is depicted in Figure 4.11. It picks up the example given in Figure 4.9. It is important to note that the ICO needs a Uniform Resource Locator (URL) that constitutes the endpoint for the statement/state API calls. This URL, which specifies the location of the testing engine, is communicated when initializing the ICO. In addition, the *name* and the *email address* of the student, which will be presented with the ICO as well as the *id* and the *name* of the ICO (specified according to MICO, cf. Figure 4.7) is sent during this initialization phase. This information is used by the ICO to uniquely identify the actor of the statement.

4.3.4.3 *Feedback Model Extension*

In addition to MICO, which is intended to be used in conjunction with IMS QTI using the well-defined *customInteraction* interface, a second extension is proposed for the QTI data model. It concerns the provision of feedback and hints to be given to the students. Currently, the specification allows defining three types of feedback (cf. Section 3.3.4.3):

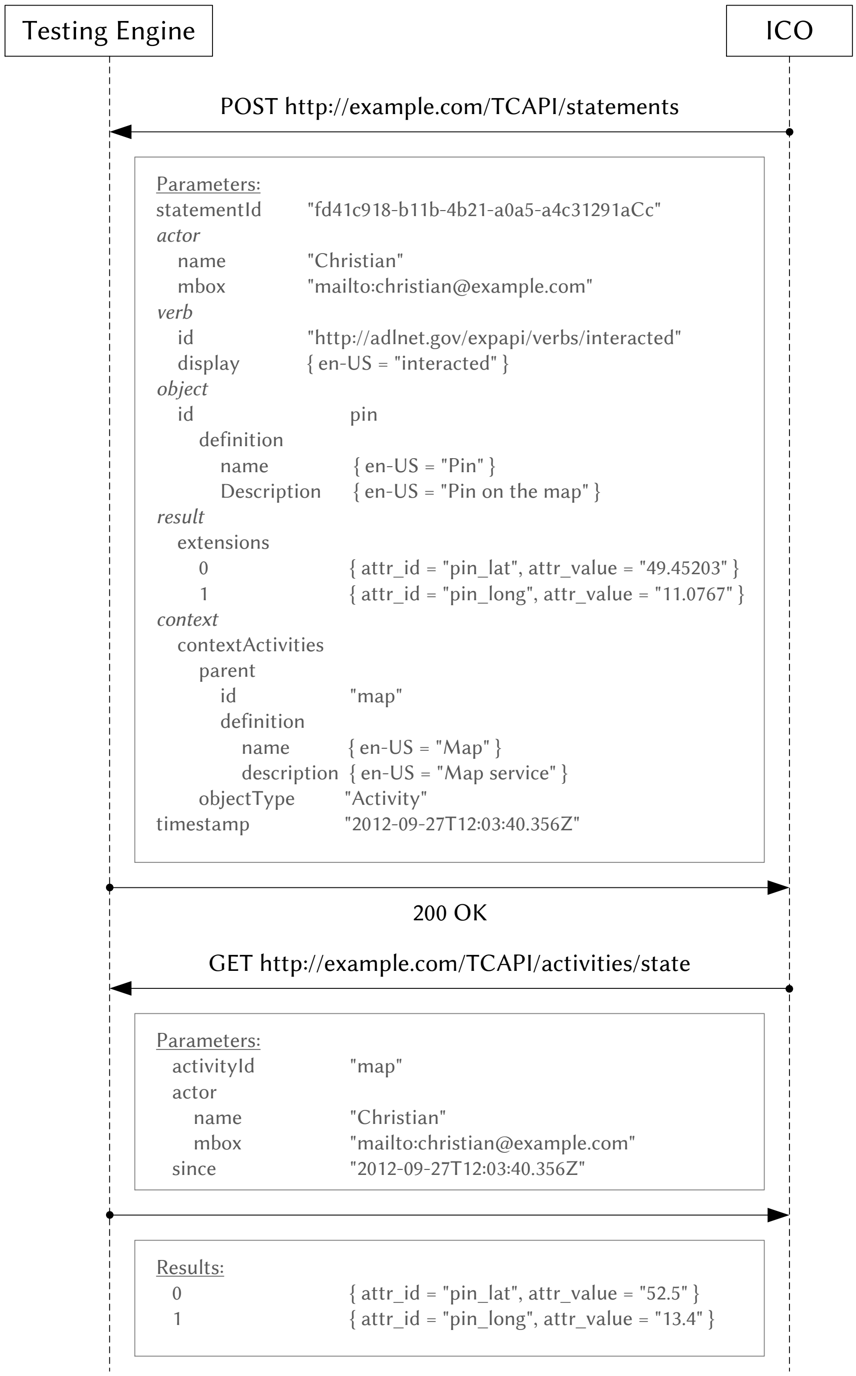


Figure 4.11: Hand-shake between ICO and testing engine

- Integrated feedback (*feedbackInline*)
- Modal feedback (*modalFeedback*)
- Test feedback (*testFeedback*)

Arbitrary feedback can be created using these data structures. However, IMS QTI does not classify feedback. This makes it difficult to select feedback

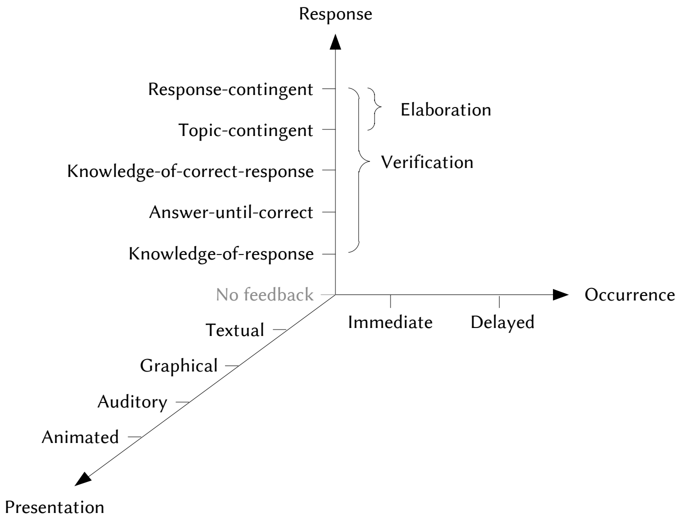


Figure 4.12: Dimensions of feedback (adopted from [SW11a])

that is most suitable for the moment and the respective student. This problem is tackled by the proposed feedback model extension. In general, the extension is based on the three *dimensions of feedback* (cf. Figure 4.12) proposed by Saul and Wuttke [SRW10, SW11a]:

- *Response*: This dimension refers to the type of information provided to the students.
- *Presentation*: This dimension refers to the way of presenting feedback.
- *Occurrence*: This dimension refers to the timing of feedback.

This structure is incorporated into the three feedback structures of the QTI data model (cf. Figure 4.13). The additional elements (attributes) are underscored. The attribute (*responseType*) allows specifying whether the feedback provides verificative and/or elaborative information. As an example, if the feedback is *"Your answer is wrong"*, this feedback element would be classified as *knowledge-of-response*. Simply put, it assigns a boolean value (i.e., true or false) to answers selected or given by the student. While this type of feedback is essential for verification purposes, it does not provide any information that would extend the students' knowledge or provide additional insight into possible errors in understanding. In contrast, *response-contingent* feedback gives response-specific feedback that explains why the incorrect answer was wrong and why the correct answer is correct. The attribute (*presentationType*) allows specifying the form of presentation (e.g., textual, graphical, auditory and animated or a combination of these). This information is relevant, for example, in cases where the device used by the student does not allow showing certain media types. The third dimension of feedback is not realized as an own attribute because the timing of feedback is determined

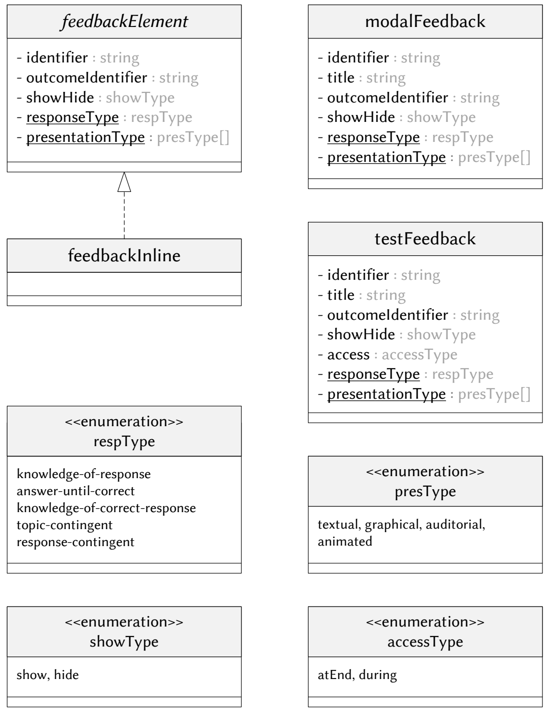


Figure 4.13: Proposed IMS QTI feedback model extension

by the use of one of the three feedback types. With the use of the *showHide* attribute and response processing, feedback can be presented to the student at every point within the test.

4.3.5 *Summary*

This section has presented the design of the QM. The QM enables describing and initializing ICOs as well as processing students' actions resulting from interacting with them. Additionally, the processing of responses in turn requires a mechanism for passing information back and forth. Consequently, not only a QM, but also a CS and a CM has been designed. As basis for the QM, the IMS QTI specification was selected and specifically extended with a data model for the integration of and the interaction with ICOs. The data model is called MICO and uses the well-defined extension mechanism provided by IMS QTI (i.e., *customInteraction* element). Although the QM is currently based on IMS QTI, which can be justified by the fact that it is currently the de-facto standard in e-assessment settings, MICO is basically QTI-independent and can also be integrated in other standards or specifications. In order to enable a reliable communication and processing of communication data between ICO and testing engine, a CS (*icoDataSchema*) has been designed. The *icoDataSchema* can be regarded as an external representation of ICO elements/events. The schema is used by the ICO to generate uniform communication messages and by the testing engine to process them accordingly. Furthermore, a CM has been designed that enables exchanging communication messages between ICO and testing engine. For this mechanism, the xAPI specification was used as basis. The main structure of the specification was kept and specifically extended using well-defined extensions. In order to leverage interoperability among different xAPI-compliant tools or systems, as much information as possible was built into the predefined elements (i.e., *actor*, *verb*, *object*, *context*) and only the *result* element was extended. In general, the run-time communication between an ICO and the testing engine is based on a *hand-shake* between both parties. That means that when an ICO reports element/event changes to the testing engine and if there is a need to communicate information back to the ICO, this will be done immediately afterward. Finally it can be stated that all the components described above can significantly increase the effectiveness of ICOs in e-assessments by allowing a more in-depth review of what a student has and has not mastered.

IN A NUTSHELL: The QM approach

- Represents traditional question types, but also allows creating ICO-enhanced questions/tasks
- Enables describing and initializing ICOs as well as processing students' responses resulting from interacting with them
- Allows passing information back and forth between ICO and testing engine
- Supports a variety of ICOs implementations (e.g., Adobe Flash or Java applets)

4.4 DOMAIN MODEL

4.4.1 *Introduction*

In general, a DM represents the body of knowledge that makes up a (knowledge) domain and allows relating questions and students' features with these fields of knowledge. In this section, the design of the DM approach is presented, which is administered by the domain modeling component. Firstly, Section 4.4.2 presents the requirements regarding the design of the DM. Then, Section 4.4.3 briefly looks into related work in the field of knowledge representation, before the own DM approach is proposed in Section 4.4.4.

4.4.2 *Requirements*

The DM has a key role in the overall architectural model. It should be able represent the body of knowledge that makes up a domain, which serves as the basis for relating questions and students' features. For that reason, the model should allow defining arbitrary elements within these domains as well as defining connections between the different elements to build arbitrary domain structures. The requirements for the design of the domain modeling approach are summarized as follows:

- *Generality*: It should be capable of representing fields of knowledge (domains).
- *Subdivision*: It should be capable of defining arbitrary elements within these domains.
- *Interconnectivity*: It should be capable of defining connections between the different elements to build arbitrary domain structures.

4.4.3 *Related Work*

As mentioned in Section 3.5.4, AASs show remarkable similarities with AHSs in maintaining different models for realizing adaptations. Over the last few years, a variety of reference models in the field of adaptive hypermedia were developed. Due to the fact that the majority of these models have a domain modeling part, Section 4.4.3.1 briefly describes these approaches and analyzes, whether such an approach can be used for the own DM approach. In addition to the specific reference models, there are traditional approaches from the field of knowledge representation that can be used to represent a set of concepts within a knowledge domain. The most common approaches are briefly outlined in Section 4.4.3.2.

4.4.3.1 *Reference Models*

DMs represent the basis for AHSs. Hence, the majority of reference models for AHSs consist of a domain modeling part. As already presented in Section 3.4.3, the most common reference models for are:

- Adaptive Hypermedia Application Model (AHAM)
- Munich Reference Model (MRM)
- LAOS

The DM of AHAM basically consists of fragments, pages and concepts, and their relationships to each other. When taking a closer look, it becomes apparent that the hierarchical structure is strongly oriented towards the taxonomy of hypermedia and supports the realization of the *adaptive presentation* technique. This can be justified by the fact that each concept consists of a presentation specification. However, when striving for another adaptive methods, which are not based on hypermedia (as it is the case here), this structure does not make sense. Summarizing it can be stated that the structure given by AHAM to realize a DM is inappropriate because it is too hypermedia-specific and too closely interlinked with the adaptive presentation technique. All conclusions made for the AHAM can also be applied for models, which are based on AHAM or follow a similar structure. Both MRM and LAOS fall into this category. Whilst the MRM domain meta-model simply defines the DM of AHAM from an object-oriented software engineering point of view, the presentation specification in LAOS is externalized to an own layer and separated from the concepts of the domain. This is a step in the right direction, however, the structure is also strongly oriented towards hypermedia and thus also inappropriate. Summarized it can be stated that the domain modeling part of the reference models are, despite its generic, too deeply rooted in the hypermedia context. Although they provide the necessary structures to represent knowledge domains, however, the definition is limited to hierarchical hypermedia. As a consequence, these reference models are applicable only if the underlying system is an AHS. Due to the fact that the overall approach aims at presenting adaptive questions and tests, the domain modeling part of these models cannot be readily applied.

4.4.3.2 *Knowledge Representation*

Besides the specific reference models, there are traditional approaches from the field of knowledge representation that can be used to represent a set of concepts within a knowledge domain. They all provide structural frameworks for organizing knowledge. The most common approaches are:

- Classification
- Thesaurus
- Ontology

A *classification* is an approach of coding and organizing knowledge according to its subject into different classes, typically arranged in a hierarchical tree structure. Classifications are mainly used in libraries to facilitate subject access by enabling users to find out what documents the library has on a certain subject and to find where the document is located [Cha07]. Basically, classifications can be used to realize DMs. However, the way how classifications are represented opposes the suggestion. Classifications use artificial strings (*notations*) for defining classes and their relationships. Notations can consist of letters, numerals or other symbols. Such notations are difficult to understand for domain authors and recognizing the hierarchical relationships is not very easy. Furthermore, classifications only provide a limited number of relationships (i.e., *part-of* and *is-a*) and thus limit the ability of reasoning about concepts.

A *thesaurus* is an approach of grouping words, also referred to as *terms*, together according to their similarity of meaning. The terms are linked with each other by using different types of relationships (i.e., *hierarchical*, *equivalency* and *associative*) [AD04]. Thesauri bypass the problem of classifications by using natural language notations. This allows describing domains and their concepts adequately. Moreover, thesauri not only allow figuring out terms, which are more general or more specific, but also those that are identical or somehow related. This offers more potential for reasoning. However, the use of a thesaurus for realizing DMs for AASs is problematic. The main reason is the requirement on completeness with respect to the terminological control. This means that all terms in a vocabulary, which have a relationship to another term, have to be identified and marked. Additionally, adequate and unambiguous descriptors need to be found. This requires a tremendous amount of effort and can only be done by domain experts.

An *ontology* is an approach that provides a *"formal specification of a conceptualization"* [Gru95]. A *conceptualization* is an abstract, simplified view of the world. It consists of objects and/or concepts and their properties and relations. This set is then formally specified in a representational vocabulary and semantic (*specification*). Ontologies are not only used to describe the domain, but can also be used to reason about the entities within that domain. Depending on the type of the ontology, they consist of:

- Individuals (instances)
- Classes (concepts)
- Attributes
- Relations
- Restrictions
- Axioms

Ontologies that only consist of individuals, classes and attributes are referred to as *lightweight* ontologies, whereas ontologies, which also include axioms and restrictions, are referred to as *heavyweight* ontologies [CFLGP03]. While lightweight ontologies are little more than taxonomies, heavyweight ontologies additionally allow complex reasoning. Ontologies are commonly encoded using ontology languages such as Simple Knowledge Organization

System (SKOS), Resource Description Framework Schema (RDFS) and Web Ontology Language (OWL).

Classification, thesauri and ontologies share some common approaches for representing and organizing knowledge, but they are developed under different paradigms and serve different purposes. Summarized it can be stated that ontologies are best suited for realizing DMs for AASs. Reasons for this are the relatively general definition and the flexibility of the elements to be used. In addition, when using the natural language to represent the elements, ontologies can replicate classifications and thesauri without their restrictions. Furthermore, ontologies allow realizing any kind of relationship and the numerous amounts of ontology languages enable the creation of formal representations of DMs, which can be exchanged between and interpreted by different applications

4.4.4 *Proposed Solution*

In meeting the requirements stated above, a domain modeling approach has been designed. It is depicted in Figure 4.14.

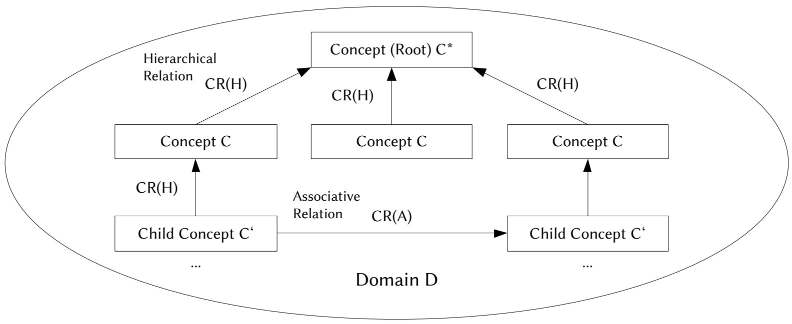


Figure 4.14: Overview of the domain modeling approach

According to the definition of ontologies, the body of knowledge needs firstly to be represented in a set of elements (conceptualization) and then formally specified (specification). The elements of the domain modeling approach are described in the following.

4.4.4.1 *Domains*

The core element of the approach is represented by a *domain*. It embodies the entire field of knowledge, a topic or a collection of content that can be described by one term. This allows making statements about specific user features related to the domain as a whole based on accumulated concept values. Moreover, it allows referring question and tests to specific fields of knowledge. As a consequence, the whole set of domains (each is representing a specific DM) form the knowledge base of the adaptive system. A formal definition of a domain is given in the following:

- Each domain D consists of a set of concepts C and concept relationships CR
- Each domain D has at least one concept C^* that represents the root of the domain
- Each domain D has a set of attributes (e.g., name, description)

4.4.4.2 *Concepts*

A concept represents a part of the underlying domain, which can be used to be overlayed by user features of the UM. In order to allow specializations, a concept can have several child elements (concepts). This kind of relationship is realized by *hierarchical* relations. Additionally, concepts can also be connected by *associative* relations in order to express relationships apart from parent-child structures. Each concept is part of a domain, but can also belong to another domain if it shares this concept (cf. Figure 4.15). A formal definition of a concept is given in the following:

- Each concept C belongs to at least one domain D
- Each concept C defines relationships to other concepts in forms of hierarchical and/or associative relations CR
- Each concept C can have one or more child concepts C', whereas each of them has a hierarchical relation to its parent concept C
- Each concept C has no, one or more associative relations $CR(A)$
- All concepts C, except the root concept C^* are child concepts
- Each concept C has a set of attributes (e.g., name, description)

4.4.4.3 *Concept Relations*

Concept relations are relations that exist between two concepts. Therefore, they serve the purpose of expressing the relationships between the elements of knowledge represented by the concepts. Such relationships can be both *associative* and *hierarchical*. While hierarchical relations are limited to a specific domain for expressing hierarchical links between two concepts, associative relations allow making connections between concepts from different domains (cf. Figure 4.15). The kind of connection is labeled by an appropriate relation type. A formal definition of a concept relation is given in the following:

- Each concept relation CR defines a source concept C and a target concept C'
- Each concept relation CR is unidirectional
- A concept relation CR is either hierarchical $CR(H)$ or associative $CR(A)$
- An associative concept relation $CR(A)$ can exist between a concept C and a concept of another domain C'
- A hierarchical concept relation $CR(H)$ can only exist between two concepts C and C' of the same domain
- Each concept relation CR is specified more clearly by a relation type T

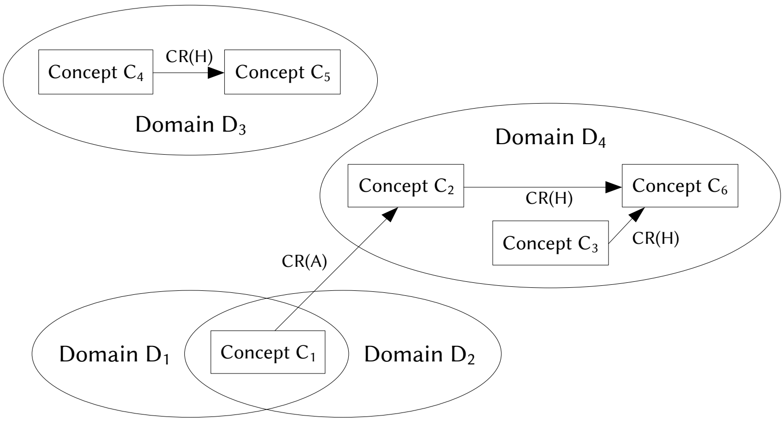


Figure 4.15: Domain concepts and concept relations

4.4.4.4 *Specification*

According to the definition of ontologies, besides the need for a conceptualization, there is a need for a formal specification of the DM in order to allow interoperability between different systems. For that purpose, different formal languages can be used. In the context of ontologies, mainly frame-based languages and description logic from the field of the semantic web are used. These languages allow drawing conclusions about the concepts and their relations. However, due to the fact that the DM to be specified is aimed at exclusively representing and structuring knowledge, less importance was attached to these languages. The priority is on exchanging the DM over the WWW. For that reason, only formal languages were analyzed, which allow specifying domains and their structures and are also established in the WWW. The following languages were considered in the analysis:

- Extensible Markup Language (XML)[6]

6 `http://www.w3.org/TR/REC-xml/` [last visited: May 21, 2014]

- Resource Description Framework Schema (RDFS)[7]
- Darpa Agent Markup Language (DAML)[8]
- Web Ontology Language (OWL)[9]
- Simple Knowledge Organization System (SKOS)[10]
- XML Topic Maps[11]

In order to identify an appropriate language, a set of requirements were defined. Both functional as well as non-functional requirements were considered in the analysis. Non-functional requirements were:

- *Modularity*: This requirement defines the ability to define domains, concepts and concept relations.
- *Hierarchical structuring*: This requirement defines the ability to realize hierarchical relations.
- *Interconnectivity*: This requirement defines the ability to realize associate relations.

Whereas, functional requirements were:

- *Scalability*: This requirement defines the ability to handle a growing amount of domains, concepts and relations.
- *Reusability*: This requirement defines the ability to reuse existing DMs.
- *Standardization*: This requirement refers to the standardization of the language.
- *Degree of popularity*: This requirement refers to the degree of popularity of the language.
- *Self-descriptiveness*: This requirement refers to the extent how the language describes itself.

The detailed analysis of the different languages according to the different requirements is provided in [Gen11]. As a result (cf. Table 4.1), it can be stated that both SKOS and OWL meet the requirements satisfactorily. However, the reason to go for OWL for specifying the DM was justified by the fact that OWL has a comparatively higher degree of popularity. Moreover, a variety of tools exist that support OWL and thus allow further processing of the ontologies created. Furthermore, OWL has the greatest range of functionalities and provides the necessary flexibility to later add new adaptive methods.

7 http://www.w3.org/TR/rdf-schema/ [last visited: May 21, 2014]
8 http://www.w3.org/TR/daml+oil-reference/ [last visited: May 21, 2014]
9 http://www.w3.org/TR/owl2-overview/ [last visited: May 21, 2014]
10 http://www.w3.org/TR/skos-reference/ [last visited: May 21, 2014]
11 http://topicmaps.org/xtm/ [last visited: May 21, 2014]

REQUIREMENT	XML	RDFS	DAML	OWL	SKOS	XML
Modularity	++	++	++	++	++	++
Hierarchical structuring	+/−	++	++	++	++	++
Interconnectivity	− −	+	++	++	++	++
Scalability	−	+/−	++	++	++	+
Reusability	−	+/−	+	+	++	+
Standardization	++	++	− −	++	++	++
Degree of popularity	++	++	−	++	−	+
Self-descriptiveness	+/−	+/−	+/−	+/−	+/−	+/−

Table 4.1: Results of the formal languages analysis (adopted from [Gen11])

++ = very satisfied

\+ = satisfied

+/− = neutral

− = dissatisfied

− − = very dissatisfied

4.4.4.5 *Visualization*

In order to intuitively support the user in dealing with complex domain structures, a user-friendly visualization is mandatory. The visualization should require prior knowledge as little as possible, instead the meaning of the different domain elements and their relationships should be intuitively understandable. Katifori et al. [KHL+07] present six groups of visualization techniques namely:

- *Indented list*: This visualization represents ontologies as a tree.
- *Node-link and tree*: This visualization represents ontologies as a set of interconnected nodes, presenting the taxonomy with a top-down or left to right layout.
- *Zoomable*: In this visualization, the user can zoom-in to the child nodes in order to enlarge them.
- *Space-filling*: This visualization is based on the concept of using the whole screen space by subdividing the space available for a node among its children.
- *Focus+context or distortion*: This visualization is based on the notion of distorting the view of the presented graph in order to combine context and focus.

- *3D information landscapes*: In this visualization, documents are placed on a plane as color- and size-coded 3D objects.

The first five techniques were considered in the analysis. The last technique was left out because the focus was laid on 2D techniques. 3D techniques would introduce an additional complexity. In order to select an appropriate technique, a set of requirements were defined:

- *Completeness*: This requirement defines the ability to display the entire DM including its concepts and concept relationships.
- *Representation*: This requirement defines the ability to visualize the structure of the DM.
- *Degree of presentation*: This requirement defines the ability to freely adjust the degree of the presentation of the DM.
- *Scalability*: This requirement defines the ability to handle a growing amount of domains, concepts and relations.
- *Realizability*: This requirement defines the ability to realize the visualization using limited resources (e.g., in web-based environments).
- *Complexity*: This requirement refers to the complexity of the visualization technique.

The detailed analysis of the techniques according to the different requirements is provided in [Gen11]. As a result (cf. Table 4.2), it can be stated that *indented list* largely meet the requirements satisfactorily. In addition it is the predominant visualization method in a variety of established ontology management tools such as protégé[12]. As an alternative to the intended list, the *node-link and tree* as well as the *zoomable* technique could be identified. Node-link and tree is suitable for displaying the integrated structure of the DMs because all elements could be represented. However, the amount of elements to be displayed needs to be limited in order to avoid problems regarding processing efficiency and delays. In contrast, zoomable techniques can be used to focus on a specific concept. Summarized it can be stated that both methods are best suited as a supplement to the indented list.

4.4.5 *Summary*

This section has presented the design of the domain modeling approach. The DM has a key role in the overall architectural model because it represents the body of knowledge that makes up a (knowledge) domain and serves as the basis for relating questions and students' features. As made clear, ontologies are the approach that best matches the requirements stated above. Due to the fact that the aim was to represent and not to reason about domains, particular

12 http://protege.stanford.edu/ [last visited: May 21, 2014]

importance was attached to the ontology elements concepts, attributes and relations. Other ontology elements such axioms and restrictions played a secondary role in the further considerations. This kind of ontologies is also referred to as *lightweight* ontologies [CFLGP03]. In contrast, *heavyweight* ontologies also include axioms and restrictions and allow complex reasoning. In order to support interoperability of the model and its information, the OWL specification was chosen as formal specification for the DMs.

IN A NUTSHELL: The DM approach

- Enables representing a wide field of knowledge through a set of associated concepts
- Supports the creation of arbitrary domain structures by defining hierarchical or associative relationships between the concepts
- Ensures interoperability of the model and its information through the use of the OWL specification

REQUIREMENT	INTENDED LIST	NODE-LINK AND TREE	ZOOMABLE	SPACE-FILLING	FOCUS+CONTEXT OR DISTORTION
Completeness	+	++	+	+/−	+
Representation	++	++	+/−	−	− −
Degree of presentation	+	+/−	+	− −	++
Scalability	+	−	+/−	−	+
Realizability	++	+	+	+	−
Complexity	++	+/−	+/−	+	− −

Table 4.2: Results of the visualization techniques analysis (adopted from [Gen11])

++ = very satisfied

\+ = satisfied

+/− = neutral

− = dissatisfied

− − = very dissatisfied

4.5 USER MODEL

4.5.1 *Introduction*

In general, the User Model (UM) in an adaptive system represents the essential information (*features*) about each individual user that is needed to provide the adaptation aimed at and to distinguish among different users. In this section, the design of the UM is presented, which is administered by the user modeling component. Firstly, Section 4.5.2 presents the requirements regarding the design of the UM. Then, Section 4.5.3 briefly looks into related work in the field of user modeling, before the own UM approach is proposed in Section 4.5.4. Parts of this section were already published in [HSW12].

4.5.2 *Requirements*

Adapting questions, tests and feedback to individual users (students)[13] requires the system to be able to make inferences about the users. The basis on which these inferences are made is the UM and the individual user profiles, respectively. It should allow representing a variety of user features ranging from user's knowledge to their individual characteristics (e.g., interests or background). The model should not be restricted to define discrete values as user features, but should also allow relating them to different domains and topics. These relationships and the information included in the user profiles should allow making complex assumptions and reasoning about the users. Furthermore, the model should also be flexible enough to be able to adapt to changing user features as they occur in learning/assessment processes. Finally, the model should incorporate established standards and specifications to support interoperability of the structures created and information gathered. The requirements for the design of the user modeling approach are summarized as follows:

- *Generality*: It should be capable of representing a variety of user features.
- *Connectivity*: It should be capable of relating user features to different topics.
- *Flexibility*: It should be flexible enough to adapt to changing user features.
- *Inferential Capabilities*: It should be able to support complex assumptions and reasoning based on information about the users.
- *Standardization*: It should be built on established standards and specifications.

13 In the following, the terms user and student are used synonymously

4.5.3 *Related Work*

A UM is an essential model of each adaptive system. However, the amount of user features represented in an adaptive system is *"determined by the kind of personalized support it provides"* [BM07]. The related work in the field of user modeling is structured along three dimensions:

- *What is modeled?*
- *How it is modeled?*
- *How the model is initialized and maintained?*

4.5.3.1 *What is modeled*

According to Brusilovsky and Millán [BM07], the most popular user features modeled by adaptive systems are:

- *Knowledge*: This user feature represents user's knowledge of the domain or subject.
- *Interests*: This user feature, also referred to as *preferences*, represents user's feeling of wanting to know or learn about something.
- *Goals*: This user feature, also referred to as *tasks*, represents the purpose for a user's work with an adaptive system.
- *Background*: This user feature represents user's previous experience outside the core domain of a specific adaptive system.
- *Individual traits*: This term aggregates the user's features that together define a user as an individual (e.g., personality factors, cognitive or learning styles)
- *Context*: This user feature represents the context of the user's work (e.g., user platform or user location)

In addition, Sosnovsky and Dicheva [SD10] complement this list with the following features:

- *Demographic information*: This user feature represents user's demographic characteristics (e.g., age or native language).
- *Emotional state*: This user feature represents user's emotions.

4.5.3.2 *How it is modeled*

After having selected a set of features to be considered, the next task is to define how these features are modeled. There are different approaches for representing student features (e.g., scalar models [BE94] or bug models [Bur82]), however, they can be broken down into two main approaches [SD10]:

- *Stereotype user modeling*: This approach clusters students into standardized and simplified groups, called *stereotypes* (e.g., novice, intermediate, expert). More specifically, it maps a specific combination of student features to one of the stereotypes. After that, all students belonging to the same stereotype are treated in the same way (cf. Figure 4.16).
- *Overlay user modeling*: This approach uses a Domain Model (DM) (cf. Section 4.4) as a template and associates certain values to each item in the domain (e.g., topics or concepts) representing student's features (e.g., knowledge, interests, goals, etc.) between the student and the item (cf. Figure 4.17).

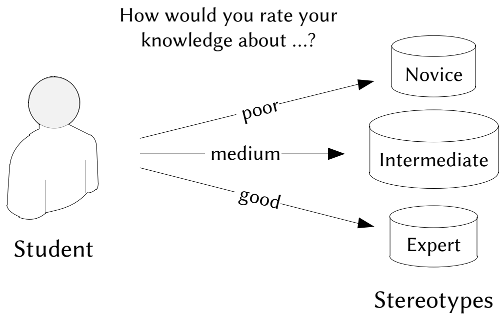


Figure 4.16: Stereotype user modeling

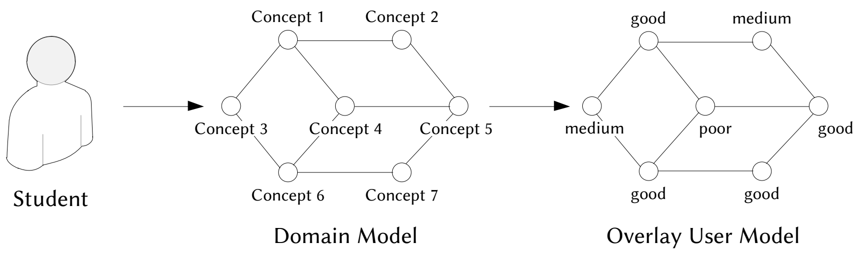
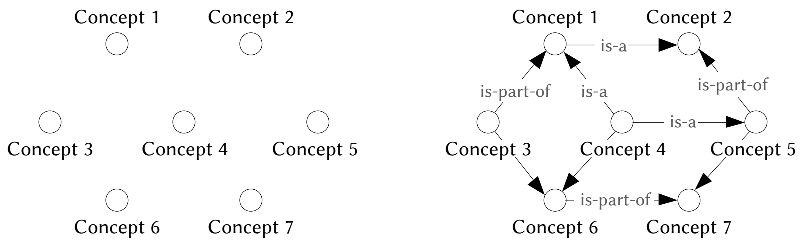


Figure 4.17: Overlay user modeling

Although stereotypes provide a useful mechanism for building UMs on the basis of a small amount of information [Ric79], the overlay approach allows a more fine-grained concept-based modeling of students' features (e.g., a knowledge level of a particular concept) [SD10]. Implementing overlay user modeling requires two essential components [BM07]:

1. *Domain Model* (DM): This component represents the body of knowledge about the domain and its structure (cf. Section 4.4).
2. *Overlay model*: This component specifies what data about each item of the domain structure is stored.

According to Brusilovsky [Bru03], DMs can roughly be divided into:



(a) Vector model

(b) Network model

Figure 4.18: Comparison between vector and network models

- *Vector models*: This kind of model represents the domain through a set of independent (unrelated) concepts (cf. Figure 4.18a).
- *Network models*: This kind of model represents the domain through a set of concepts connected by different kinds of relationships (e.g., *is-a* or *is-part-of*) (cf. Figure 4.18b).

With respect to the overlay model, Brusilovsky and Millán [BM07] distinguish between three types:

- *Binary overlay models*: This kind of models represents student's features as a binary value (e.g., known/not known or 0/1).
- *Weighted overlay models*: This kind of models distinguishes several levels of student's knowledge, interests, etc.
  - *Qualitative models*: This kind of models represents student's knowledge, interests, etc. of a concept as a qualitative value (e.g., good, medium, bad).
  - *Simple numeric models*: This kind of models represents student's knowledge, interests, etc. of a concept as a quantitative value (e.g., from 0 to 100).
  - *Uncertainty-based models*: This kind of models uses different forms of uncertainty management to represents student's knowledge, interests, etc.
- *Layered overlay models*: This kind of models stores several values to represent student's knowledge, interests, etc. of each concept.

4.5.3.3 *How the model is initialized and maintained*

In order to ensure that an adaptive system provides adequate individualized support, the UM needs to be initialized and maintained. That means, the UM serves as template and is filled with (initial) values according to the individual student. As a result, UM instances (i.e., user profiles) are created/updated. What features need to be initialized and maintained depends on the characteristic of the features (*static* or *dynamic*). Against the background of *"unobtrusive user modeling"* [SD10], which states that initializing and maintaining user profiles should try to minimize the interference with the main task performed by the student (i.e., solving question and tasks), this means that static features need to be initialized, but *not* maintained and dynamic features need to be initialized *and* maintained.

4.5.4 *Proposed Solution*

The design of the UM is also structured along the three dimensions used to categorize the related work.

4.5.4.1 *What is modeled*

Although many user features were identified in the literature, it is important to select only those features for the use in the UM that have a direct influence on the assessment process. They should be as much as needed in order to generate an accurate adaptation, but not more in order to avoid designing a UM that is unnecessarily complex. That is the reason why each adaptive system only uses a subset of these features depending on the adaptations it is aimed at. The user features to be considered in the proposed UM approach are:

- Knowledge
- Interests
- Goals
- Background
- Demographic information

Due to the fact that assessment aims at identifying, gathering, etc. information about students' knowledge (cf. Section 2.2.1), *knowledge* is one of the most important user features a UM for a personalized e-assessment system must consider. Besides, this student feature is used by about one third of all adaptation techniques [Bru96]. Moreover, the assessment of students' knowledge should consider students' *interests* and preferences because they encourage their intrinsic motivation. Students who are intrinsically motivated choose to engage in tasks solely for the pleasure, interests and satisfaction derived from performing those tasks [LCP11]. Due to the fact that questions and tests can not only be used for grading, but also for self-assessment (cf. Section 2.2.4.2), the consideration of students' *goal* is highly beneficial. For example, a student could define that he/she only wants answering questions that match a specific difficulty level or address a specific concept. For that reason, this feature should be an essential feature of the proposed UM. Students' *background* represents the experience gained outside the system. Although this feature is similar to students' knowledge, it is regarded as a relevant feature that also needs to be considered. Considering student's *demographic information* can not only be beneficial in educational [CCPM04], but also in e-assessment settings. This feature provides basic user information such as gender, age, native language and more complex such as formal education. This information is vital to be aware of, for example, to select the test with the right language if it is available in more than one language.

Due to the fact that only a few success stories on using student's *individual traits* in adaptive systems were reported, this feature will not be represented in the UM. Individual traits are traditionally gained through special designed psychological tests. The effort of these tests is enormous. Furthermore, although the development of pervasive and ubiquitous computing progresses quickly and student's *context* is becoming more and more important in adaptive systems [SB11], however, this feature was not regarded as significant as the other features. Finally, the consideration of student's *emotional state* provides one more feature for possible adaptations. Although a potential for using this information in e-learning settings doubtless exists [WB09], however, less significance was attached to this feature in e-assessment settings. In addition, reliable recognizing students' emotions are still one of the main challenges in affective computing.

Compared to the UMs provided by the AASs described in Section 3.5.3, which mainly rely on users' knowledge, the proposed UM enables providing adaptations more extensively and accurately (cf. Table 4.3). Each feature can be distinguished according to its characteristic (i.e., *static* or *dynamic*) as well as with respect to the way how this information is obtained (i.e., *implicit* or *explicit*) (cf. Table 4.4). A static feature does not change or only within a long period of time. Examples of such features are students' interests, background, demographic information and individual traits. In contrast, a dynamic feature changes from time to time such as students' knowledge. It increases and decreases from test to test or even within the same test because students are always learning and sometimes forgetting. Students' goals is another dynamic feature, if a goal is reached (e.g., by successfully passing a challenging test) another goal could emerge and move into focus. In an adaptive e-assessment system, student's knowledge can be derived by analyzing the answers and interactions given by the students. This information is gathered implicitly. In contrast, the information is gathered explicitly, when they are provided by user inputs. For example, students' knowledge and context can be inferred by observing students' work, whereas their interests, goals, background and demographic information are nearly impossible to infer and thus mostly provided explicitly.

4.5.4.2 *How it is modeled*

In this UM approach, the *overlay user modeling* approach was chosen in order to represent the students' features selected. This allows adaptations on a very detailed level. Moreover, the underlying DM is capable to dynamically and precisely reflect the evolution of students' features over the time. This is especially important for AASs, where the UM needs to be updated after each

	PROPOSED UM	SIETTE	PASS	COSYQTI	IADAPTTEST
Knowledge	**x**	x	x	x	x
Interests	**x**	–	–	(x)	(x)
Goals	**x**	–	–	(x)	(x)
Background	**x**	–	–	–	–
Individual Traits	**–**	–	–	–	–
Context	**–**	–	–	–	–
Demographic Information	**x**	–	–	–	–
Emotional State	**–**	–	–	–	–

Table 4.3: Comparison of the features addressed by the proposed UM with UMs of state-of-the-art AASs

	FEATURE CHARACTERISTIC	INFORMATION ELICITATION
Knowledge	dynamic	implicit
Interests	static	explicit
Goals	dynamic	explicit
Background	static	explicit
Individual Traits	static	explicit
Context	dynamic	implicit
Demographic Information	static	explicit
Emotional State	static	explicit

Table 4.4: Characteristics and information elicitation of students' features

test. In the following, the modeling of each student feature with respect to the DM and overlay model is described in detail.

Knowledge In general, overlay knowledge modeling represents student's knowledge as a subset of the DM. Network models are predestined for modeling students' knowledge because the links between concepts allow deriving knowledge beyond direct observation. For example, if a student answered a question wrong that is based on a specific concept, a link to prerequisite concept can help selecting question that assess relevant basic knowledge. This helps identifying whether the student has just made a mistake or fundamental knowledge is missing. Due to the fact that vector models lack this kind of connections between concepts, they are not appropriate for representing students' knowledge. In order to store students' individual knowledge of each concept, an overlay model has been designed, which stores several values of each concept. Storing more than one value per concept is important because when modeling students' knowledge the information *"a student knows a concept"* is not sufficient at all. It is also important, for example, to know the level and type of knowledge as classified by Anderson and Krathwohl [AKA+01] (cf. Section 2.2.6.1). The layered approach enables storing this kind of information for each concept in the domain. The overlay model not only allows storing several values of each concept, but also introduces a uncertainty management. In particular when modeling students' knowledge, there is often the need to deal with information, which is uncertain. Following the example mentioned above, when a student fails answering a question, does it mean that he/she does not know the underlying concept? This information is uncertain. For that reason, the UM approach has to take that into account. In the literature, different approaches are used to deal with uncertain or imprecise information. They primarily base on theory of Artificial Intelligence (AI) or statistics [ND09]. In this UM approach, the *Bayesian inference* [Jen96], a form of statistical method, is used to collect evidence about students' knowledge. It uses the *Bayes' theorem*, also called as *Bayes' law*, to express the *conditional probability* or *posterior probability* of a hypothesis H after evidence E is observed in terms of the *prior probability* of H and E and the *conditional probability* of E given H:

$$P(H \mid E) = \frac{P(E \mid H) \cdot P(H)}{P(E)} = \frac{P(E \mid H) \cdot P(H)}{P(E \mid H) \cdot P(H) + P(E \mid \overline{H}) \cdot P(\overline{H})}$$

where:

- $P(H)$ is prior probability of the hypothesis H.
- $P(E)$ is prior probability of occurring evidence E.
- $P(\overline{H})$ is the complementary of H.
- $P(E \mid H)$ is the conditional probability of E given H.
- $P(E \mid \overline{H})$ is the conditional probability of E given $\overline{H}$.

- $P(H \mid E)$ is the conditional probability of H given E, also called as posterior probability.

This can be illustrated by means of an example: A question requires students to recall a specific term. A student either *knows* or *not knows* the term. It is assumed that we know the following probabilities:

- $P(H) = 0.2$ specifies the probability that a random student knows this term.
- $P(\overline{H}) = 1 - P(H) = 0.8$ specifies the probability that a random student is not able to recall this term.
- $P(E \mid H) = 0.99$ specifies the probability that a student who knows the term gives the correct answer.
- $P(E \mid \overline{H}) = 0.02$ specifies the probability that a student who does not know the term gives the correct answer (e.g., by guessing).

The Bayes' theorem can be used to calculate the probability that a student knows the term when he/she has given the correct answer. The numeric answer can be obtained by inserting these values into the formula:

$$P(H \mid E) = \frac{0.99 \cdot 0.2}{0.99 \cdot 0.2 + 0.02 \cdot 0.8} = \frac{0,198}{0,198 + 0,016} = 0,925$$

That means, the probability that a student knows the term, given that he/she has answered correctly, is about 93%. In combination with overlay models, *Bayesian networks* could be built. A Bayesian network is a combination of graph theory and Bayesian inference and provides a graphical model in which each node represents a variable, which can be a hypothesis or an evidence, and each edge a probabilistic relationship between two nodes. Advantageous of Bayesian networks is the fact that they allow for *diagnosis* (inferences about possible causes of given certain evidences) and *prediction* (future state of variables given a set of evidence). Although both kinds of reasoning are no doubt useful in the case of user modeling, however, constructing of a Bayesian network requires an enormous effort. Basically, it consists of two steps [BM07]:

1. *Development of the qualitative model*: This step involves the definition of the structural model (i.e., nodes and edges)
2. *Development of the quantitative model*: This step involves the specification of the conditional and prior probability or distributions.

Especially the second step has commonly been cited as one of the main difficulties when building a Bayesian network [BM07]. Often, domain experts are used to get the structure/parameters. But, the more complex the structure of the network, the higher the number of parameters needed. It grows exponentially with the number of parent nodes. For that reason, this UM approach is initially limited to the Bayesian inference method, even though it provides the foundation for building Bayesian networks. An exemplary layered uncertainty-based overlay model is depicted in Figure 4.19. It is based on Anderson and Krathwohl taxonomy and shows student's knowledge related to four concepts. It not only shows that a student knows a specific concept, but also of what level and type of knowledge it is about and what is the probability of the hypothesis (i.e., how certain it is).

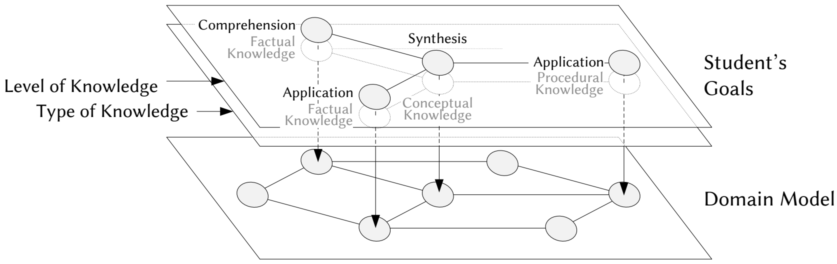


Figure 4.19: Overlay modeling of student's knowledge

Interests Students' interests represent their feeling of wanting to know or learn about something. For that reason, the DM used for knowledge modeling is also used for interests modeling. This is also often done in AHSs such as, for example, the SeAN system [AC01]. The use of a network model allows an accurate representation of interests and in addition enables deriving interests beyond direct observations or inputs. For example, if a student has set that he/she is very interested in a specific concept, it is most likely that he/she is also interested in a concept that is strongly linked to this concept. In order to store the individual interests of each concept, an overlay model has been designed that is based on the weighted overlay model (qualitative). That means it represents students' interests as qualitative values (e.g., none, weak, medium or strong). An exemplary overlay modeling of student's interests is depicted in Figure 4.20.

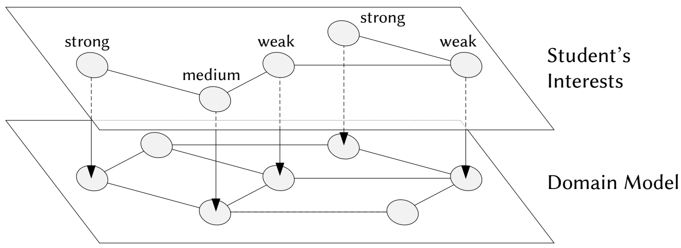


Figure 4.20: Overlay modeling of student's interests

Goals Students' goals represent the purpose for a student's work within the system. In an e-assessment system, students' goals mostly address their knowledge. For example, a student goal could be to provide evidence that he/she can *apply* (i.e., the third level of the cognitive dimension in Anderson and Krathwohl's taxonomy, cf. Section 2.2.6.1) *procedural* (i.e., the third level of the knowledge dimension in Anderson and Krathwohl's taxonomy) knowledge regarding a specific domain concept. Therefore, the DM used for knowledge and interests modeling is used again for goals modeling. That means that students' knowledge, interests and goals are three separate overlays over the same network of concepts. In order to store the individual goals associated to each concept, an overlay model has been designed, which is based on the layered overlay model. That means, it allows storing several values of each concept (e.g., level and type of knowledge). An exemplary overlay modeling of student's goals based on Anderson and Krathwohl taxonomy is depicted in Figure 4.21.

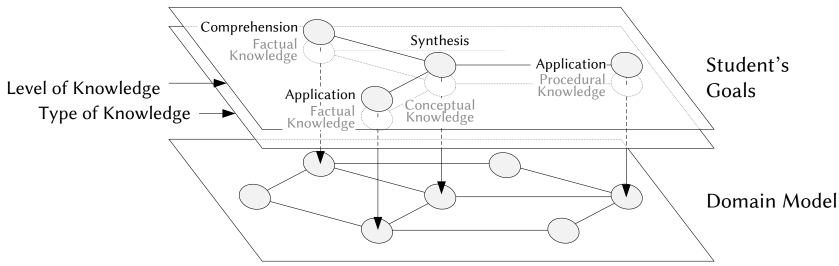


Figure 4.21: Overlay modeling of student's goals

Background

Students' background represents their previous experiences outside the core domain of the system. Although it is similar to students' knowledge, this information is usually static and coarse-grained [SD10]. Moreover, due to the fact that detailed information about the background is not necessary, vector models are best suited as domain structure for this kind of student feature. For this student feature, an overlay model has been designed, which is based on the qualitative weighted overlay model. That means, student's background is represented by set of concepts and an associated qualitative value (e.g., novice, advanced or expert). An exemplary overlay modeling of student's background is depicted in Figure 4.22.

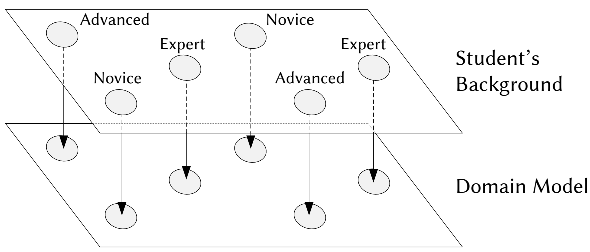


Figure 4.22: Overlay modeling of student's background

Demographic Information

Students' demographic information includes characteristics such as gender, age or native language. As detailed information about these characteristics are necessary, vector models are best suited as domain structure for this kind of student feature. In order to store the individual demographics of each user, an overlay model has been designed, which combines qualitative and simply numeric overlay models. This allows storing both qualitative (e.g., male or female) and quantitative information (e.g., age) and enables more accuracy and flexibility when designing a concrete DM for students' demographic information. An exemplary overlay modeling of student's demographics is depicted in Figure 4.23.

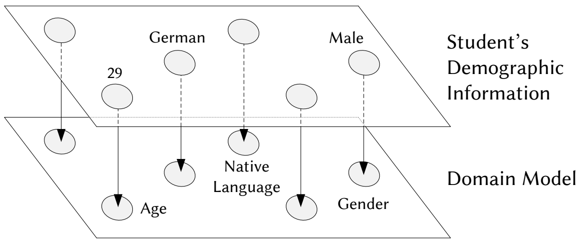


Figure 4.23: Overlay modeling of student's demographic information

As mentioned above, vector models are the simplest form of DMs. They represent the domain through a set of unrelated concepts. The concepts for representing students' demographics are less domain-specific than the concepts of the other features. Thus, they do not need to be defined by a domain expert. However, in order to promote interoperability of the UM, its syntax and semantics should comply with an established standard or specification. Currently, there are mainly two specifications that address the exchange of student data between different systems:

- IMS Learner Information Package (LIP)[14]
- IEEE Public and Private Information (PAPI)[15]

Each of them provides syntax and semantics for characterizing student data, however, this UM approach is based on the categories of IMS LIP for structuring students' demographics. The decision was made in favor of LIP because problems were identified in PAPI in the lack of selectivity between the different categories as well as in the limited expandability. In addition, IMS has shown that PAPI can completely mapped to LIP (cf. Figure 4.24) [SD10]. IMS LIP specifies the following 11 core categories, which can be used to define arbitrary UMs:

- *identification*: This category includes biographic and demographic data.
- *goal*: This category includes information about personal objectives and aspirations.
- *qcl*: This category includes information about qualifications, certifications and licenses granted by recognized authorities.
- *activity*: This category includes information about any learning-related activity in any state of completion.

14 http://www.imsproject.org/profiles/ [last visited: May 21, 2014]
15 http://www.cen-ltso.net/Main.aspx?put=230 [last visited: May 21, 2014]

- *transcript*: This category includes information about academic achievements.
- *interest*: This category includes information about hobbies and recreational activities.
- *competency*: This category includes information about skills, knowledge and abilities acquired.
- *affiliation*: This category includes information about memberships in professional organizations.
- *accessibility*: This category includes information about cognitive, technical and physical preferences.
- *securitykey*: This category includes information about passwords and security keys assigned.
- *relationship*: This category includes information about relationships between the core components.

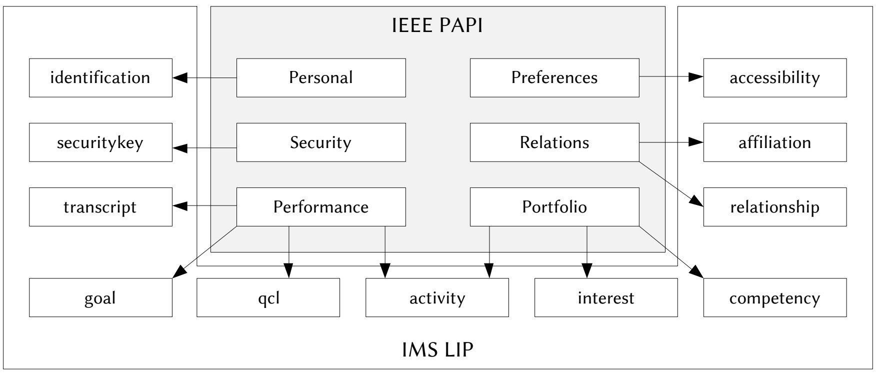


Figure 4.24: Relationship between IMS LIP and IEEE PAPI (adopted from [SD10])

Each of these categories is in turn divided into sub-categories, which accommodate the actual data objects such as *name*, *address*, *country*, etc. as in the example of *identification*. These data objects and structures are used by the UM approach to build up DMs for students' demographic information as needed for implementing overlay user modeling.

An overview of the modeling of each student feature with respect to the DM and overlay model is presented in Table 4.5.

			KNOWLEDGE	INTERESTS	GOALS	BACKGROUND	DEMOGRAPHIC INFORMATION
Domain Model	Vector model		–	–	–	x	x
	Network model		x	x	x	–	–
Overlay Model	Binary		–	–	–	–	–
	Weighted	Qualitative	–	x	–	x	x
		Simple Numeric	–	–	–	–	x
		Uncertainty-based	x	–	–	–	–
	Layered		x	–	x	–	–

Table 4.5: Overview of the modeling of each student feature

	INITIALIZING	MAINTAINING
Knowledge	(x)	x
Interests	x	–
Goals	x	x
Background	x	–
Demographic Information	x	–

Table 4.6: Initializing and maintaining of students' features

4.5.4.3 *How the model is initialized and maintained*

As mentioned above, what features need to be initialized and maintained depend on the characteristics of the features. As shown in Table 4.4, students' *knowledge* and *goals* are dynamic features, whereas *interests*, *background* and *demographic information* are static. Against the background of *"unobtrusive user modeling"* [SD10], Table 4.6 shows which features need to be initialized and which need to be maintained in the proposed UM approach. In the following, it is described how each of these features will be initialized and maintained, respectively.

Knowledge

Initializing students' knowledge in adaptive system can be done in a variety of ways [AC01, TV02, CZQ+05]. Although the approaches clearly differ from each other, they mostly use a pre-test to obtain initial information about this student feature. Then, this information is used as basis, for example, by AHSs to present adequate content. But, in a UM for an AAS, this feature does not necessarily need to be initialized because information about students' knowledge is obtained during student's work with the system. Due to the fact that an AAS is all about questions and tests, a pre-test would be redundant. But, maintaining students' knowledge is all the more important. Maintaining information about students' knowledge can be done based on student's interactions with the system. Each question or task posed to the student has a score to be reached and is linked to at least one concept of a domain. Moreover, each question/task is associated to at least one intellectual skill (thinking skill). Simply put, these skills are needed to fulfill the question/task, The other way around, if a student has successfully mastered this item, it can be assumed that he/she has this skill with a certain probability. This probability is either manually defined or calculated using the Bayesian inference method and ranges from zero (not probable) to one (highly probable). Moreover, these skills are hierarchically structured, that means that lower level skills are needed to master higher levels. Which (intellectual) skill taxonomy (e.g., Bloom's or Anderson and Krathwohl's taxonomy, cf. Section 2.2.6) can be used, strongly depends on the specific application domain. This information combined with the results of the students is then used to infer students' knowledge (level) of domain concepts and to update the user profiles accordingly. In order to compensate varying knowledge levels (e.g., caused by incorrect or unintended interactions), changes should be limited to a maximum and a minimum. That means, the skill level of a student can only rise and fall by a specific value. These boundaries can either be defined globally or skill-specific.

Interests Students' interests are rather static and do not need to be updated regularly, but they need to be initially defined when the student firstly interacts with the system. The initialization of this part of the UM can be done by visually presenting a DM to the students and require them to define their interests by choosing concepts and specifying corresponding qualitative values. Although this procedure is only required once at the beginning, students are able to update their interests anytime. As this feature does not change or only within a long period of time, there is no need to maintain the information about this feature.

Goals Initializing students' goals is similar to initially elicit their interests. Students are required to define their goals by choosing a concept and specifying corresponding values. In contrast to students' interests, there is not only one corresponding value, but several values, for example, one value could represent the level and another value the type of knowledge. Maintaining this part of the UM can be done automatically, when a student has successfully passed a question that matches one of the goals. The system notifies that a goal has been reached and enables defining new goals. Besides, students are able to update their goals anytime.

Background As this feature does not change or only within a long period of time, there is no need to maintain the information about this feature. However, initializing students' background is all the more important. This can be done by explicitly asking the student to define information about his/her background. The background is defined as a set of unrelated concepts and corresponding values. In contrast to students' knowledge, the mastery of each concept is only represented as qualitative value.

Demographic Information Initializing students' demographic information is similar to define their background. It can be done by asking the student to input its gender, age or native language. Depending on the type of the demographic, either a qualitative or quantitative value is associated to the corresponding information. Although this procedure is only required once at the beginning, students are able to update their demographic information anytime. As this feature does not change or only within a long period of time, there is no need to maintain the information about this feature.

4.5.4.4 *Open User Modeling*

In order to avoid intractability of user modeling and to provoke students to reflect on their results, the UM should be made open to the students. That means that the system should provide comprehensive information about students' features and their current values. According to Sosnovsky and Dicheva [SD10], a concept-based DM, as used in this UM approach, is best suited for providing an open and editable user modeling. The level of students' control is not only limited to *scrutinizing*, but also allows *modifying* the information. In contrast to students' interests, goals, background and demographic information, their knowledge can only be scrutinized and not modified directly. In other words, when a student beliefs that his/her actual knowledge (level) does not correspond to the measured knowledge, he/she can change his/her goals accordingly. Then, the next questions posed to the student take these new goals into account. This insight into the different features and their actual values can encourage students' reflection of their learning [BM03].

4.5.5 *Summary*

This section has presented the design of the UM. The UM is the basis of the user modeling component, which is an important pillar of the overall architectural approach. The user features, which are modeled, are *knowledge*, *interests*, *goals*, *background* and *demographic information*. Moreover, the overlay user modeling approach was chosen in order to represent the features selected. While background and demographic information were represented using a set of independent and unrelated concepts (vector models), knowledge, interests and goals were modeled by network models, which represent the domain through a set of concepts connected by different kinds of relationships. In addition, the UM approach allows dealing with uncertain or imprecise information by the use of the Bayesian inference method. Furthermore, in order to avoid intractability of user modeling and to provoke students to reflect on their results, the UM approach allows students to scrutinize/modify their individual user profiles. Finally, the UM not only provides an ideal framework for representing student's features using overlay modeling, but also allows deriving useful assumptions about true knowledge as well as strengths and weaknesses of the students.

IN A NUTSHELL: The UM approach

- Enables representing a variety of user features ranging from user's knowledge to its individual characteristics (e.g., interests, goals or background)
- Supports complex assumptions and reasoning based on information about the users
- Allows dealing with uncertain and imprecise information
- Ensures interoperability of the model and its information through the use of the IMS LIP specification

4.6 ADAPTATION MODEL

4.6.1 *Introduction*

In general, the AM can be considered as the core of an adaptive system because it defines *what* can be adapted, as well as *when* and *how* it is to be adapted. In this section, the design of the AM is presented, which is administered by the adaptation modeling component. Firstly, Section 4.6.2 presents the requirements regarding the design of the AM. Then, Section 4.6.3 briefly looks into related work in the field of adaptation methods, before the own AM approach is proposed in Section 4.6.4. Parts of this section were already published in [SW13a, SW13b].

4.6.2 *Requirements*

Tests in traditional e-assessment systems and tools are characterized by a static structure, which means they have an ordered sequence of questions and a predefined start and ending. They are self-paced by the students and provide immediate feedback, both positive and negative. This linear procedure corresponds to Skinner's approach to education called *programmed instruction* [Ski54]. This approach does not consider students individual knowledge or characteristics. Instead, each student has to answer the same questions. However, the aim of this overall approach for an adaptive e-assessment system is not only to identify, but also to support and even compensate deficits in students' individual learning by considering students' strengths or preferences. This requires that students' responses should determine what question is presented next. This corresponds to Crowder's approach of learning [Cro60]. He states that instructions should be branched so that some students can be presented with additional information if they do not respond well enough and that more advanced students can be exposed to more challenging content. For that reason, the AM approach should be able to adapt the sequence, selection and presentation of questions to the individual student. Feedback plays an important role in the assessment process and can be regarded as the so called speaking tube of the question and test evaluation and thus able to communicate the result of the assessment to the students as well as other information, which may contain reasons for incorrect answers, hints or advices for continuing the assessment. Due to the fact that there is no *one-size-fits-all* approach of providing appropriate feedback, feedback also needs to be adapted to the individual student and/or its responses. As shown in earlier research [SW12, SW11a], feedback personalization is still neglected or even not addressed in state-of-the-art AASs. For that reason, the AM approach should also be able to adapt the selection and presentation of feedback to the individual student. This allows providing guidance and support when required and in turn maximizes the motivation of the students [Con05]. The information on which these adaptations are made are not only obtained from students' performances in the tests, but also from their individual features (e.g., prior knowledge, background or interests) stored in the UM. The requirements for the design of the adaptation modeling approach are summarized as follows:

- *Individualization*: It should be capable of adapting the sequence, selection and presentation of questions to the individual student.
- *Assistance*: It should be capable of adapting the provision of feedback given during the assessment.
- *Flexibility*: It should be capable of exploiting both students' performances as well as their individual features for adaptations.

4.6.3 *Related Work*

Over the last few years, a lot of research has been done to adapt learning content to individual students and groups of students (cf. Section 3.4.2). In order to analyze and compare the variety of adaptive methods provided by the different systems and approaches, Specht [Spe98] compiled a simple taxonomy scheme to meet this challenge. It separates adaptive methods of educational hypermedia applications into four dimensions:

- *What does the system adapt?* (adaptation means)
- *To what does the system adapt?* (adaptation information)
- *Why does the system adapt?* (adaptation goals)
- *How does the system adapt?* (adaptation process)

In the following, this scheme is used to structure the related work in the field of adaptation modeling.

4.6.3.1 *Adaptation Means*

This dimension focuses on the elements of an adaptive system that are adapted by an adaptive method. In the field of educational hypermedia, Leutner [Leu92] differentiates between three elementary adaptation means, which can be used to adapt learning and teaching to students:

- *Teaching goals*: This adaptation mean can be adapted, for example, by varying the amount of instructions or by selecting curricula of varying difficulty.
- *Teaching methods*: This adaptation mean can be adapted, for example, by composing curricula from different learning objects.
- *Teaching times*: This adaptation mean can be adapted, for example, by shortening the time available to learn when a student has not answered a question correctly.

However, adaptation means strongly depend on the type of the adaptive system. In e-assessment settings, adaptation means mostly focus on questions and tests and their attributes (e.g., number of choices, score, or feedback). In addition to adapt learning and teaching, Specht also mentioned adaptation means with respect to the User Interface (UI) such as the functionality, amount, position and presentation of objects (e.g., color, size and degree of details) or the interface language.

4.6.3.2 *Adaptation Information*

This dimension focuses on the individual characteristics of a student as well as all situation-related characteristics that are used to provide adequate adaptations. In most adaptive educational hypermedia systems, a UM is the basis for the adaptation. Nevertheless, there are several other sources of information, which are used for adaptations. This includes student's current tasks and location as well as information, which are obtained during student' interaction with the system.

4.6.3.3 *Adaptation Goals*

This dimension focuses on the pedagogical reasons behind the adaptations. Salomon [Sal75] differentiates between three heuristical models:

- *Assistance model*: The aim of this model is to support students' learning deficits by additional learning/teaching activities.
- *Compensation model*: The aim of this model is to compensate deficits in students' learning premises (e.g., motivation) by appropriate activities.
- *Preference model*: The aim of this model is to use students' strengths to compensate deficits not diagnosed or hard to diagnose.

In general, adaptations in educational hypermedia settings mainly aim at compensating knowledge deficits, ergonomic reasons or adapt to learning styles for an easier introduction into a topic [SB07].

4.6.3.4 *Adaptation Process*

This dimension focuses on the actual adaptation process and the preceding acquisition of adaptation information. The choice of method to collect adaptation information strongly depends on features to be captured. While user features such as the user knowledge can be obtained from tests or questionnaires, other features such as learning styles or interests are much more difficult to capture (e.g., by monitoring and analyzing students' interactions with the system). Having this information, the actual adaptations can be performed. Dieterich et al. [DMKSH93] have subdivided the adaptation process into four steps. They distinguish between:

1. *Initiative*: In this step, the initiative to perform an adaptation is taken.

2. *Proposal*: In this step, different alternatives for adaptations are proposed.

3. *Decision*: In this step, at least one alternative is chosen.

4. *Execution*: In this step, the adaptation is performed.

Depending on how active or reactive the user and the system are in the adaptation, Dieterich et al. differentiate between six types of adaptive systems ranging from *self-adaptation* to *adaptation* (cf. Figure 4.25).

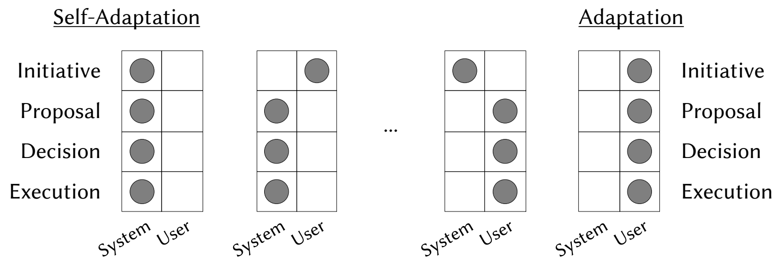


Figure 4.25: Different types of adaptive systems (adapted from [DMKSH93])

For proposing and aiding decisions about alternatives, adaptive systems often make use of techniques from the field of *expert systems* [Jac93]. Typically, an expert system consists of a *knowledge base* and an *inference engine*. The knowledge base expresses the knowledge to be exploited by the expert system and the inference engine is designed to reason about this knowledge. Both, for representing knowledge as well as for drawing conclusions from the knowledge, different approaches can be used [Lug01]:

- *Rule-based reasoning*: In this approach, the knowledge (base) is expressed by a set of *IF-THEN* (inference) rules. The inference engine searches the knowledge base for rules that match certain facts. There are two methods of reasoning when using inference rules:
  - *Forward chaining*: This method, also known as *data-driven reasoning*, searches for rules, where the antecedent is known to be true. If successful, the consequent is concluded and added as new information to the data. This procedure is repeated until a certain goal is reached.
  - *Backward chaining*: This method, also known as *goal-driven reasoning*, searches for rules, where the consequent matches a desired goal. If the antecedent of that rule is not known to be true, then it is added to the list of goals. This procedure is repeated until a certain conclusion is reached.

- *Case-based reasoning*: In this approach, the knowledge (base) is expressed by a set of experiences (*cases*). The inference engine searches the knowledge base for a previous situation similar to the current one and uses that to solve the new problem. Generally, it is a four-step process:

1. *Retrieve*: In this step, the case or cases are retrieved whose problem is most similar to the new problem.
2. *Reuse*: In this step, the solutions from the retrieved cases are reused to create a solution for the new problem.
3. *Revise*: In this step, the solution is revised in order to take account of the differences between the new problem and the problems in the retrieved case(s).
4. *Retain*: In this step, the new problem and its revised solution are retained as a new case for the knowledge base.

Each of these approaches has its advantages and its disadvantages [PH07]. The establishment of a complete knowledge base in rule-based reasoning is extremely difficult and complex. The more powerful and demanding the system should be, the more inference rules need to be defined. In cases information is lacking, contradictions could occur. However, if the knowledge base is precisely defined, the results can be expected to be correct and complete. Although the majority of knowledge can explicitly be expressed using rules, the combination of rules additionally allows processing complex information. The addition of rules during the operation does not pose any problems to the rule-based reasoning approach, however, it must be done with care to avoid introducing contradictions.

The strength of the case-based reasoning approach is apparent in the *retain* step. It allows the system to learn independently and continuously so that knowledge can be gained during the operation. Moreover, it allows integrating exception cases (specialized knowledge) more easily than in the rule-based approach. However, the case-based reasoning approach also shows some serious shortcomings. An incompleteness of the reference cases can result in selecting an inappropriate solution and in performing an incorrect adaptation. Furthermore, if certain cases never occur, they will also not be added to the knowledge base. In order to ensure that the system is working well, an adequate number of reference cases need to be available. However, if the knowledge base contains too many cases, the search process may become inefficient. In addition, in order to find a solution for the new problem, an adequate procedure is necessary to determine similar cases. The evaluation of similarity is an interpretative process and difficult to implement using computers (*matching problem*) [BKI03]. The characteristics of each case need to be described by a suitable vocabulary and compared using a reliable similarity measure.

4.6.4 *Proposed Solution*

In meeting the requirements stated above, an AM consisting of several adaptation methods has been designed. The structure used to categorize the related work is also used to present the model and its methods.

4.6.4.1 *Adaptation Means*

The elements, which are adapted by the adaptive methods in this AM are:

- Question sequence
- Question selection
- Feedback selection
- Question presentation
- Feedback presentation
- Question difficulty

Question Sequence

The first adaptation mean encompasses the order of the questions presented to the students. The idea of branching the sequence of questions in the design of the AM was motivated by Crowder's approach of learning [Cro60]. For adapting the sequence of questions, the AM allows to:

- Retry questions
- Retry tests
- Move to certain questions
- Branch tests

Using these control structures, a variety of question sequences can be defined (cf. Figure 4.26). Ranging from simple sequences without any control structures (1) to complex sequences using branches, returns and loops (5). While the first sequence conforms to Skinner, the last sequence corresponds to Crowder. The second sequence (2) allows repeating some questions. Additional hints could be provided after an incorrect try that helps students in finding the correct solution. This corresponds to Salomon's assistance model. The third sequence (3) allows skipping questions, for example, for more advanced students. This corresponds to Salomon's preference model and the fourth sequence (4) allows compensating deficits by using alternative question paths. This corresponds to Salomon's compensation model.

Question Selection

The second adaptation mean describes the composition of the tests. The AM allows dynamically generating a test consisting of questions, which meet specific requirements. Such requirements could be (among other):

- The question is referred to a specific domain or concepts of the DM (e.g., biology)
- The question is referred to a specific difficulty taxonomy or level (e.g., synthesis level in Bloom's taxonomy)
- The question has a specific interaction type (e.g., multiple-choice)
- The question has not yet presented to the respective student

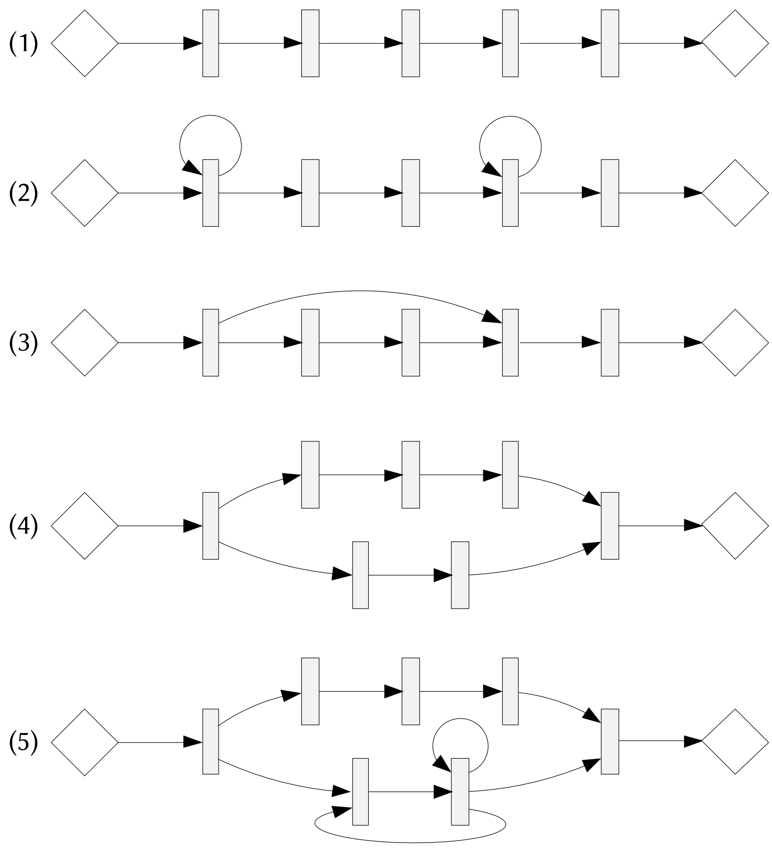


Figure 4.26: Different types of sequences (adapted from [DMKSH93])

Feedback Selection

The third adaptation mean is similar to the previous one in selecting specific feedback as response to a given answer. In order to allow selecting different feedback, the feedback created needs to be classified according to a uniform scheme. In this system, the classification developed in Section 4.3.4.3 is used. It allows selecting feedback that is most suitable for the moment and the respective student. Example conditions for selecting feedback could be (among other):

- The feedback is enriched with elaborative information (e.g., response-contingent feedback)
- The feedback is enriched with graphics or animations
- The feedback is limited to textual information
- The feedback is given immediately after the student has given the answer

Question Presentation

The fourth adaptation mean is related to the presentation of the questions. This is realized by choosing a specific interaction type if one and the same question is created using different interaction types and by hiding or showing information of the question text and its associated media.

Feedback Presentation

The fifth adaptation mean is similar to the previous one in adapting the presentation of feedback. Having classified the feedback using the scheme mentioned above, the *presentation* dimension is used to identify feedback that is best suited for the respective student.

Question Difficulty

The sixth adaptation mean is related to the difficulty of the questions. The difficulty of the questions is varied by the amount of additional information (hints), which is given during the assessment and the time available to answer the question.

4.6.4.2 *Adaptation Information*

The information that can be used to provide adaptations in this AM can roughly be divided into:

- Student performance
- Student features

Student performance encompasses the information, which are obtained while a student takes a test. It can be accessed at any point of the test. This includes the amount of question correctly and incorrectly answered, the scores achieved as well as the time needed to answer the question(s). On the contrary, student features encompass all information, which are stored in the UM (cf. Section 4.5). This includes students' knowledge, which is one of the most important user features a personalized e-assessment system must consider as well as interests, goals and background. Depending on the student feature, qualitative (e.g., good, medium, bad) or numeric values (e.g., from 0 to 10) are presented for selection. As the UM can be extended as required, the adaptation information is also not limited to this set of information.

4.6.4.3 *Adaptation Goals*

The main reasons behind the adaptations in this AM are:

- Identification of students' strengths and weaknesses more valid and reliable
- Consideration of students' strengths and preferences to compensate weaknesses and deficits
- Encouragement of students' motivation by considering individual aspects (e.g., interests and goals)

4.6.4.4 *Adaptation Process*

As mentioned above, the adaptation process starts with the acquisition of adaptation information. In this AM, the adaptation information is students' performance in the test as well as the current values of students' features. Consequently, the former kind of information is directly obtained from the adaptive testing engine during the assessment process. In contrast, the latter kind of information is obtained from the user modeling component. When creating the adaptive test, this information is referenced inside the UM. At runtime, the adaptive testing engine uses the user profiles to fill the references with concrete values. Based on the results of the test, the user profiles are updated accordingly (cf. Section 4.5.4.3). With regard to the classification of adaptive systems provided by Dieterich et al., the system performs all four steps itself. Consequently, it can be referred to as *self-adaptive*. However, the initiative to perform an adaptation can only be taken at specific points during the testing procedure. These so-called *trigger points* are:

1. Before presenting the question
2. During the interaction with the question
3. After answering the question

For proposing and aiding decisions about alternatives, the AM uses the *rule-based reasoning* approach. The main reason for selecting this approach was the ability of the author to directly and easily influence the adaptation process by creating rules. Rules are a very natural knowledge representation method, with a high level of comprehensibility because they look like natural language expressions (e.g., in such a situation, do this and this). The effort in building the initial knowledge base is almost the same in both approaches. In order to achieve an optimal result, the rule-based reasoning approach requires a comprehensive set of rules, whereas the case-based reasoning approach requires a sufficient number of reference cases. Nevertheless, by the use of rules, the author is able to directly influence and optimize students' testing procedure. In contrast, the case-based reasoning approach independently selects a next question based on a reference case. Although the author has defined or at least accepted these cases, it is not guaranteed that the system adapts in the interests of the author. In contrast, a rule-based reasoning system always provides an explanation for the derived conclusions in a straightforward manner. In general, a rule consists of two parts, an IF part called *antecedent*, *premise* or *condition* and an THEN part called *consequent*, *conclusion* or *action*. The IF part relates given information or facts to some action in the THEN part. A rule can have multiple antecedents joined by conjunctions (AND), disjunctions (OR) or a combination of both. However, it is recommended to avoid mixing conjunctions and disjunctions in the same rule. The antecedent of a rule in turn consists of two parts, an *object* and its *value*. They are linked by an *operator*. This AM uses not only symbolic (e.g., *is* or *is not*), but also mathematical operators (e.g., $\leqslant$, $=$ or $\geqslant$) to compare an object with a value. For example:

IF *"answer_1" is wrong* THEN *"retry_question"*
IF *"score_1"* $\leqslant$ *5* THEN *"move_to_question(2)"*

The general syntax of an adaptation rule is depicted in Figure 4.27.

According to Durkin [Dur94], rules can represent different types of semantics namely *relations*, *recommendations*, *directives*, *strategies* and *heuristics*. In

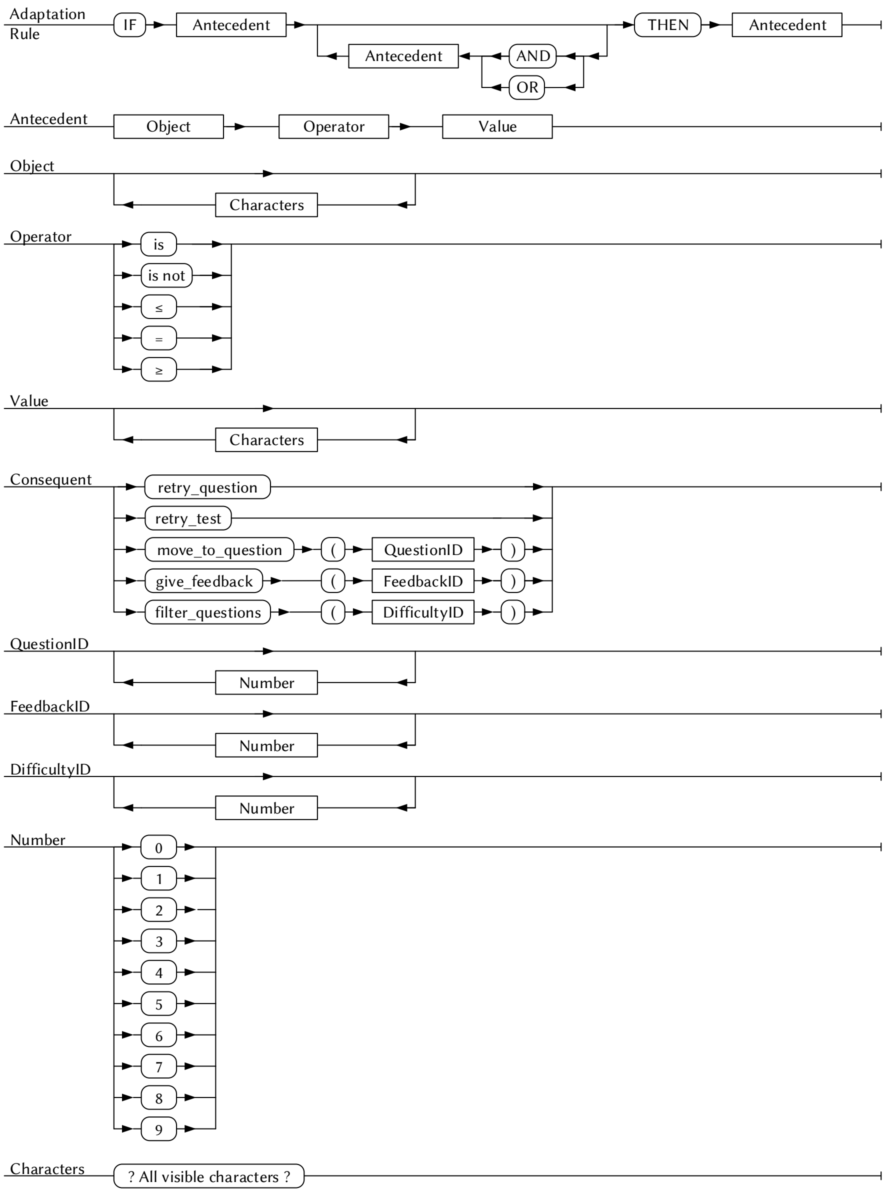


Figure 4.27: Syntax diagram of an adaptation rule

this AM, rules are mainly used to specify *strategies*. This means that when the antecedent of a rule has been satisfied (*matched*), a set of actions will be triggered (*fired*) in order to achieve a specific goal (cf. Figure 4.28).

Rule-based reasoning systems also distinguish between the ways in which rules are executed (i.e., *forward* and *backward chaining*). In this AM, forward chaining is used as inference method because the adaptive system firstly gathers (adaptation) information and after that it tries to infer new facts from it. In contrast, backward chaining begins with a hypothetical solution and then attempts to find facts to prove it. Due to the fact that the data determines which rules are selected, this method is also called as *data-driven*. In order to avoid conflicts between rules when more than one rule can fire at the same time, but have inconsistent consequences, a *conflict resolution* is needed.

In this AM, the rule with the highest priority will be fired. The priority is established by placing the rules in an appropriate order in the knowledge base. Consequently, the adaptation process basically consists of four steps:

1. *Matching*: Find all rules whose antecedents are true and mark them as being applicable.
2. *Conflict Resolution*: If more than one rule can fire, select the rule with the highest priority.
3. *Action*: Execute the consequent of the lowest numbered applicable rule. If none applies then stop.
4. *Reset*: Reset the applicability of all rules and return to step (1).

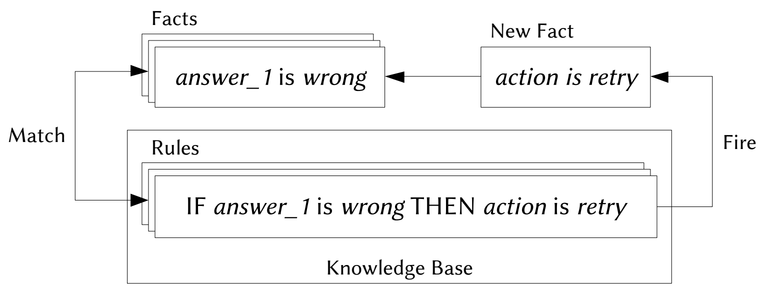


Figure 4.28: Match-fire procedure (adapted from [PH07])

Finally, the complete structure of the rule-based reasoning system is shown in Figure 4.29. The structure clearly shows the basic structure of a rule-based expert system consisting of a *knowledge base*, *database* (facts), *inference engine* and *explanation facilities* [Neg04] and its mapping to the components of the proposed adaptive system architecture (cf. Section 4.2.3). The knowledge base is represented as a set of rules. In this AM, the rules are associated with questions and compiled to *adaptive tests*. The compilation of the tests as well as management of the rules (i.e., creating, modifying, etc.) is realized by the adaptation modeling component. Then, the tests are handed over to the adaptive testing engine, which represents the inference engine and carries out the reasoning. The facts needed for matching the rules are provided both by the engine itself (student performance) and the user modeling component (student features). A crucial element of each expert system is the explanation facility, which provides information how a particular conclusion is reached and why a specific fact is needed. In addition to get general statistics about the tests taken (e.g., reached scores, correct and incorrect answers or the domains or concepts covered), it gives students an opportunity to get informed about which and more importantly, why adaptation decisions were carried out.

In relation to the two types of adaptive techniques that are applied in AASs namely *adaptive testing* and *adaptive questions* (cf. Section 3.5.2), this AM corresponds to the latter one. The adaptive question technique is based on rules, which allow dynamically selecting appropriate questions at run-time.

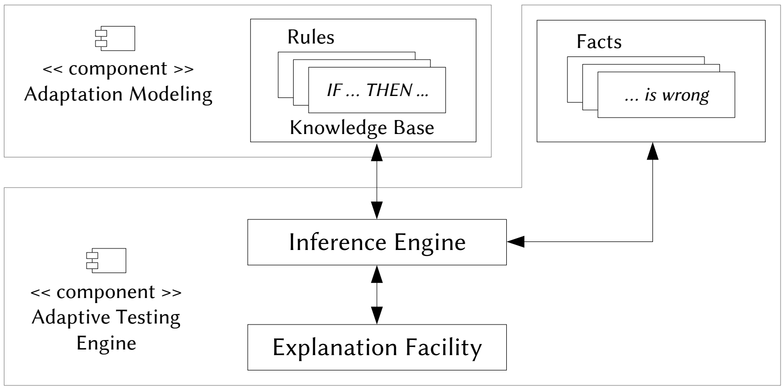


Figure 4.29: Structural overview of the rule-based reasoning system

For that reason, this AM lays the foundations for adaptive e-assessment system following the adaptive question technique.

4.6.4.5 *Modeling of Adaptivity*

Even though the AM provides a powerful instrument for tailoring tests to students' features, the use of it to create complex adaptive tests could be challenging for some authors. The most authoring tools in adaptive systems are too fine grained so that authors easily loose the overview [CC03]. In order to prevent this, a novel adaptation modeling approach has been developed. It supports the authors in modeling the dynamic behavior of adaptive tests and helps them to cope with the inherent complexity. The approach is based on Finite State Machines (FSMs), a mathematical model of behavior from the field of *automata theory* [HMU06]. FSMs are one of the most widely used models in computer programming and also used in a variety of other areas ranging from electrical engineering, mathematics to linguistics.

In general, the FSM is an abstract machine that has a finite number of *states* and state *transitions*. A state represents any possible status of a system or object that is waiting to execute a transition. A transition is a state change triggered by a condition. It is also possible to associate *actions* with a state, for example, when entering or exiting. FSMs can be represented by a directed graph, which is called a *state diagram* (cf. Figure 4.30). Each state is represented by a node (circle) and the edges show the transitions from one state to another. Each arrow is labeled with the condition that triggers that transition. The start state is shown with an arrow pointing at it from anywhere and the final states are represented by a double circle.

Analyzing the characteristics of adaptive tests, it shows that the containing questions can also be regarded as objects with dynamic behavior. That means, depending on the response of the student, different actions will be triggered. This could be the provision of feedback or the selection of other questions. Consequently, the answer options of a question have been modeled as states and the opportunities to switch between these options has been expressed as transitions. In other words, changing the answer selected within the question

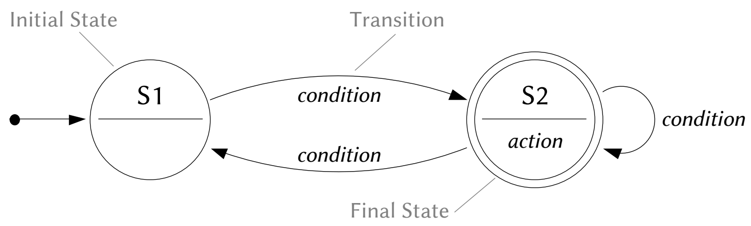


Figure 4.30: Finite state machine

results in a state transition. Having this graphical representation, the author of the adaptive test has a quick overview about any possible status of a question and is able to respond directly by settings trigger points for adaptations. Trigger points cannot only be defined at states, but also at transitions. This not only allows initiating adaptations at any point during the question/test, but also to track students' answer path and to respond accordingly. As an example, Figure 4.31 shows a FSM model of a multiple-choice question with three answer options (*A*, *B* and *C*). Assumed that *A* is correct, the following scenarios could be realized:

- A student switches between *A* and *B* several times. By counting the number of transitions between both states, his/her uncertainty can be recognized and supported by giving hints (feedback).
- A student firstly selects *A* and retains the selection for a long time, but in the end submits *C*. Then, the student could get feedback to rethink the answer. In contrast, if the student switches from *C* to *B*, no feedback will be presented.
- A student switches between *B* and *C*, and submits *B*. It could be derived that *A* was not an option and that the question was too difficult for the student. The following questions could have a decreased difficulty.

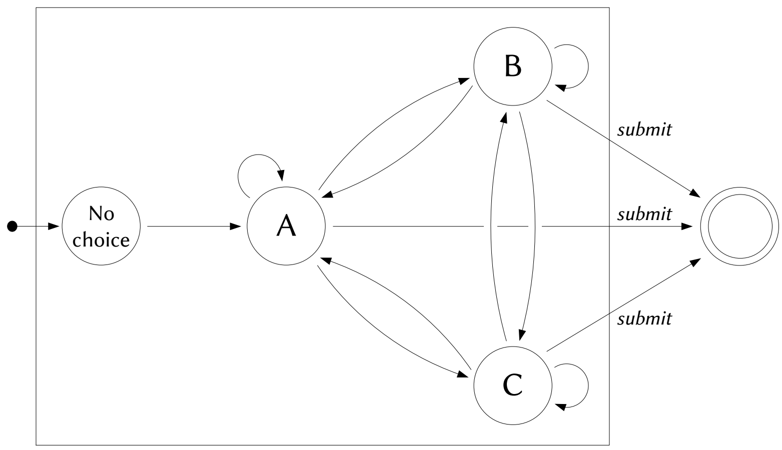


Figure 4.31: FSM state diagram of a multiple-choice question

Summarized it can be stated that using FSMs opens up new possibilities for adaptation modeling. They graphically support the authors in defining trigger points for adaptations at any point during the question/test. In addition, students' behavior while answering the question could be tracked and used to determine students' strengths and weaknesses more precisely. However, with a growing number of answer options, the number of states and state transitions is increasing as well and the state diagram could quickly become confusing. Due to the fact that this strongly depends on the question type (e.g., given n answer options, a multiple-choice question results in $n+1$ states and n^2 transitions, whereas a multiple-answer question result in 2^n states and $2n \cdot 2^{n-1}$ transitions), the decision to use this graphical support must be decided on a case-by-case basis.

4.6.4.6 *Standard-compliant Implementation of Adaptive Tests*

As presented in Section 4.3.4, the IMS QTI specification was selected as basis on which the own QM has been built on. This enables assessment content reuse across different systems and tools. However, having standard-compliant questions on the one hand and adaptation rules specified in an own format on the other hand, would break this principle. As a consequence, this would make it necessary to update any other testing engine that would like to render and process these adaptive tests. The most testing engine provider would not consider the added effort as reasonable. But, when using the full potential of IMS QTI, adaptive tests can be implemented fully compliant with this specification. This can be made possible by the use of QTI's *response* and *outcome processing* as well as *pre-conditions* and *branching rules*. Section 3.3.4.3 describes these data structures in detail.

As mentioned above, the adaptive tests are based on adaptation rules. Depending on the adaptation mean to be addressed (cf. Section 4.6.4.1), the rules are either implemented on question or on test level. For example, the selection and presentation of feedback is done within an individual question and could be realized using the response processing. The response processing of IMS QTI evaluates student's responses and sets outcome variables (prior defined in response declaration sections) accordingly. Then, the values of these variables trigger the selection and presentation of adequate feedback.

In contrast, the sequence, selection and presentation of questions is done during a test. Consequently, adaptation rules addressing these adaptation means could be realized using pre-conditions, branching rules and the outcome processing. Pre-conditions and branching rules are new features, which were firstly specified in the IMS QTI specification v2.1. While pre-conditions will be evaluated just before the student enters a test part (a test part consists of at least one question), branching rules determine where next after a test part has been completed by a student. These elements allow skipping and retrying of questions, move to certain questions, etc. based on variables set by the outcome processing. A complete adaptive test according to the IMS QTI specification can be found in Appendix B.

4.6.5 *Summary*

This section has presented the design of the AM. It can be regarded as the core of an adaptive system because it defines what can be adapted, as well as when and how it has to be adapted. The model and its methods were presented using Specht's taxonomy scheme consisting of four dimensions. The first dimension (*adaptation means*) focuses on the elements, which are adapted by the adaptive methods. In this AM, the adaptation means are the *question sequence*, *question selection*, *feedback selection*, *question presentation*, *feedback presentation* and the *question difficulty*. The second dimension (*adaptation information*) focuses on the information that can be used to provide adaptations. The information used by this AM are *student's performance* and *student's features*. The third dimension (*adaptation goals*) focuses on the pedagogical reasons behind the adaptations. The main reasons behind the adaptations in this AM are the identification of students' strengths and weaknesses more valid and reliable, the consideration of students' strengths and preferences to compensate weaknesses and deficits and the encouragement of students' motivation by considering individual aspects (e.g., interests and goals). The fourth dimension (*adaptation process*) focuses on the actual adaptation process and the preceding acquisition of adaptation information. In this AM, the adaptation information is directly obtained from the adaptive testing engine during the assessment process and the user modeling component, respectively. For proposing and aiding decisions about alternatives, the AM uses the *rule-based reasoning* approach. More precisely, *forward chaining* was used as inference method. Thus, the AM corresponds to the *adaptive questions* technique. In addition, an adaptation modeling approach has been provided that supports authors in modeling the dynamic behavior of adaptive tests and helps them to cope with the inherent complexity. Finally, a standard-compliant implementation of adaptive tests was proposed.

IN A NUTSHELL: The AM approach

- Enables adaptive e-assessments based on a variety of adaptive methods
- Allows identifying students' strengths and weaknesses more valid and reliable
- Allows considering students' strengths and preferences to compensate weaknesses and deficits
- Encourages students' motivation by considering individual aspects
- Supports flexible adaptation strategies using rule-based reasoning and forward chaining

4.7 ARCHITECTURAL COMPONENTS

4.7.1 *Introduction*

While the previous sections explain the underlying models of the architectural approach, this section gives an insight into the design considerations of the different architectural components. First of all, Section 4.7.2 describes the functional and non-functional requirements that imposed constraints on the design and implementation of the architectural components. Then, Section 4.7.3 and 4.7.4 present the design of two components of the system in detail, structured according to the overall system design pattern (cf. Section 4.2.4).

4.7.2 *Requirements*

In order to derive the functional requirements of the different components, three application scenarios were created. They illustrate how the different components are envisaged to be used by the main actors (i.e., administrator, author and student, cf. Section 2.2.2). All application scenarios can be found in Appendix C. Based on these descriptions, a set of requirements were derived, which define the functions of the components from the users point of view. Table 4.7 shows exemplary all functional requirements related to the adaptation modeling component. In addition, assumptions were made about the importance of the different requirements. The smaller the value, the higher the priority of the described requirement and its contribution to the overall implementation of the system, respectively. The functional requirements of the other components can be found in Appendix D.

The functional requirements serve as the fundamental basis for the implementation of the different components. Simply put, they specify the tasks that must be accomplished. However, these tasks must also be implemented in a way that allows users accomplishing them efficiently and effectively. In other words, the interfaces for the users have to have a high *usability*. This results in increased user acceptance and satisfaction. Satisfying this non-functional requirement is addressed by the field of *usability engineering* [Nie93]. Usability engineering recommends the application of different methods at different steps within the software development process. For example, performing *contextual inquiry* or *focus groups* when analyzing requirements, using *use case* diagrams or UI *mock-ups* when designing or considering usability *guidelines* when implementing a specific system [RF07]. In this context, usability mainly concerns the dialogs between a user and the system. For that reason, different parts of the ISO 9241 standard covering ergonomics of human-computer interaction can be used to derive requirements and recommendations:

- Part 14: Guidance on menu dialogs
- Part 16: Guidance on direct manipulation dialogs
- Part 110: General dialog principles

ID	REQUIREMENT	PRIORITY
AM-1	Authors are able to create, edit and delete adaptive tests. An adaptive test is a group of questions sequenced by adaptation rules.	1
AM-2	Authors are able to create, edit and delete adaptation rules. An adaptation rule consists of at least one condition and an action.	1
AM-3	Authors are able to set trigger points (i.e., the times when a rule will be checked).	1
AM-4	Authors are able to define conditions using students' performance in the test as well as their individual profiles.	1
AM-5	The system allows linking more than one condition using 'and'/'or'.	3
AM-6	Authors are able to define actions to be executed.	1
AM-7	Authors are able to assign adaptive tests to students or groups of students.	1

Table 4.7: Functional requirements for the adaptation modeling component

- Part 143: Guidance on form-filling dialogs
- Part 151: Guidance on world wide web user interfaces

More details about these parts and how their consideration contribute to the usability can be found in Appendix D.

4.7.3 *Adaptation Modeling Component*

4.7.3.1 *Introduction*

This section focuses on the design of adaptation modeling component. Figure 4.32 depicts the component based on the structure presented in Section 4.2.4. In the following, the different layers are described in detail.

4.7.3.2 *Data Layer*

In general, the data layer consists of a set of models, which abstract the data and provide access for reading and manipulating the data. With respect to the adaptation modeling component, the data layer comprises the following models:

- *Test*: This model represents the tests managed by the component.
- *Part*: This model represents groups of sections.
- *Section*: This model represents groups of item references.

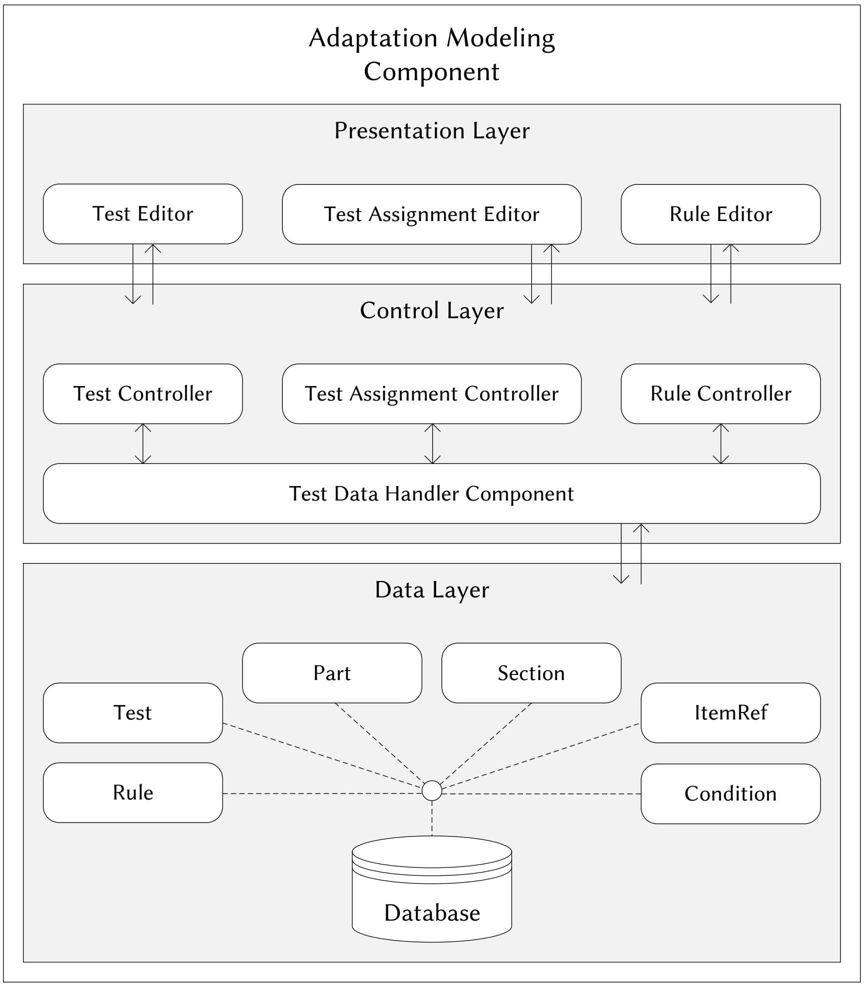


Figure 4.32: Architectural structure of the adaptation modeling component

- *ItemRef*: This model represents the references of questions or tasks within a test. They can be regarded as question/task representatives.
- *Rule*: This model represents the rules managed by the component. Due to the fact that a rule can only define one action, the action to be executed is also represented by this model.
- *Condition*: This model represents the conditions under which a rule will be triggered. A rule can have one or more conditions.

It is important to note that the data structure of the *Part*, *Section* and *ItemRef* model corresponds to IMS QTI v2.1 specification to support the import and export of tests. Furthermore, the different models are also linked to each other using different associations (cf. Figure 4.33). Each (adaptive) test consists of at least one part, which in turn consists of one or more sections. Such a section consists of at least one question (reference). Furthermore, each question can be linked with a rule, which in turn consists of one or more conditions. Finally, tests can be assigned to different users.

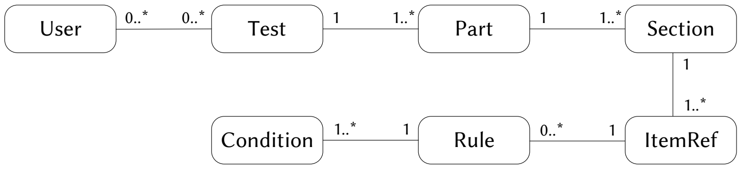


Figure 4.33: Relations between the adaptation modeling component models

4.7.3.3 *Control Layer*

The control layer of the adaptation modeling component consists of a set of controllers and components, which provide the functionality associated with the control and workflow of the component. These are the *test*, *test assignment* and *rule controller* as well as the *test data handler component*.

In general, the test, test assignment and rule controller are responsible for receiving and preparing user inputs from the corresponding views to be passed to the data handler component. Additionally, they prepare the data obtained from the data handler component to be presented to the user by the different views. The test controller provides all actions for creating, editing and deleting adaptive tests. This also includes adding questions to a test as well as changing their order. The test assignment controller provides all actions for assigning adaptive tests to students or groups of students. Additionally, the rule controller provides all actions for creating, editing and deleting rules. A rule consists of one or more conditions, which either refer to student's performance in the test or to a specific value in student's profile, and an action that will be executed when the conditions have been met.

The test data handler component encapsulates all core functions, which are shared between the adaptation modeling component controllers. This encompasses creating, editing and deleting adaptive tests including rules, conditions and actions. In doing this, the data handler instantiates all related models and make them available by a set of standardized data access functions. Furthermore, the data handler component provides access to the question and user modeling component and its models, respectively. This enables referring to question details (e.g., specific feedback) or to a specific feature in students' profile when creating adaptive rules.

4.7.3.4 *Presentation Layer*

The presentation layer of the adaptation modeling component consists of three main views (editors), each encapsulates certain functionality. The *test editor* concentrates all functions concerning the creation, editing and deletion of adaptive tests. The *rule editor* concentrates all functions related o the creation and editing of adaptation rules and the *test assignment editor* concentrates all functions concerning the assignment of adaptive or non-adaptive tests to students or groups of students. The views or the component as such are exclusively designed for being used by authors. A graphical representation of the various ways, an author can interact with the adaptation modeling component, is depicted in Figure 4.34.

4.7.3.5 *Summary*

This section has presented the design of the adaptation modeling component according to the MVC structure presented in Section 4.2.4. It is based on a variety of interrelated models such as the test, section, rule, and condition model. These models are used by different controllers to process user inputs and to initiate the rendering of different views. The views are presented in different editors such as the test editor for creating, editing and deleting adaptive tests or in the rule editor for creating adaptation rules. The adaptation modeling component is exclusively designed for being used by authors.

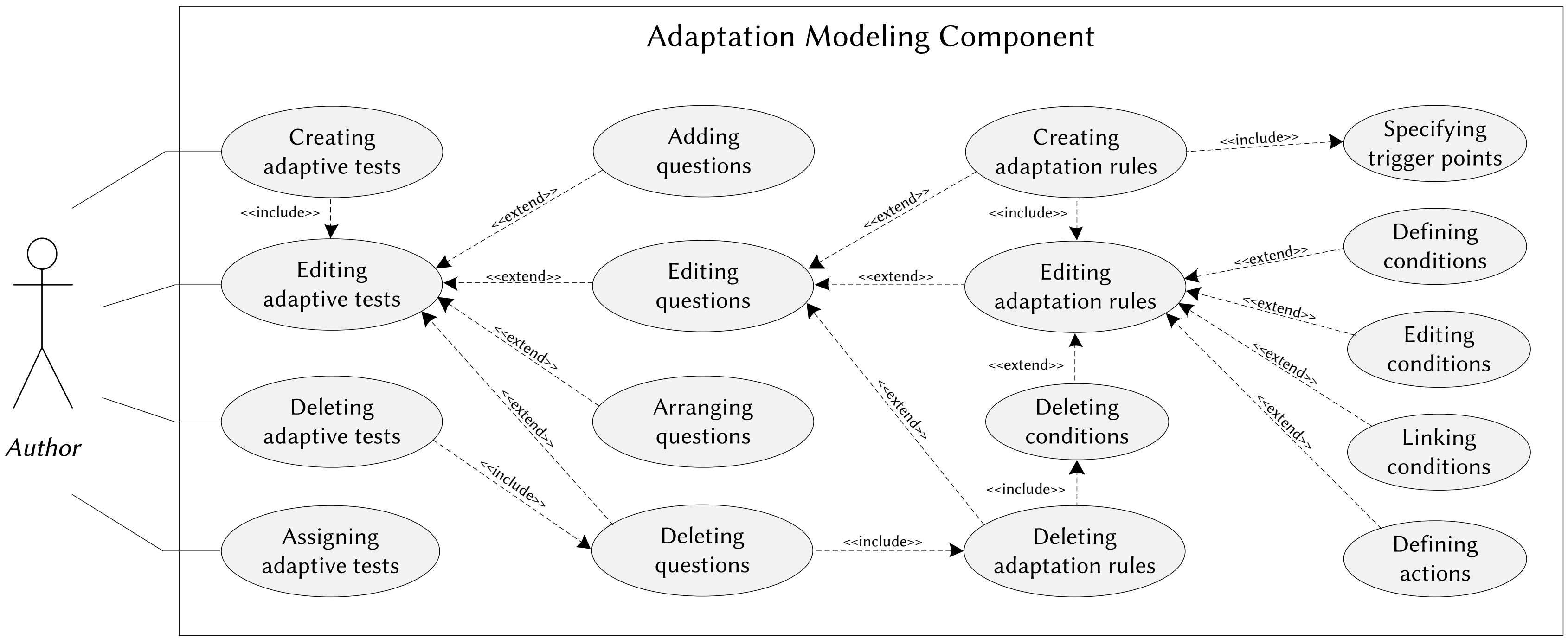


Figure 4.34: Use case diagram of the adaptation modeling component

4.7.4 *Adaptive Testing Engine Component*

4.7.4.1 *Introduction*

This section focuses on the design of the adaptive testing engine component. Figure 4.35 depicts the component based on the underlying MVC structure.

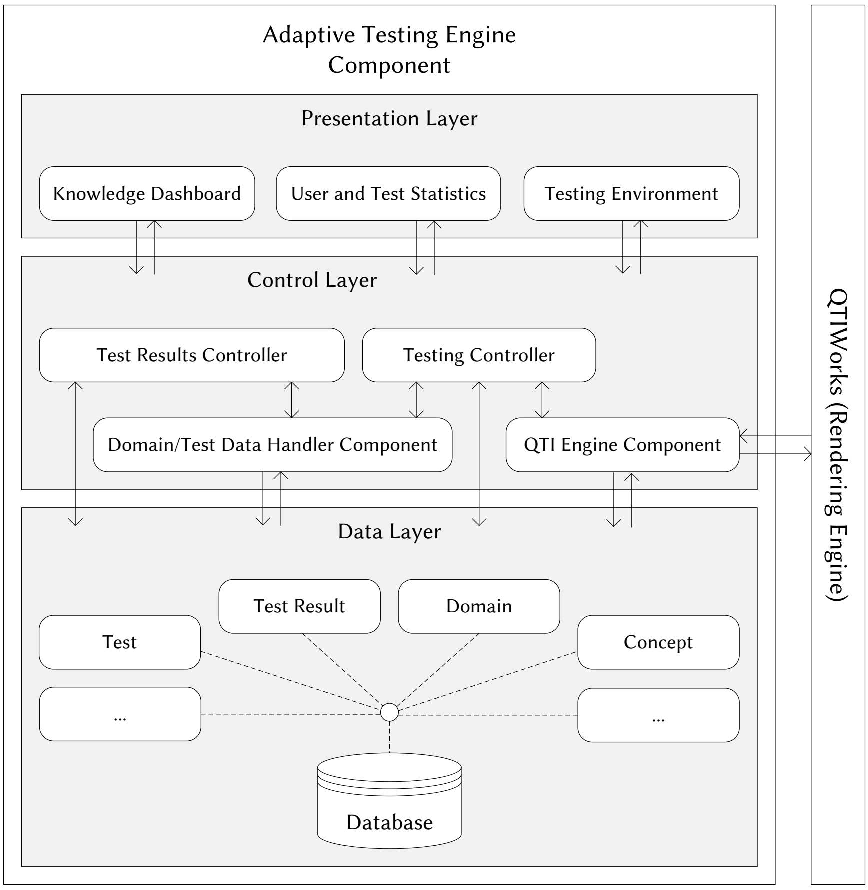


Figure 4.35: Architectural structure of the adaptive testing engine component

4.7.4.2 *Data Layer*

The adaptive testing engine component intensively uses models originally managed by other components. This includes the test, part and rule model, which are administered by the adaption modeling component (cf. Section 4.7.3.2) as well as the domain and concept model originally managed by the domain modeling component. However, the component also maintains an own model:

- *Test Result*: This model represents the results achieved by the users of the (adaptive) tests.

4.7.4.3 *Control Layer*

The control layer of the adaptive testing engine component comprises the *testing* and *test results controller*, the *domain* and *test data handler* as well as *QTI engine component*. The domain and test data handler component is reused from the domain and adaptation modeling component, respectively.

Testing Controller

The testing controller is responsible for performing the actual adaptation by implementing the adaptive tests, interpreting the including adaptation rules, presenting the questions to the students and processing their responses. It also records students test results to be evaluated by the test results controller later on. For presenting the questions of the tests and evaluating the responses (except for ICO questions), the adaptive testing engine component makes use of the QTIWorks[16] rendering engine. QTIWorks is an open-source tool for delivering QTI v2.1 assessment items and tests. It allows rendering questions in QTI format and evaluates students responses according to the response processing included in the file. Due to the fact that the IMS QTI specification was selected as basis on which the QM was built on, any QTI engine such as QTIWorks could be used easily for the own purposes. Moreover, the use of an established engine saved the effort of implementing an own rendering engine. The QTIWorks rendering engine is not closely coupled to the testing controller, but encapsulated by a QTI engine component. This component easily allows replacing the QTIworks engine at a later time without the need to modify the testing controller. In addition, this also allows using the question rendering to be included in other components (e.g., for previewing questions in the question modeling component).

QTI Engine Component

Before explaining the overall testing process and how the testing controller and QTIWorks are involved, it is advisable to be aware of the main structure of an adaptive test (cf. Figure 4.36). An adaptive test has a specific start and ending and consists of a set of question blocks. Simply put, a question block is a question/task plus optional rules at the beginning or/and at the end (in the following referred to as pre- and post-question rules). The design and implementation of the adaptation model also provides the opportunity to define rules based on interactions within a question/task (cf. Section 4.7.3). Summarized it can be stated that rules can be triggered before presenting, during the interaction with and after answering/solving the question/task.

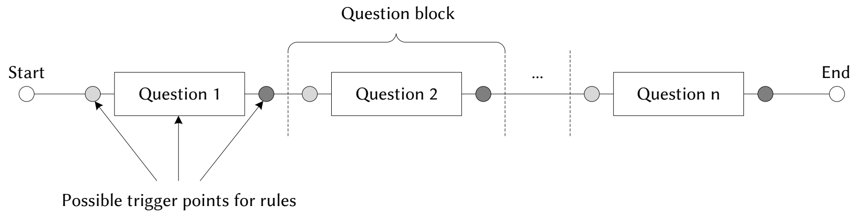


Figure 4.36: Main structure of an adaptive test

16 https://www2.ph.ed.ac.uk/qtiworks/ [last visited: May 21, 2014]

The overall testing process including all parties involved (i.e., student, testing controller and QTIWorks) is presented in Figure 4.37. Furthermore, a more detailed view how the question blocks are processed (including searching for rules and performing of actions) is depicted in Figure 4.38. As shown in the figure, the test results will be stored immediately after the student answered/solved the question/task. Due to the fact that the component inherits student's knowledge from these results, this means that succeeding adaptation rules (i.e., pre- and post-question rules) will be evaluated using this updated knowledge. Therefore, adaptations can not only be made prior to initiating, but also within a test session. Lopes and Bidarra [LB11] introduced the terms *offline* and *online* for these kinds of adaptivity.

As depicted in Figure 4.37, when a student responds to a question, the response is forwarded by the testing controller to the QTIWorks engine for evaluating. The information that will be returned by QTIWorks are student's plain response (e.g., the answer options selected), the score achieved as well as the time a student took in answering the question. Together with question details such as the corresponding test, the competency and conceptual relationship and the timestamp, this information is stored as a test result record in the database. The component not only uses these results to compile statistics, but also to update student's knowledge profile. Actually, the component does not maintain student's knowledge, but the probability that he or she has a specific (cognitive) ability corresponding to a certain thematic (domain) concept. While the cognitive ability and the thematic concept are derived from the question and its competency and conceptual relationship, respectively, the probability of student's knowledge will be calculated after each question/task response. Simply put, the probability increases when a student answered correctly and decreases if he or she was not successful. The extent to what the probability will increase and decrease is defined by the author. Consequently, the testing controller uses both values to calculate the probability increase or decrease. The detailed calculation rules are as follows:

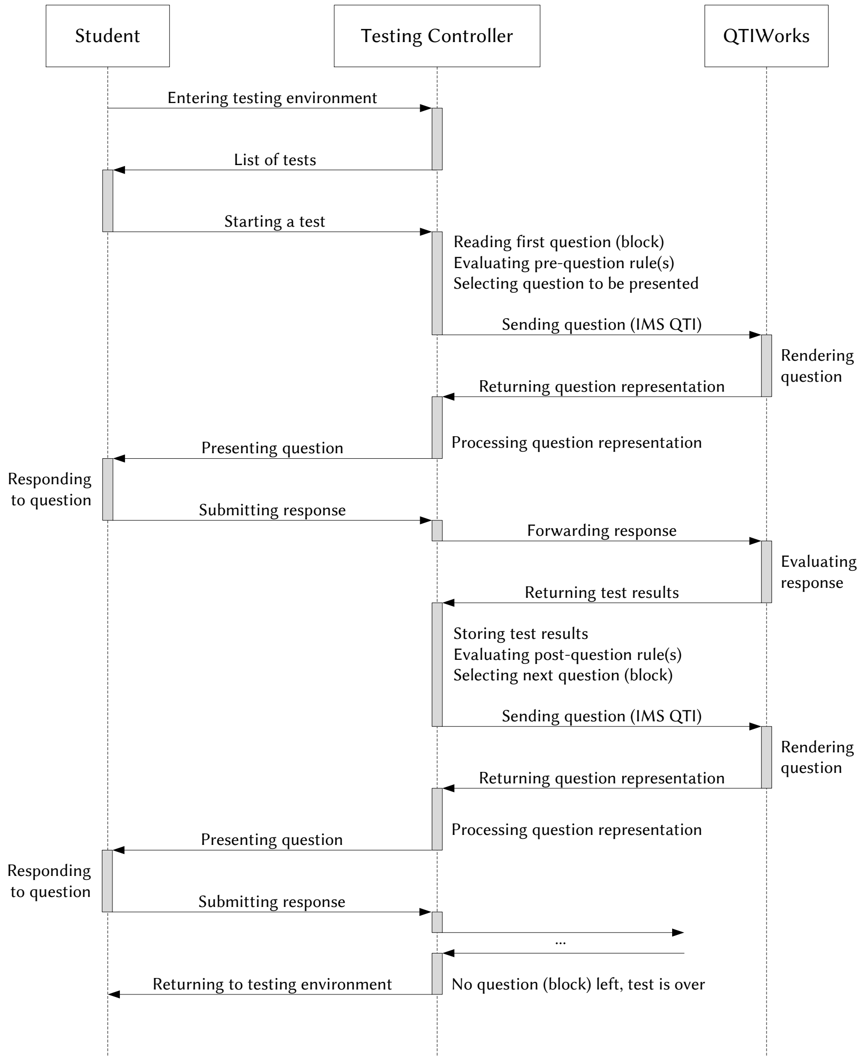


Figure 4.37: Overall testing process

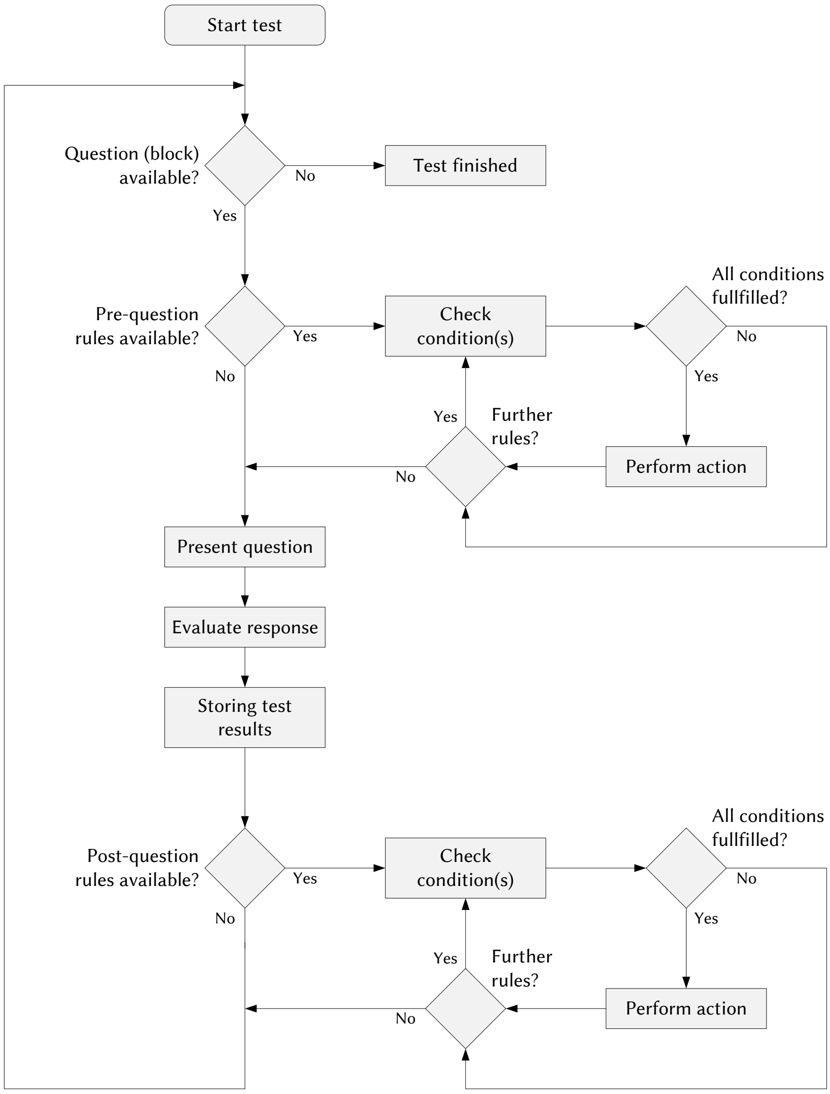


Figure 4.38: Processing of a question (block) consisting of pre- and post-question rules

If the student answered correctly and has no prior knowledge (probability):

$$P_{new} = V_{inc}$$

If the student answered correctly and has a prior knowledge (probability):

$$P_{new} = P_{old} + V_{inc}, \text{ where } 0 \leqslant P_{new} \leqslant 1$$

If the student answered incorrectly and has no prior knowledge (probability):

$$P_{new} = 0$$

If the student answered incorrectly and has a prior knowledge (probability):

$$P_{new} = P_{old} - V_{dec}, \text{ where } 0 \leqslant P_{new} \leqslant 1$$

where

- P_{new} is student's new (calculated) probability
- P_{old} is student's old (latest stored) probability
- V_{inc} is the increasing value
- V_{dec} is the decreasing value

Test Results Controller

The test results controller is responsible for compiling statistics about students' test results. These statistics are made available for both authors and students. The primary reason to open the test results for students is to encourage reflection about his or her learning and knowledge (cf. Section 4.5.4.4). In this way, students are able to examine their own as well as system's perspective of their knowledge (meta-cognition). The main reason for making the results available to teachers is not only inspection (i.e., monitoring the current status), but also tuning (e.g., adapting teaching or supporting students with particular problems).

4.7.4.4 *Presentation Layer*

The presentation layer of the adaptive testing engine component consists of the *knowledge dashboard*, *user and test statistics* and the *test environment*. While the knowledge dashboard concentrates all functions related to the presentation of student's own test results, user and test statistics groups all functions concerning the presentation of statistics about students' test results. In contrast, the test environment is dedicated to the execution of adaptive and non-adaptive tests. A graphical representation of the various ways authors and students can interact with the adaptive testing engine component is depicted in Figure 4.39.

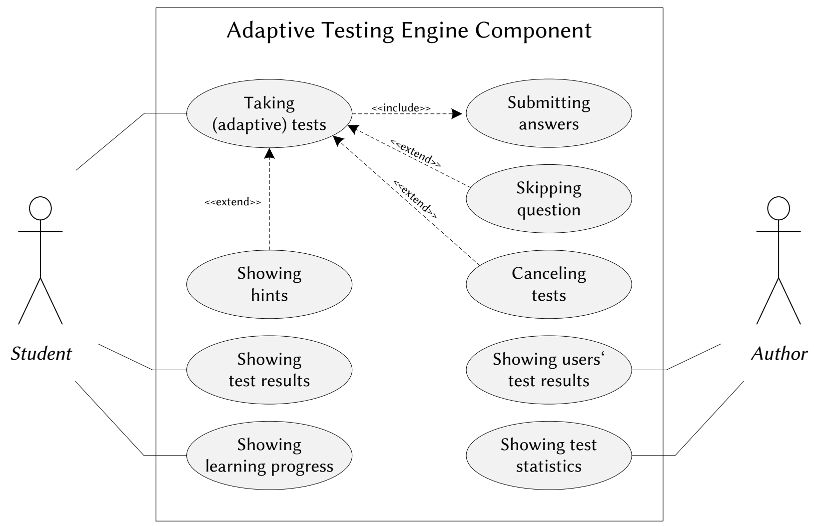


Figure 4.39: Use case diagram of the adaptive testing engine component

4.7.4.5 *Summary*

This section has presented the design of the adaptive testing engine component. It performs the actual adaptation by taking adaptive tests as input and applying the rules according to student's actions, but also compiles statistics about students' test results and makes them available for both authors and students. Although this component uses a variety of models originally managed by other components, but also specifies an own model (i.e., test result) to process user inputs and to initiate the rendering of the different views. The functionality associated with the control and workflow of the component is provided by the testing and test results controller. Finally, the presentation layer of the adaptive testing engine component consists of three main views (i.e., knowledge dashboard, user and test statistics, and the test environment).

4.7.5 *Summary*

This section has given an insight into the design considerations of the different architectural components. Based on three application scenarios, a set of functional and non-functional requirements were derived that imposed constraints on the design and implementation of the different architectural components. As an example, the design of the adaptation modeling and the adaptive testing engine component has been described. Both components are based on a variety of interrelated models that are used by different controllers to process user inputs and to initiate the rendering of different views. The views are presented in different editors such as the test editor, the knowledge dashboard and the test environment.

4.8 INTEGRATION WITH ESTABLISHED LEARNING ENVIRONMENTS

4.8.1 *Introduction*

Over the last few years, a variety of LMSs and LCMSs have been developed. They were designed to support and enhance learning and training in educational settings and also mostly provide opportunities to require students to answer questions at the beginning, during or at the end of a learning activity (cf. Section 2.2.4). However, the question types available are very limited and personalization is still insufficiently implemented or even not addressed by these systems. Hence, the integration and communication with established systems and tools is a prerequisite for a prompt and widespread adoption of the model and its implementation developed in this thesis. This section focuses on the design of a seamless integration of LMSs and LCMSs, in the following referred to as LEs, and (external) personalized e-assessment systems, in the following referred to as AASs. Parts of this section were already published in [SDW11].

4.8.2 *Requirements*

In order to enable a successfully integration of LEs with AASs, the LE requires *control* as well as *assessment information* from the AAS. Moreover, in order to seamlessly launch the tests and questions provided by AASs by any LE, the AAS require *student* as well as *assessment information* from the LE (cf. Figure 4.40). 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 activity, for example, to lock further content or to start the test environment again upon logging into the LE for the next time. This information needs to be communicated continuously until the LE has been informed that the student has finished the test. In addition, the AAS should provide mechanisms to resume the e-assessment in case the connection was interrupted or lost. The LE needs to be informed about the results achieved after the student had finished answering the questions. This kind of information encompasses the question that was asked, the final answer, the scores achieved as well as the attempts the students took in getting the final answer. The assessment information can be used by the LE to report and compute overall test scores. Furthermore, 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. Due to the fact that AASs personalize the assessment, they need to be informed about the individual characteristics of the student. A further requirement of the AAS with respect to student information is to uniquely identify students.

Control Information

Assessment Information

Student Information

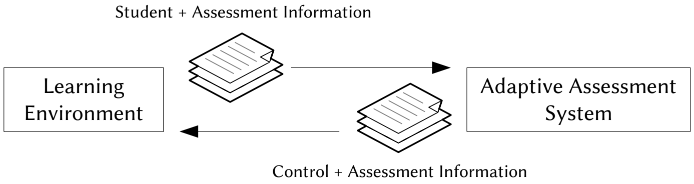


Figure 4.40: Information exchange

4.8.3 *Related Work*

There are two standards/specifications to realize communication between LEs and external e-assessment systems namely Remote Question Protocol (RQP) and Open Protocol for Accessing QUestion Engines (OPAQUE). RQP is a web service protocol based on SOAP[17] that has been developed by the Serving Mathematics project aiming at developing assessment tools in mathematics education. Due to the fact that mathematical questions often require intensive processing, different question engines were developed dedicated to certain question formats. In order to allow LEs to provide support for the specific question formats, an attempt to develop a standard interface has been started. Although RQP looked very promising by making it possible to access different assessment tools through a single interface, the effort ran out of resources and the protocol has never been finished. Conceptually similar to RQP and in a working state is OPAQUE[18]. It 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 passes a response back to the calling LE. Although OPAQUE has been implemented into the LE Moodle and as well as into the question engines OpenMark[19] and Stack[20], it is designed to allow interoperability between arbitrary different types of LEs and question engines.

RQP

OPAQUE

4.8.4 *Proposed Solution*

As stated above, LEs require control information to determine the state of the student in the learning activity/process. Thus, the communication needs to be based on a session management, which enables pausing and resuming in case the students interrupts the assessment session or loses 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, this was one reason to select this specification as general communication protocol for the own approach. It proposes a data structure and

17 `http://www.w3.org/TR/soap/` [last visited: May 21, 2014]
18 `http://docs.moodle.org/dev/Opaque` [last visited: May 21, 2014]
19 `http://www.open.ac.uk/openmarkexamples/` [last visited: May 21, 2014]
20 `http://stack.bham.ac.uk/` [last visited: May 21, 2014]

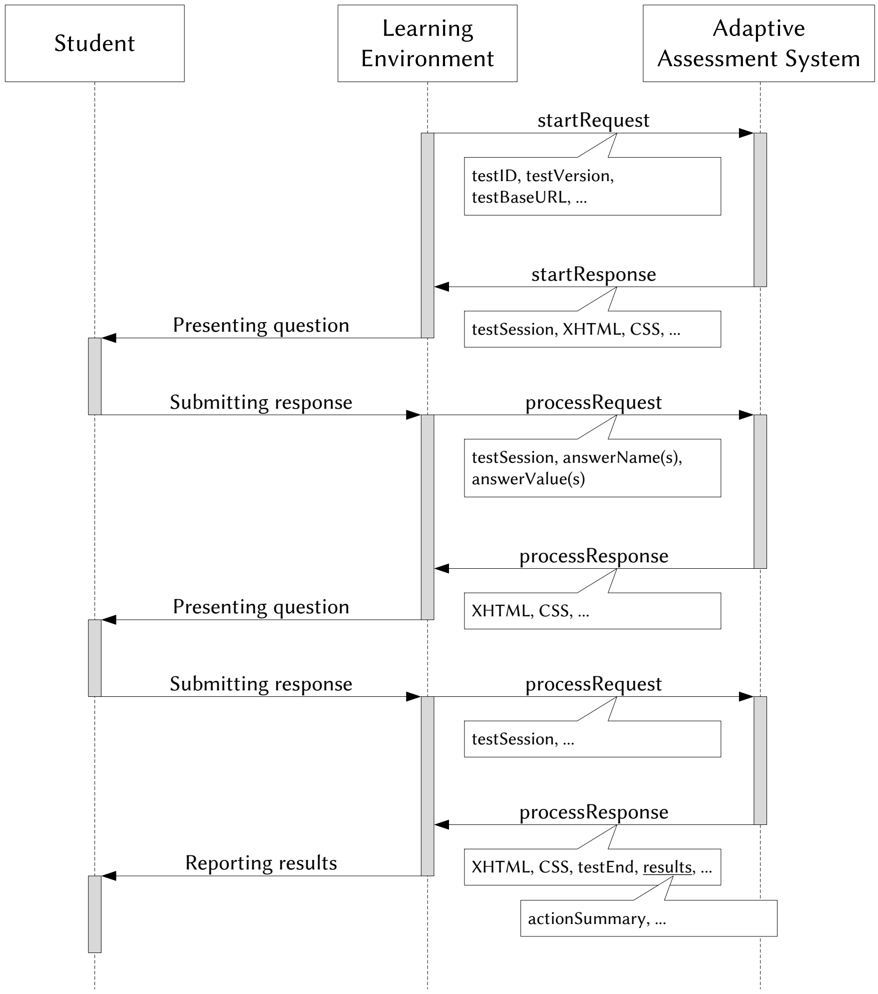


Figure 4.41: Communication between student, LE and AAS

a communication mechanism to achieve a seamless communication between LEs and AASs. The approach facilitates the communication of *control*, *student* and *assessment information* and thus enables personalized e-assessment. The communication includes the following actions:

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

Figure 4.41 presents a UML sequence diagram that shows the overall communication between the partners involved (i.e., student, LE and AAS). More details about the design and the implementation can be found in Appendix F.

4.8.5 *Summary*

This section has presented the design of a seamless integration of LEs and AASs. Based on the requirements of the information to be exchanged namely *control*, *assessment* and *student information*, a communication mechanism was proposed that allows launching and completing (adaptive) tests, but also communicating between AAS and LE at run-time. The proposed mechanism is based on the open-source web service protocol OPAQUE, which is currently implemented among others by Moodle, OpenMark and Stack.

4.9 SUMMARY

This chapter focused on the design of the architectural model, whose components and sub-models developed in this thesis. It enables integrating ICOs and allows complementing these with personalization aspects. Section 4.2 defined the overall structure of the model and its constituent components. The *user*, *domain* and *adaptation modeling* as well as the *question modeling component* maintain a specific model, which determines whose behavior and functionality. The following sections were dedicated to describe these models in more detail.

Section 4.3 presented the design of the Question Model (QM) approach. It enables describing and initializing ICOs as well as processing students' responses resulting from interacting with them. As a result, the QM was not completely designed from the scratch, instead the IMS QTI specification was selected as basis and specifically extended with a data model for the integration of and the interaction with ICOs. The data model is called MICO. Furthermore, due to the fact that the processing of responses requires passing information back and forth, not only a QM, but also a Communication Schema (CS) and a Communication Mechanism (CM) was designed. In order to promote interoperability, the Experience API specification was used as basis for this mechanism.

Section 4.4 proposed the design of the Domain Model (DM) approach. It has a key role in the overall architectural model because it represents the body of knowledge that makes up a (knowledge) domain and serves as the basis for relating questions and students' features. As a result, the ontology approach has been used to represent and organize knowledge. Furthermore, the OWL specification was chosen as formal specification of the DM to support interoperability of the model and its information.

Section 4.5 provided the design of the User Model (UM) approach. As an important pillar of the overall model, it enables representing a variety of user features and allows making inferences about the users. As a result, the UM consists of the user features *knowledge*, *interests*, *goals*, *background* and *demographic information*. Moreover, the overlay modeling approach was chosen in order to represent the features selected. Additionally, the UM approach is based on the categories of IMS LIP for structuring students' demographics in order to promote interoperability and reuse of this information.

Section 4.6 addressed the design of the Adaptation Model (AM) approach. It can be regarded as the core of an adaptive system because it defines what can be adapted, as well as when and how it has to be adapted. As a result, the elements, which are adapted by the adaptive methods, are the *sequence*, *selection*, *presentation* and *difficulty* of questions as well as the *selection* and *presentation* of feedback. The information that is used to provide adaptations are student's *performance* and student's *features*. The pedagogical reasons behind the adaptations are the *identification* of students' strength and weaknesses more valid and reliable, the *consideration* of students' strengths and preferences to compensate weaknesses and deficits and the *encouragement* of students' motivation by considering individual aspects. The adaptation information is directly obtained from the adaptive testing engine during the assessment process and the user modeling component. Besides, for proposing and aiding decisions about alternatives, the AM uses the *rule-based reasoning* approach. Thus, the AM corresponds to the adaptive questions technique.

Following the description of the different models, Section 4.7 has given an insight into the design of the different architectural components based on the overall design pattern. Finally, Section 4.8 has presented the design of a seamless integration of LEs and AASs.

5 IMPLEMENTATION

5.1 INTRODUCTION

This chapter gives an insight into the implementation of the architectural model proposed in Chapter 4. The implementation is realized by the web-based e-assessment system *askMe!*. It covers the whole life-cycle of e-assessments starting from creating questions over to presenting it to the students up to preparing the results and presenting them to the teachers. The questions and tests can consider individual aspects so that e-assessments can perfectly be tailored to students or groups of students (personalization). Moreover, the author of the adaptive tests is not limited to traditional question types such as multiple-choice or hotspot, but can use Interactive Content Objects (ICOs) (cf. Section 3.3.2) to create sophisticated (interactive) e-assessments. First of all, Section 5.2 briefly describes the main technologies used for realizing the *askMe!* system. Then, Section 5.3 presents the implementation of two components of the system in detail, structured according to the overall system design pattern (cf. Section 4.2.4). The implementation of the remaining components is described in Appendix E. Finally, Section 5.4 summarizes the main results.

5.2 IMPLEMENTATION TECHNOLOGIES

5.2.1 *Introduction*

In this section, the technologies used for realizing the *askMe!* system are briefly described. For the web-based implementation, the open-source general-purpose server-side scripting language PHP in combination with open-source web application framework CakePHP was used. For structuring and presenting content, it has made use of Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS). In order to give the system an attractive and easy to use Graphical User Interface (GUI), the JavaScript library jQuery as well as the Asynchronous JavaScript and XML (AJAX) technology was used. Finally, the Extensible Markup Language (XML) was used in many different locations inside the system (e.g., to import and export content created).

5.2.2 *PHP and CakePHP*

PHP[1] is an open-source server-side scripting language. It is mainly used for creating dynamic web applications because it can easily be embedded into

1 `http://www.php.net/` [last visited: May 21, 2014]

HTML pages. PHP source code is not sent to the browser directly, but need to be interpreted by a web server with a PHP processor module. The resulting web page can then be displayed using a standard web browser. The PHP code remains on the server and is hidden for the user. PHP is characterized by a wide database support and internet protocol connection as well as by the availability of many additional libraries. With the release of version 5.0, PHP included new features such as improved support for object-oriented programming, which makes the development of web applications even more efficient. At the moment of writing, *askMe!* uses PHP version 5.1.6.

CakePHP[2] is an open-source web application framework that uses well-known software engineering concepts and follows the Model-View-Controller (MVC) software design pattern as prescribed by the overall system design pattern (cf. Section 4.2.4). The framework provides a basic organizational structure (e.g., file names and database table names) to rapidly build web applications. CakePHP takes the monotony (e.g., validation, database interaction and localization) out of web development so that it can be concentrated on the logic specific to the respective application. At the moment of writing, *askMe!* uses CakePHP version 1.3.

5.2.3 *HTML and CSS*

HTML[3] is the main markup language for displaying web pages and other information in a web browser. It allows creating structured documents using text, graphics and hyperlinks by the use of structural semantics. For describing the presentation semantics (i.e., the look and formatting) of these structures, the style sheet language CSS is mostly used. This separation of document content (written in HTML) from document presentation, including elements such as the layout, colors and fonts (written in CSS) improves the overall content accessibility and provides more flexibility and control in the specification of presentation characteristics. Furthermore, it reduces the complexity and repetitions in the structural semantics.

The latest version of HTML, which is still under development, is HTML5[4]. It adds many new syntactic features that easily allow handling multimedia and graphical content on the web without having proprietary plug-ins or APIs. This is realized by the new *video*, *audio* and *canvas* elements, as well as the integration of Scalable Vector Graphics (SVG) content and MathML for mathematical formulas. Particular noteworthy is the canvas element, which allows for dynamic, scriptable rendering of 2D shapes and bitmap images. The *askMe!* system used this element within the adaptation modeling component for modeling the adaptive tests in a graphically appealing way.

2 `http://www.cakephp.org/` [last visited: May 21, 2014]
3 `http://www.w3.org/TR/html/` [last visited: May 21, 2014]
4 `http://www.w3.org/TR/html5/` [last visited: May 21, 2014]

5.2.4 *JavaScript and jQuery*

JavaScript is a scripting language for creating enhanced GUIs and dynamic web pages. It is primarily used on client-side to be executed in a web browser so that user inputs can locally be interpreted and new content generated/displayed without the need to reload the entire page. JavaScript uses the DOM[5] for interacting with HTML and CSS elements. The DOM is an abstraction of the web page structure in the form of a hierarchical tree. The nodes of the tree can be accessed using JavaScript and whose attributes read, edited or removed.

jQuery is a open-source multi-browser JavaScript library that simplifies the client-side scripting. It easily allows navigating through the web page structure, selecting DOM elements, creating animations, handling events and developing AJAX applications. Moreover, it circumvents the differences and weaknesses of various web browsers and provides a uniform cross-platform scripting environment. jQuery also allows developers to create plug-ins on top of the JavaScript library. One of these libraries, which is used by the *askMe!* system is jQuery UI[6]. It provides a set of GUI interactions (e.g., drag and drop or sorting), effects, widgets and themes to build highly interactive web applications. At the moment of writing, *askMe!* uses jQuery version 1.9.1 and jQuery UI version 1.10.1.

5.2.5 *AJAX*

AJAX describes the concept of sending and retrieving data between a server and a web browser asynchronously without interfering with the display and behavior of the existing web page. It allows creating web applications that behave similarly to applications running on a desktop computer. The *askMe!* system uses this concept to realize smooth interactions and immediate system responses without having long waiting times caused by page reloads. This increases the usability of the *askMe!* system significantly.

It is not a single technology, but a group of technologies:

- HTML and CSS for the presentation
- DOM for the interaction with the data
- XML or JavaScript Object Notation (JSON) for the exchange of the data
- XMLHttpRequest object for the asynchronous communication
- JavaScript for bringing these technologies together

5 `http://www.w3.org/DOM/` [last visited: May 21, 2014]
6 `http://www.jqueryui.com/` [last visited: May 21, 2014]

5.2.6 *XML*

XML[7] is a markup language for encoding documents in a structured and hierarchical way. It is not only human-, but also machine-readable. In contrast to HTML, XML has no predefined elements, they can be defined in any way. Elements are marked by tags, which in turn can have attributes. Attributes are defined by a name/value pair. Due to its flexibility, a variety of markup languages based on XML evolved over the time. For example, IMS QTI and Moodle XML (cf. Section 3.3.4) are two representatives of these markup languages.

5.2.7 *Summary*

This section has presented a brief description of the main technologies used for realizing the *askMe!* system.

7 `http://www.w3.org/TR/REC-xml/` [last visited: May 21, 2014]

5.3 ARCHITECTURAL COMPONENTS

5.3.1 *Introduction*

This section gives an insight into the implementation of the architectural components within the *askMe!* system. Section 5.3.2 and 5.3.3 show and describe the presentation layer of the adaptation modeling component and the adaptive testing engine component, respectively. Implementation details about the other components can be found in Appendix E.

5.3.2 *Adaptation Modeling Component*

This section focuses on the implementation of the presentation layer of the adaptation modeling component. As presented in Section 4.7.3, the presentation layer consists of three main views (editors) namely the *test editor*, the *rule editor* and the *test assignment editor*. Parts of this implementation were supported by Kerstin Heyder in the course of her diploma thesis [Hey12].

5.3.2.1 *Test Editor*

The test editor concentrates all functions concerning the creation, editing and deletion of adaptive tests. The starting point of this editor is a list of tests created by the author. A test is specified by a *title* and a reduced *description* as well as *creation* and *modification date*. Using the icons at the end of each row, tests can be edited or deleted immediately. New tests can be created by clicking on the *New Test* button below the table. When selecting a specific test, the title and the whole description of the tests are shown again. Underneath this area, the structure of the adaptive test is presented (cf. Figure 5.1). It shows the questions included in this test vertically arranged according to their order in the test. They are represented by a block that shows its title and question type. New questions can be added to this list using the button in the footer of the area. Then, a window will pop up showing a list of all questions that can be added to the test. The questions will be inserted at the end of the list, however, the order can be intuitively changed by using drag and drop. This functionality, but also the whole test structure was realized using JavaScript and HTML5. Each question block in this structure shows a black square in the top right corner. By clicking on it, the author can choose between deleting the respective question and adding an adaptive rule. After creating a rule (using the rule editor), this rule will be visualized in this structure by an additional block, titled by the name of the rule and linked to the associated question with a line. In this way, requirement AM-1 regarding the adaptation modeling component (cf. Section 4.7.2) is fulfilled.

5.3.2.2 *Rule Editor*

The rule editor concentrates all functions related o the creation and editing of adaptation rules. When an author creates a new rule, he/she will

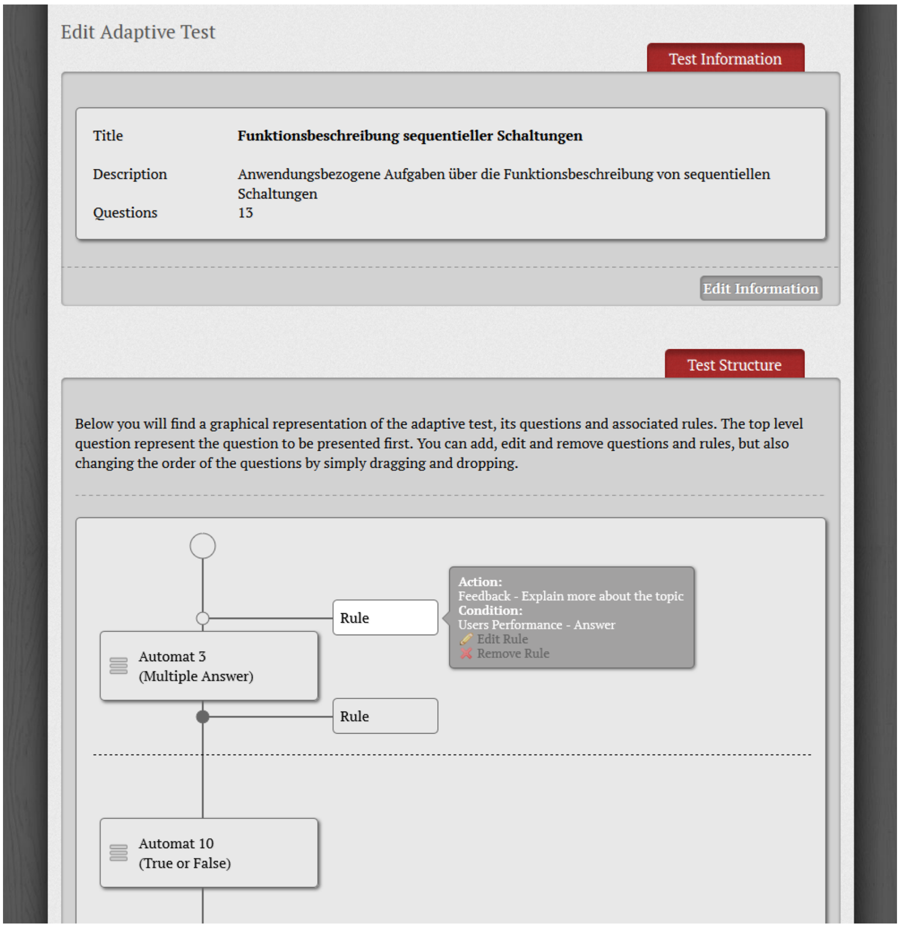


Figure 5.1: askMe! test editor

be presented with a graphical visualization that refers to the structure of the question selected (cf. Figure 5.2). This visualization corresponds to the adaptivity modeling approach using FSMs proposed in Section 4.6.4.5. Here, it is mainly used to graphically support the authors in specifying trigger points for adaptation rules. In this way, trigger points can not only be set before presenting (marked by the unfilled circle) and after answering a question (marked by the filled circle), but also during the interaction with the question. That means, when a student has selected a specific answer option. This is realized by a FSM state diagram that shows the author a network of *nodes* and *edges*. The nodes represent any possible state when answering the question. In contrast, the edges represent any possible transition from one state to another (e.g., a student switches between different answer options). Due to the fact that with growing number of answer options, the number of states and state transitions is increasing as well and the state diagram is become confusing, only a subset of question types (i.e., true/false, multiple-choice, multiple-answer, match, slider and select point) are modeled using this graphical approach. For the other question types, trigger points can only be specified before presenting and after answering a question. Thus, requirement AM-3 is also met.

After specifying a trigger point, the author will be requested to name, but also to define the condition(s) and action of the rule. This view is subdivided into four areas. In the first area, the author can define a name for the rule so that he/she can identify it more easily. The second area focuses on defining conditions (cf. Figure 5.3). A single condition is enclosed by a rectangle in order separate them from each other. Conditions can be added and deleted as desired. If a rule has more than one conditions, they are linked to each other using logical conjunction (and) or disjunction (or). That means, when using logical conjunction an action will only be executed, if all conditions are true. On the contrary, when using logical disjunction an action will be executed, if at least one condition is true. In accordance to the adaptation model, the information that can be used to provide adaptations are students' performance and features (cf. Section 4.6.4.2). As a consequence, it can be chosen between these categories when defining a condition. Depending on the choice, different input fields are presented. For example, when selecting *students' performance* as adaptation information, students' answer, score and their response time is used to define conditions. The third area allows configuring the action to be executed. Here, the author can choose between different adaptation means such as *question sequence*, *question selection* and *feedback selection* (cf. Section 4.6.4.1). Similar to the adaptation information, depending on the selection, different input fields are given. For example, when choosing question sequence as adaptation mean, the author is able to move to a certain question. Consequently, the requirements AM-2 and AM-4 to AM-6 are also satisfied. Finally, the fourth area shows once again the trigger point specified in the previous step. When all inputs were made, the rule definition will be completed by saving the rule.

5.3.2.3 *Test Assignment Editor*

The test assignment editor concentrates all functions concerning the assignment of adaptive or non-adaptive tests to students or groups of students. To this end, the author is presented with a matrix. The y-axis lists all students that were assigned to him or her and on the x-axis, all tests created by the authors are listed. Now, the author is able to individually assign a test to a student by simply clicking on the intersection point in the matrix. The icon on the intersection point will change and the test is now assigned to the student. Removing an assignment is done in the same way. If an author wants to assign a test to more students, he or she can additionally use the list view of the test assignment editor. There, the author is able to make multiple selections. In this way, the remaining requirement AM-7 is also fulfilled.

5.3.3 *Adaptive Testing Engine Component*

This section focuses on the implementation of the presentation layer of the adaptive testing engine component. As shown in Section 4.7.4, the presentation layer consists of three main views namely the *knowledge dashboard*,

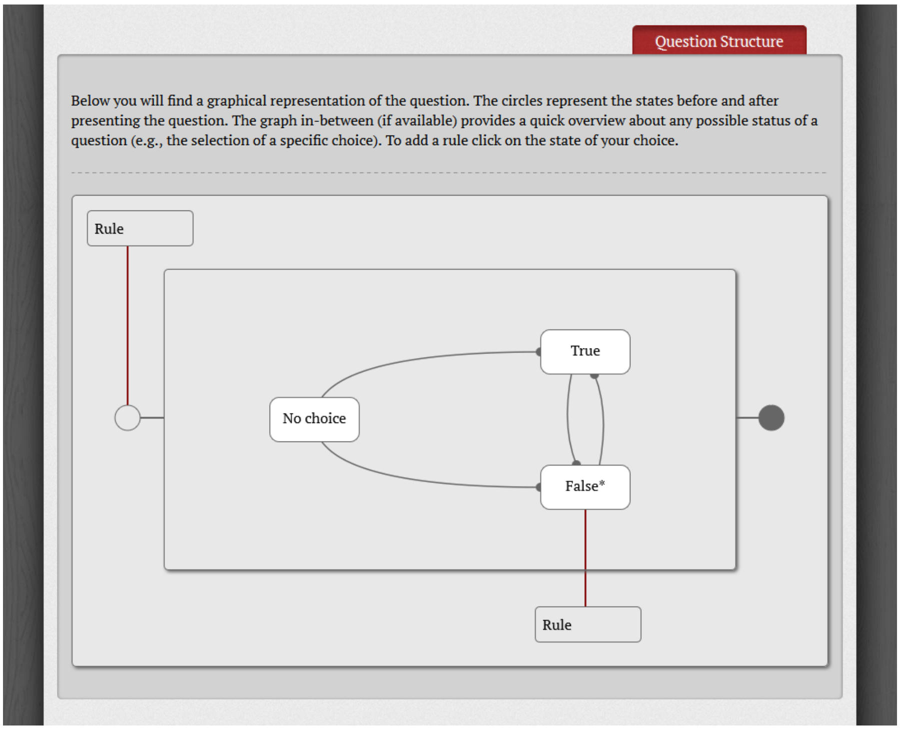


Figure 5.2: askMe! rule editor for setting trigger points

user and test statistics and the *test environment*. Parts of this implementation were supported by Nedal Alaqraa in the course of his master thesis [Ala14].

5.3.3.1 *Knowledge Dashboard*

The knowledge dashboard concentrates all functions related to the presentation of student's own test results. At first, the student is presented with several textual and graphical representations grouped in widgets that shows his/her test results, strengths, weaknesses, learning progress, etc. at a glance (cf. Figure 5.4). The widget on the left hand site shows the number of tests completed in relation to the number of all tests assigned. Underneath this widget, the number of questions completed by the student is presented as well as the average question score. The widget in the middle of the page lists concepts, the student knows best as well as concepts, the student had the most problems in. The widget on the right hand site shows the learning goals of the student and its current status. In addition, the different kinds of information are mostly complemented with bar charts. The reason for that is the presentation of the information not only in an additional, but also in a compact and intuitive way. The different colors contribute to this as well.

The knowledge dashboard also provides a more detailed view of students' test results as well as learning goals. The detailed view of the test results is depicted in Figure 5.5. It shows an overview of all tests the student completed recently. By clicking on a test, details about the questions included will be presented. This includes information about the concepts, knowledge levels and types addressed by the question, the time required to complete the

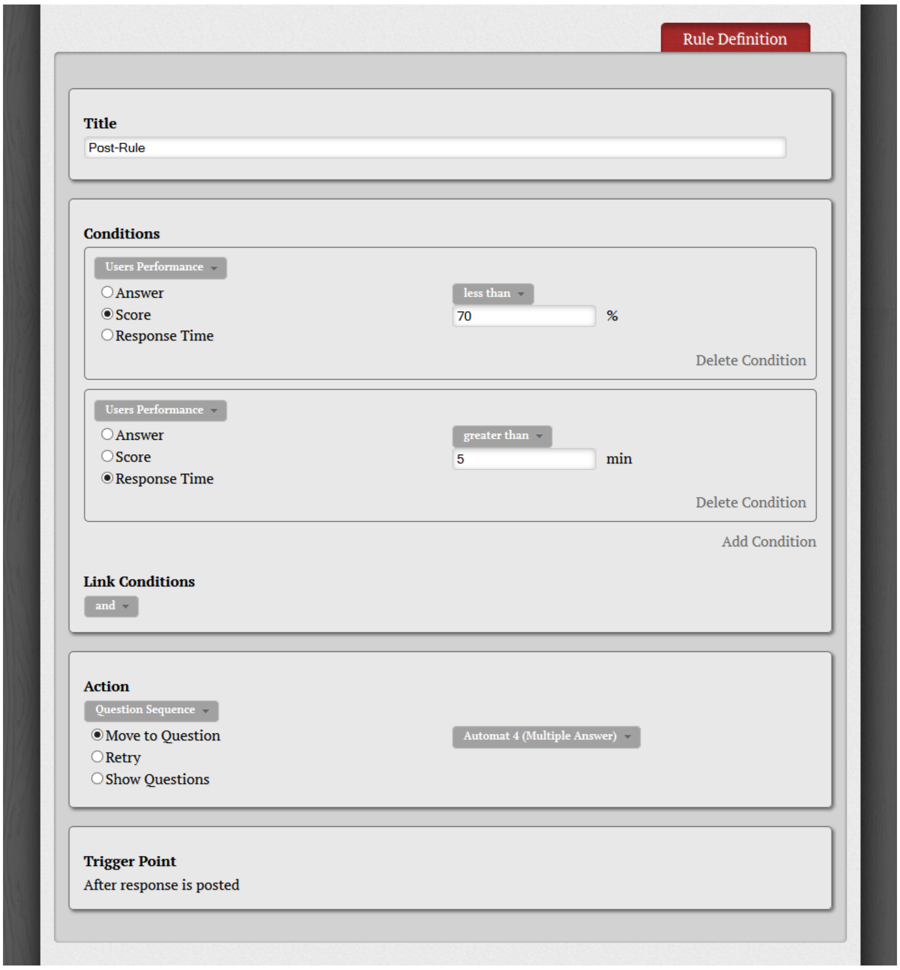


Figure 5.3: askMe! rule editor for setting conditions and actions

question as well as the score achieved for this question. Additionally, a graphical representation of the question results expressed as a percentage of the maximum question score is shown below.

The detailed view of the learning goals is in charge of presenting student's learning progress with respect to its learning goals. By selecting a learning goal, a line chart is shown, which represents student's learning progress over time. The points of the graph represent the questions completed and their contribution to the probability that a student achieved a specific goal. Each question answered correctly increases the probability. The higher the value, the more likely a student achieved this goal. Thus, requirement ATE-3 is also met.

5.3.3.2 *User and Test Statistics*

The user and test statistics concentrates all functions concerning the presentation of statistics about students' test results. It presents an overview of all students those supervision was assigned to the author as well as how many tests they completed and learning goals they achieved. In this way, requirement ATE-2 is satisfied.

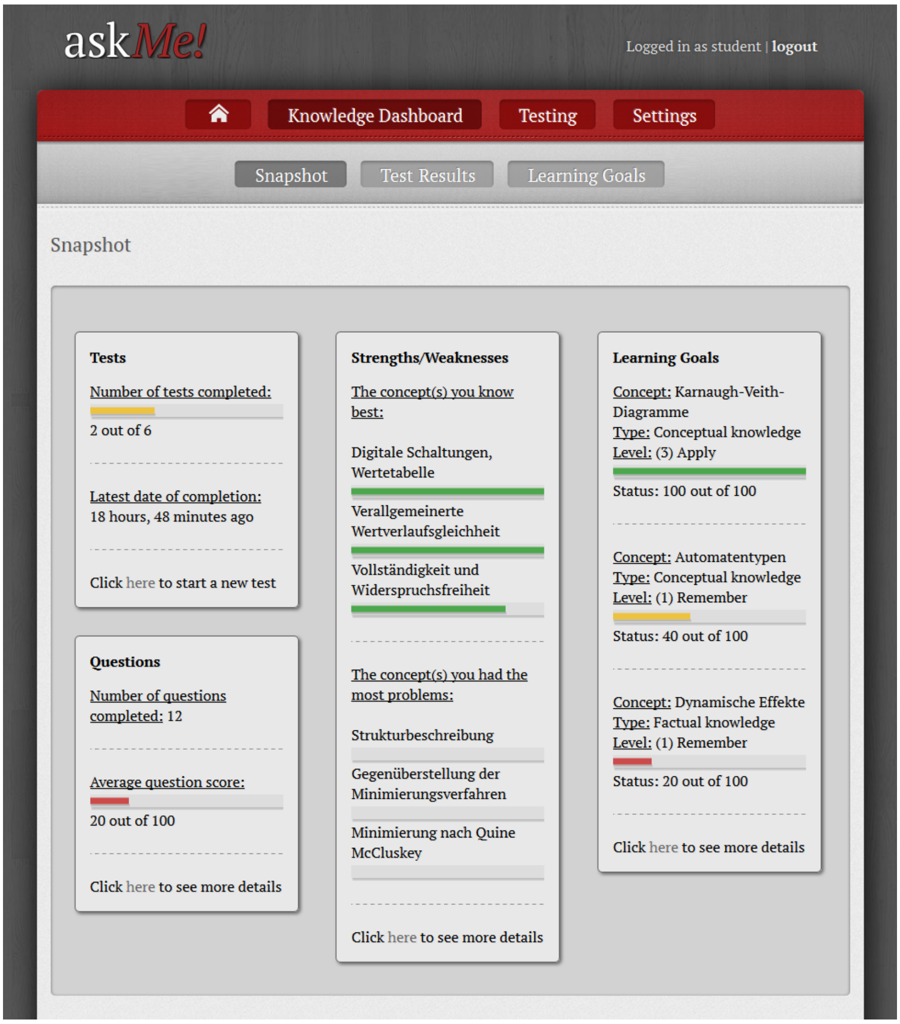


Figure 5.4: askMe! knowledge dashboard (snapshot)

5.3.3.3 *Test Environment*

The test environment concentrates all functions concerning the execution of adaptive and non-adaptive tests. The starting point of this environment is a list of tests assigned by authors. A test is specified by a *title*, a reduced *description* and a *status*, which shows the date of completion if the test was already taken. Using the buttons at the end of each row, tests can directly be *started* and *restarted*. In case a student decides to restart a test, prior results to this test will be discarded. Either way, the student will be directed to the actual test environment, which hides unneeded User Interface (UI) elements (e.g., the navigation) and allows focusing on completing the respective question or task (cf. Figure 5.7). There, the student is presented with the prompt of the question/task and interactions to answer/solve, but also with information about the surrounding test, the amount of question/tasks to be done (only available if the test is not adaptive) as well as buttons to submit an answer/solution, skip a question or cancel the whole test. Furthermore, if hints are added to the question, they can be shown on request. If the student

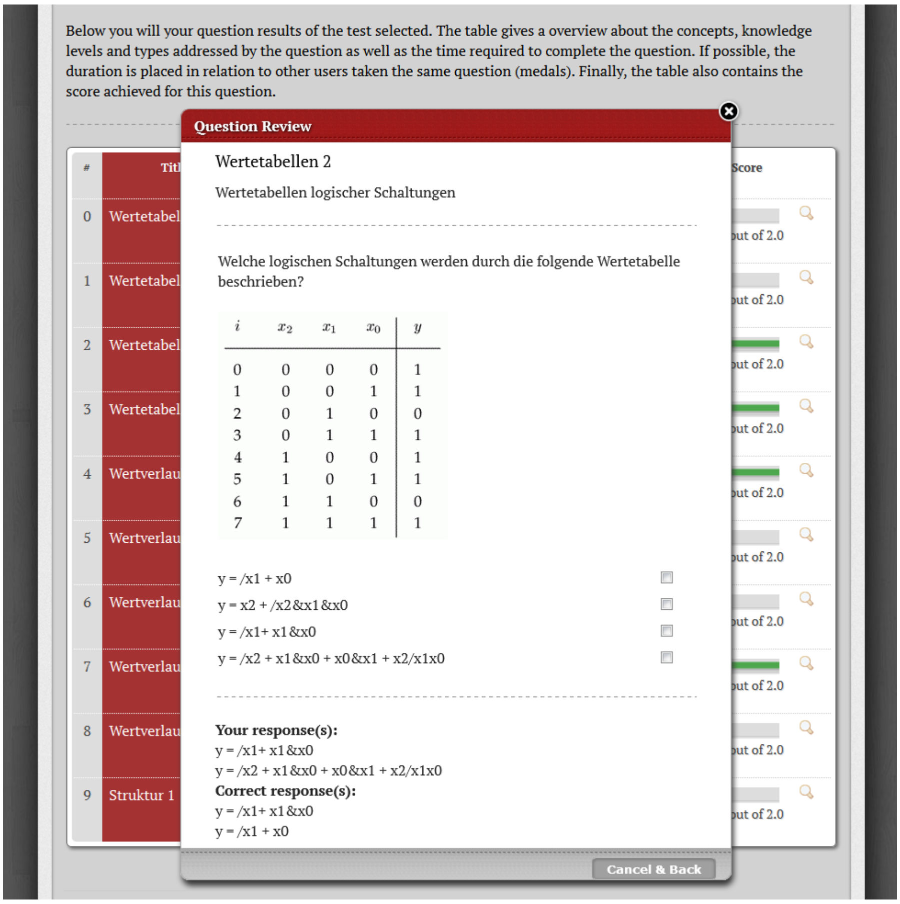


Figure 5.5: askMe! knowledge dashboard (test results)

completed the test, he/she will be redirected to the list of tests. Thus, the remaining requirement (i.e., ATE-1) regarding the adaptive testing engine component (cf. Section 4.7.2) is also fulfilled.

5.3.4 *Summary*

This section provided an insight into the implementation of the architectural components within the *askMe!* system. As an example, the different views (editors) of the presentation layer of the adaptation modeling and the adaptive testing engine component were presented and explained. The presentation layer of the adaptation modeling component consists of the test editor, the rule editor and the test assignment editor. In contrast, the presentation layer of the adaptive testing engine component comprises the knowledge dashboard, user and test statistics as well as the test environment.

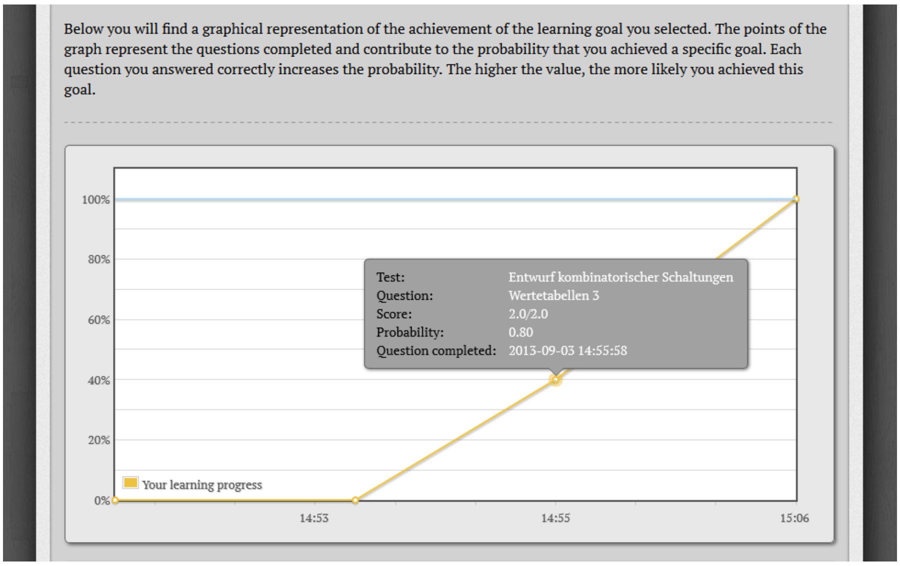


Figure 5.6: askMe! knowledge dashboard (learning goals)

5.4 SUMMARY

This chapter presented the implementation of the architectural model proposed in Chapter 4 by the web-based e-assessment system *askMe!*. The *askMe!* system covers the whole life-cycle of interactive and personalized e-assessments starting from creating questions and group them to (adaptive) tests, presenting it to the students up to preparing the results and presenting them to the teachers. Furthermore, the author of the adaptive tests is not limited to traditional question types such as multiple-choice or hotspot, but can make use of ICOs to create real interactive e-assessments.

Section 5.2 has briefly described the main technologies used for realizing the *askMe!* system. The open-source general-purpose server-side scripting language PHP in combination with open-source web application framework CakePHP were used as basis. In addition, for structuring and presenting content, HTML and CSS were used. Moreover, in order to give the system an attractive and easy to use GUI, the JavaScript library jQuery as well as the AJAX technology was used.

Following on this, Section 5.3 has given an insight into the implementation of the architectural components. Exemplary, the different views of the presentation layer of the adaptation modeling and the adaptive testing engine component were presented and explained. Details about the implementation of the remaining components can be found in the Appendix E.

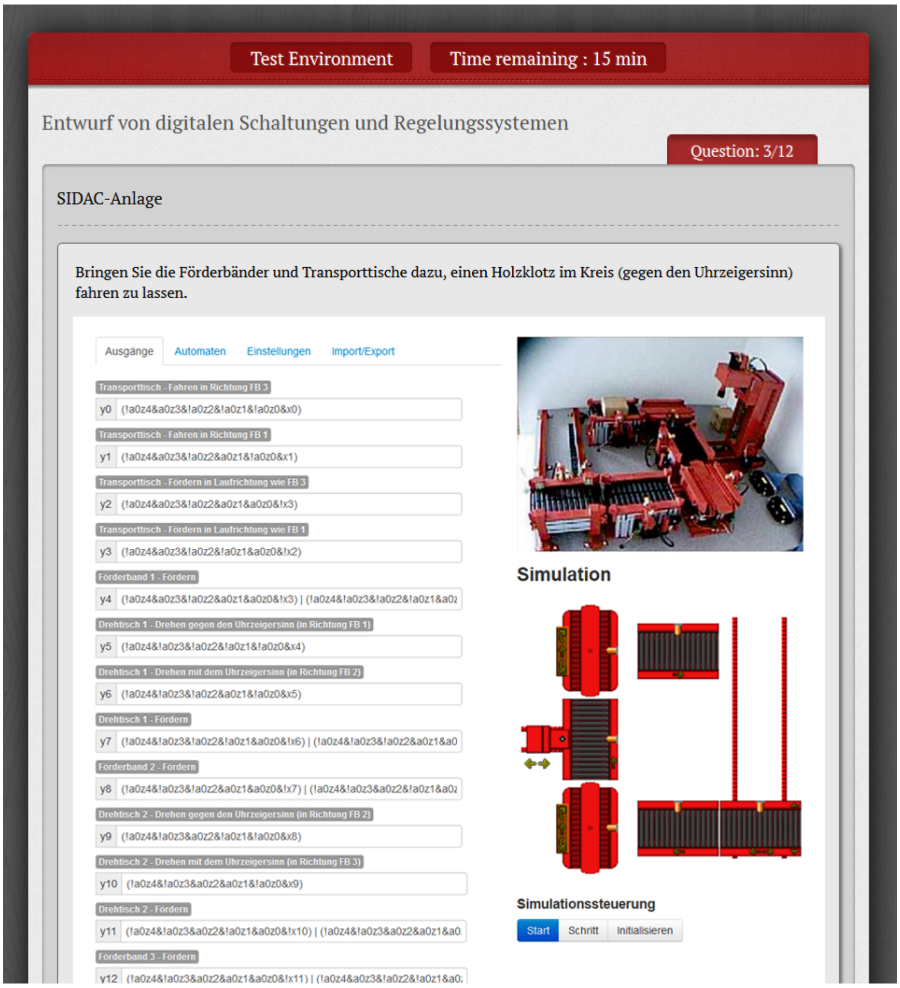


Figure 5.7: askMe! test environment

6 EVALUATION

6.1 INTRODUCTION

This chapter focuses on evaluating the usability, user experience and educational benefits of the *askMe!* system. These evaluation objects were investigated in two user studies. Section 6.2 presents the first user study mainly addressing the authoring process of adaptive tests. In contrast, Section 6.3 gives an insight into the second user study more focusing on the educational benefits of the system. Finally, Section 6.4 summarizes the main results. Parts of this chapter were already published in [SW13c].

6.2 FIRST USER STUDY

6.2.1 *Introduction*

The first user study focused on evaluating the usability and user experience of the *askMe!* system. Due to the fact that the system allows creating, editing and presenting interactive and personalized questions and tests, the authoring process of adaptive tests was in focus of this study. First of all, Section 6.2.2 describes the objects under investigation. Then, Section 6.2.3 gives an insight into the design of the study (i.e., test persons, data gathering methods). Section 6.2.4 looks into the different stages of the study and Section 6.2.5 analyzes its resulting data. Finally, Section 6.2.6 summarizes the main results.

6.2.2 *Objects under Study*

The objects under investigation in this user study were:

- Usability
- User Experience

Usability is defined by ISO 9241-11 [ISO98] as the *"extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use"*. The evaluation of usability aims at giving feedback whether the functional requirements of the *askMe!* system are implemented in a way that allows users accomplishing them efficiently and effectively. This also directly influences the user acceptance and satisfaction. In contrast, user experience is defined by ISO 9241-210 [ISO10] as a *"person's perceptions and responses that result from the use and/or anticipated use of a product, system or service"*. This includes all the emotions, beliefs, preferences, perceptions, physical and psychological responses, behaviors and accomplishments that occur before, during and after use (cf. Figure 6.1). Summarized it can be stated that usability includes pragmatic aspects (i.e., getting a task done), whereas user experience focuses on users' feelings resulting from both pragmatic and hedonic aspects of the system.

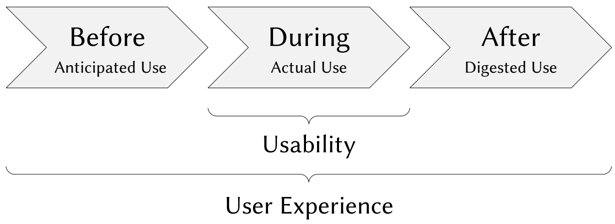


Figure 6.1: Comparison of usability and user experience

6.2.3 *Design*

6.2.3.1 *Test Persons*

The aim of this study was getting feedback about the usability and user experience of the authoring process of adaptive tests in the *askMe!* system. Consequently, teachers or people with teaching context were selected as test persons because the authoring tools of the system are intended to be used by those people. Moreover, a requirement for the persons was that they are familiar with computers and should at best have already experience with e-learning and/or e-assessment systems or tools. Eight voluntary test persons were acquired at the Fraunhofer Institute for Digital Media Technology and the Ilmenau University of Technology. According to Dumas and Redish [DR99], six to twelve test persons are sufficient to uncover about 80-90% of all usability problems that users are likely to have with a product.

6.2.3.2 *Data Collection*

In general, data collection is an important aspect of any type of research or user study. Inaccurate data collection can impact the results of a study and lead to invalid results. Data collection methods for evaluation can be grouped into two basic categories:

- *Quantitative methods*: These methods mostly produce hard numbers.
- *Qualitative methods*: These methods mostly produce words, but can also include photos, videos, audio recordings and other non-text data.

Examples of quantitative methods are *surveys*, *questionnaires* and *statistical analyzes*, whereas *interviews*, *focus groups* and *thinking aloud* are examples of qualitative methods. In order to obtain authentic and useful results, both quantitative as well as qualitative data collection methods were chosen in this user study. The methods that were used are:

- *Questionnaires*: This method consists of a number of questions that the user has to answer in a given format.
- *Thinking aloud*: This method involves users to verbalize their ideas, beliefs, explanations, doubts and observations during their use of the system under test.

Questionnaires were used in this user study because they allow collecting data from a large user group in a short period of time. In addition, the results are gathered in a standardized way, can usually quickly and easily be quantified and are more objective than, for example, interviews. In addition to questionnaires, the thinking aloud method was used because it get to know what the test persons think, expect and feel while performing the predefined tasks. The test persons were asked to think out loud, whereas an observer objectively took notes of everything that the test persons said. It is important to note that the observer did not attempt to interpret their actions and words. This method allows making explicit what is implicitly present in test persons performing a specific task.

6.2.4 *Procedure*

6.2.4.1 *Setting and Hardware*

The user study was carried out at the Fraunhofer Institute for Digital Media Technology in Ilmenau, Germany. The PC on which the evaluation was performed is equipped with an Intel(R) Core(TM)2 Duo CPU T8300 at 2.40GHz and 4GB of RAM running on Microsoft Windows 7 Enterprise. The resolution of the monitor is 1440 x 900 pixels and has a visible diagonal of 14,1 inch. Finally, accessing the web-based system was done by using the Mozilla Firefox web browser (ESR 17.0.7). Due to the fact that it was performed under controlled conditions and in an uninterrupted environment, this study is also called a *lab study*. The test persons were greeted by a researcher who introduced to the system and the goal of the study. The researcher also served as the observer for the thinking aloud method.

6.2.4.2 *Tasks*

After the brief introduction, the test persons were required to carry out specific tasks. These tasks are carefully prepared tasks that are typical of the tasks, for which the system was designed. This includes showing creating questions, grouping them to adaptive tests and showing user statistics. The sequence of the different tasks was fixed and the test persons had to stick to them. According to the thinking aloud method, the test persons were asked to verbalize what they think, expect and feel while performing the predefined tasks. The observer not only made detailed notes of test persons' interactions and what they said, but also measured the time they took in completing the tasks. The entire list of tasks can be found in Appendix G.

6.2.4.3 *Questionnaires*

After the test persons completed the tasks, they were asked to fill out four questionnaires:

- E-learning experience questionnaire
- ISONORM 9241/110 questionnaire
- User Experience Questionnaire (UEQ)[1]
- Demographic questionnaire

The first questionnaire asks users about their experience with e-learning, the average usage time of computers during working time as well as the general interest in new and innovative Information and Communications Technology (ICT). This data helps gauging the test persons' level of expertise in dealing with computers and e-learning software in particular. The second questionnaire is the extended (long) version of the ISONORM 9241/110 questionnaire [PSPT06]. It addresses seven dimensions (i.e., suitability for the task, self-descriptiveness, conformity with user expectations, suitability for learning, controllability, error tolerance and suitability for individualization) which are in accordance with ISO 9241-110 [ISO06]. The items for the last dimension were removed because individualization was out of focus in the design and implementation of the *askMe!* system. Each of the six dimensions consists of five statements (except suitability for the task, which consists of only four statements) that need to be judged on a numeric rating scale of 1 (+++) to 7 (−−−). Originally, the long version of the ISONORM requires test persons to give an example where the system breaches any of the aspects mentioned before (e.g., controllability). However, this input was omitted in this study because of time reasons. This questionnaire allows getting a total value for the usability of the *askMe!* system. As third questionnaire, the German version of the UEQ was chosen, which allows evaluating the user experience of interactive software such as the *askMe!* system. The format of the questionnaire supports users to immediately express feelings, impressions and attitudes that arise when using the software. The scales of the questionnaire cover a comprehensive impression of user experience (i.e., attractiveness, efficiency, perspicuity, dependability, stimulation and novelty) and result in 26 pairs of contrasting attributes. Each pair of attributes has seven circles in between representing graduations between the opposites. The order of the positive and negative attributes for an item is randomized in the questionnaire. Per dimension half of the items start with the positive and half with the negative term. The test users can express their agreement with the attributes by ticking the circle that is most closely. The last questionnaire is rather short and addresses the demographic of the test persons. It includes questions such as test persons' age, gender and profession. This data supports the analysis of the test persons' results with respect to usability and user experience. All questionnaires can be found in Appendix G.

1 www.ueq-online.org/

6.2.5 *Data Analysis*

Demographics

The data collection took place from 17.07.2013 until 22.08.2013 on the premises of the Fraunhofer Institute for Digital Media Technology in Ilmenau. In total, eight test persons took part in the study. All tests persons were between 25 and 35 years old, the average age was 30,25 (std.dev. = 3,5 years). Six male and two female participants took part in the study and each of them holds an academic degree.

E-Learning Experience and Attitude towards ICT

After a short introduction, the test persons carried out the specified tasks and at the end, they filled out the questionnaires. The average time for performing the tasks was 28min : 47s (std.dev. = 5min : 10s). The first four questions addressed the experiences with e-learning as well as the attitude towards ICT. With respect to question 1, half of the test persons characterized their knowledge about e-learning as good, one has average knowledge and three less knowledge. The question about the experience with other e-learning or e-assessment was answered equally. Four test persons have no experiences, whereas the remaining persons have already experiences with systems such as Questionmark Perception (cf. 3.3.3.1) or Moodle (cf. 3.3.3.5). Question 3 asked the test persons about their interests in new and innovative ICT. Six test persons stated a strong and two persons an average interest in ICT. Question 4 asked for the average usage time of computer during working time. Almost all (seven test persons) indicated that they use the computer very often and one test person stated a frequent use. Summarized it can be stated that the test persons correspond to those that were envisaged in the study design.

Usability

Following up these initial questions, the questions addressing the main objects under investigation in this study (i.e., usability and user experience) were presented to the students. As mentioned before, the usability of the *askMe!* system was measured using the ISONORM 9241/110 questionnaire. As shown in Table 6.1, the consistency reliability (measured using Cronbach's Alpha [Cro51]) of the different scales (except for suitability for the task and learning) given the small amount of test persons can be considered as sufficiently high. The results presented in the following can thus be considered as reliable. The low reliability value for the scales suitability for the task and learning can be explained by the fact that the items of the respective scale do not represent parallel measures of the construct. A factor analysis showed that the items of the scale suitability for learning constitute two and the items of the scale suitability for the task three components/constructs. Another reason for these low reliability values could also be the fact that the items of the scales were interpreted by several test persons in an unexpected way.

Table 6.2 and Figure 6.2 show the mean and standard deviation value for each scale. It can be seen that the mean value is always between 3 and 2, which means a uniform positive agreement to the items of the respective scale. In addition, the lower standard deviation values indicate that the data is clustered closely to the mean value. A detailed analysis of the different items of each scale can be found in Appendix G.

SCALE	ITEMS	RELIABILITY
Suitability for the task	4	0,18
Self-descriptiveness	5	0,52
Conformity with user expectations	5	0,58
Suitability for learning	5	0,37
Controllability	5	0,92
Error tolerance	5	0,89

Table 6.1: Reliability of the usability scales

SCALE	N	MEAN	STD. DEV.
Suitability for the task	8	2,03	0,59
Self-descriptiveness	8	2,20	0,47
Conformity with user expectations	8	2,43	0,74
Suitability for learning	8	2,08	0,52
Controllability	8	2,38	0,88
Error tolerance	8	2,63	1,06

Table 6.2: Mean and standard deviation value per usability scale

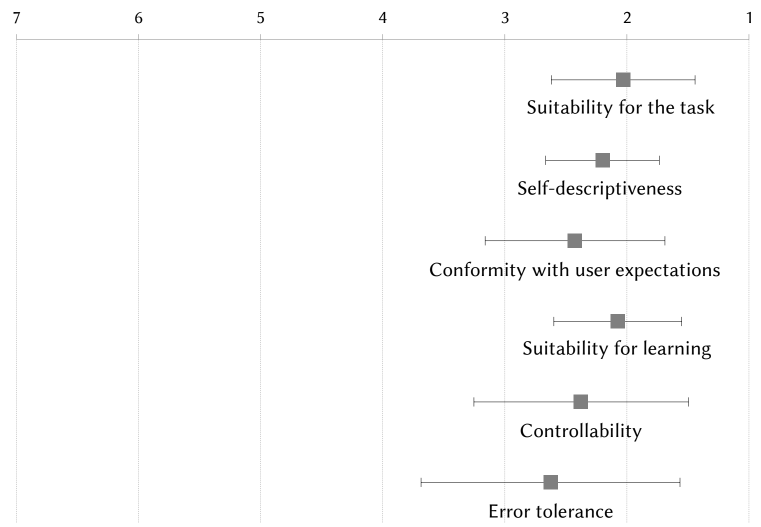


Figure 6.2: Mean and standard deviation value per usability scale

User Experience

The user experience from the authors' point of view was measured using the UEQ. Table 6.3 gives an overview about the internal consistency reliability of the different scales. With the exception of the second scale, all scales can be considered as sufficient consistent. However, the items of the scale perspicuity need to be interpreted very carefully. In contrast to the usability questionnaire, UEQ has a rating scale between -3 (−−−) and +3 (+++). Table 6.4 lists and Figure 6.3 depicts the mean and standard deviation values for each user experience scale. It can be noticed that all mean values are between 0,5 and 2. Against the background of different test persons with different opinions and answer tendencies, the values for the scales look from the purely visual standpoint on a scale range of -3 to +3 not as positive as they really are. A more detailed look at the different scales and their respective items is provided in Appendix G.

SCALE	ITEMS	RELIABILITY
Attractiveness	6	0,81
Perspicuity	4	0,13
Efficiency	4	0,48
Dependability	4	0,68
Stimulation	4	0,65
Novelty	4	0,86

Table 6.3: Reliability of the user experience scales (first user study)

SCALE	N	MEAN	STD. DEV.
Attractiveness	8	1,35	0,59
Perspicuity	8	1,81	0,40
Efficiency	8	1,19	0,62
Dependability	8	1,59	0,55
Stimulation	8	0,97	0,80
Novelty	8	0,53	1,01

Table 6.4: Mean and std. deviation value per user experience scale (first user study)

Author Comments

While the test persons performed the different tasks, they said their ideas, doubts and observations out loud. These notes and comments were analyzed and differentiated according to their characteristics (i.e., positive comment, negative comment or suggestion for improvement), but also by the task/-component (e.g., user modeling component) they address. The whole list of comments can be found in Appendix G. The most comments are suggestions

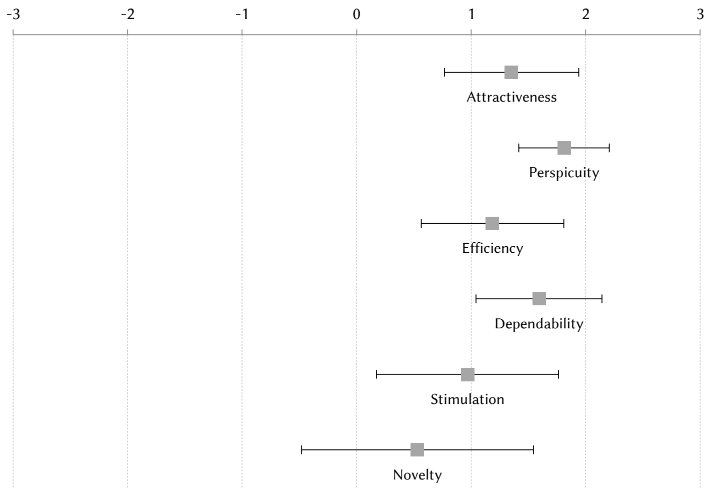


Figure 6.3: Mean and std. deviation value per user experience scale (first user study)

for improvement, however, there are also more than ten positive and only a few negative comments.

6.2.6 *Summary*

This section has presented the design, the procedure and the data analysis of the first user study. The aim of this study was getting feedback about the usability and user experience of the authoring process of adaptive tests in the *askMe!* system. Consequently, teachers or people with teaching context were selected as test persons. They were required to carry out specific tasks that are typical of the tasks, for which the system was designed. In order to obtain authentic and useful results, both quantitative (questionnaires) as well as qualitative data collection methods (thinking aloud method) were applied in this study. After the test persons completed the tasks, they were asked to fill out four questionnaires (i.e., e-learning experience questionnaire, ISONORM 9241/110 questionnaire, user experience questionnaire and a demographic questionnaire). Summarized it can be stated that usability and user experience were rated very well by the test persons in this study. The overall results are good, but leave room for further developments/improvements of the *askMe!* system from the authors' point of view. The comments derived from the thinking aloud method will also contribute to that objective.

6.3 SECOND USER STUDY

6.3.1 *Introduction*

In contrast to the first user study, the second user study mainly focused on evaluating the educational benefits of the *askMe!* system. For that reason, it was tested in a real-life setting at the Ilmenau University of Technology. The first ideas of this evaluation were already presented in [SS12]. First of all, Section 6.3.2 describes the objects under investigation. Then, Section 6.3.3 gives an insight into the design of the study, before Section 6.3.4 looks into the different stages of the study. Section 6.3.5 analyzes its resulting data and finally, Section 6.3.6 summarizes the main results.

6.3.2 *Objects under Study*

The objects under investigation in this user study were:

1. Learning support
2. User Experience

The first evaluation object, referred to as *learning support*, is defined by the author of this thesis as the ability of the system to support students' learning by providing them an increased awareness of their strengths and weaknesses in order to get the most out of their learning. The (subjective) awareness will not only be raised by *run-time* support during testing, for example, in the form of an adapted question sequence that is better suited to prior knowledge or the presentation of individualized feedback (hints) that helps in a specific situation, but also by providing students an all the time, easily accessible, tailored and easy to understand reporting on their individual progress. The second evaluation object is the same as in the first user study and focused on students' feelings, impressions and attitudes resulting from using the *askMe!* system.

6.3.2.1 *Research Question*

In order to make reliable statements about the learning support of the *askMe!* system, it needs to be made measurable. While the user experience of the *askMe!* system can be measured using standardized questionnaires such as the UEQ, with regard to learning support this is much more complicated. In this user study, learning support was intended to be measured by the results of the final examination. The ideal case would be: the more the student uses the *askMe!* system, the better is its exam result. In order to verify that, the following empirical research question for this user study has been formulated: *"What is the effect of using the askMe! system on students' exam results?"*

HYPOTHESIS	NULL HYPOTHESIS
H1: The more frequently the *askMe!* system is used, the better is the exam result.	**H0:** The frequency of use of the *askMe!* system has no effect on the exam result.
H2: The more extensive the *askMe!* system is used, the better is the exam result.	**H0:** The extent of use of the *askMe!* system has no effect on the exam result.
H3: Students that use the *askMe!* system have more stable exam results.	**H0:** The use of the *askMe!* system has no effect on the stability of the exam results.

Table 6.5: Research hypotheses

6.3.2.2 *Empirical Research Model*

In order show the interrelationships between the variables related to the main object under investigation in this study (i.e., learning support), an empirical research model has been developed (cf. Figure 6.4). As depicted, the use of the *askMe!* system is defined as an *independent* variable and is predicted to have an impact on students' exam results (*dependent* variable). It is also expected that students' demographics, their learning motivation and attitude towards ICT as well as the examiner have an influence on the exam result. They are referred to as intervening variables because they support a better analysis of the relationship between the independent and dependent variables when the variables appear to not have a definite connection.

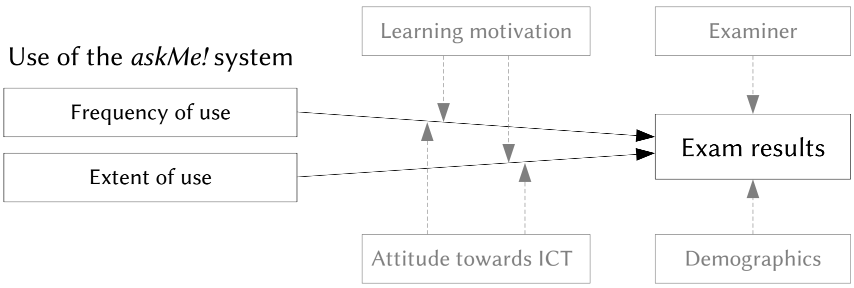


Figure 6.4: Empirical research model

6.3.2.3 *Hypotheses*

Based on the aforementioned research model and the main (empirical) research question, several hypotheses were formulated. The testing as well as the comparison groups served as basis for the hypotheses listed in Table 6.5. Each hypothesis is accompanied by a null hypothesis that says that there is no relationship between the independent and dependent variables.

6.3.3 *Design*

6.3.3.1 *Test Persons*

The aim of this study was getting feedback about the ability of the system to support students' learning as well as about the user experience from the students' point of view. As test persons, students from the digital systems design course at the Ilmenau University of Technology were selected. The number of registered students for this course was about 80. The objectives of the course are getting acquainted with basic knowledge in the field of computer engineering, the design of digital systems and the understanding of computer internal information processing, computer architectures and organizations. Generally, the course consists of a lecture and an accompanying seminar. While in the lecture primarily theoretical basics are taught, the seminars focus on exercising the application of the theoretical foundations. The course is completed with a final oral exam. Over the last years, it could be recognized that students had problems with specific tasks in the exams. This concerned in particular the application of knowledge in design, analysis and synthesis tasks. This evaluation took up these findings and specifically addressed them in this study. In doing so, the *askMe!* system was made available for students preparing for their final exams (formative self-assessment, cf. Section 2.2.4.1). The course consisted of three seminar groups from which one was randomly selected as testing group, the other two groups represented the comparison groups. All students of the testing group were informed about and got access to the *askMe!* system. This sampling technique is also called as *cluster sampling*. Due to the fact that the seminar groups were also selected randomly, the testing group is a small scale representation of the total population.

6.3.3.2 *Data Collection*

In order explain the causal connections presented in the research model (cf. Figure 6.4), this study also made use of different data collection methods. The methods that were used are:

- *Questionnaires*: This method consists of a number of questions that the user has to answer in a given format.
- *Data log analysis*: This method automatically records users' behavior (e.g., date/time of access or actions performed) during the system under test.
- *Document-based analysis*: In this method, the expert uses existing checklists or other documents in addition to his/her own judgment.

Questionnaires were used in this user study in the same way as in the first user study, to collect data from a large user group in a short period of time. In addition to questionnaires, the data log analysis method was used because it allows making conclusions about students' frequency and extent of use of the *askMe!* system. The documents that were used and analyzed in the third data collection method are students' exam results. In doing so, the exam results of the testing group were compared with that of the comparison groups as well as the overall results with that of the last semesters.

6.3.4 *Procedure*

6.3.4.1 *Setting*

In contrast to the first user study, this user study was carried out as a *field study*, which means that it was not performed in an artificial setting such as in a *lab study*. Instead, the students were freely to decide when, where and using which device they are going to use the system. Due to the fact that the system is web-based, the students only needed a web browser to access the system. The system was made available five weeks before the first examination date. In this way, both students that learn in a long-term as well as students that learn in a short-term manner could be served. In addition, the availability of the system over this long period of time also increases the validity and reliability of the evaluation results.

6.3.4.2 *Questionnaires*

After the students passed their final exams, they were asked to fill out five questionnaires:

- ICT attitude questionnaire
- Learning motivation questionnaire
- *askMe!* questionnaire
- User experience questionnaire
- Demographic questionnaire

The user experience and the demographic questionnaire are more or less the same as in the first user study (cf. Section 6.2.4.3). The ICT questionnaire [WWM12] assesses students' attitude towards ICT and experience with it. It consists of six scales:

- *Exploration*: This scale measures the degree how analytical vs. exploratory individual interfaces and functions are acquired.
- *Skepticism*: This scale measures the degree of mistrust in ICT concerning financial activities.
- *Anxiety*: This scale measures the degree of insecurity and anxiety towards technology.
- *Interest*: This scale measures the degree of interest and fascination towards new technology.

- *Competence*: This scale measures the degree of expertise, experience and technical affinity towards ICT.
- *Surface*: This scale measures the degree of importance of brands, appearance and design.

Based on the answers to items associated to these scales, the questionnaire allows categorizing users as *anxious*, *trust-guided*, *interested amateur*, *pragmatic inspired*, *experienced*, *playful* or *gadget loving* ICT users. The learning motivation questionnaire includes four questions about the general attitude towards learning. The questionnaire about the *askMe!* system includes a general assessment in terms of robustness, navigation and accessibility, but also includes questions about how long and how often the students have used the system, whether the system was helpful, is desirable for use in other courses, etc. All the questionnaires mentioned before require the students to judge the respective item on a five-point Likert scale, whereas 1 means strongly agree (++) and 5 totally disagree (−−). All questionnaires can be found in Appendix H.

6.3.5 *Data Analysis*

The data collection took place from 26.06.2013 until 12.09.2013. In total, 31 students took part in the study, from which 21 students actively used the system for exam preparation. The average usage time of the *askMe!* system was about two hours and the average number of system logins was about three. All in all, 101 tests were completed and 714 questions were answered (including repetitions) by the different students over a period of 12 weeks. The analysis of the usage statistics showed that only a few students tested and used the system after they got login accounts, instead the most students used the system shortly before the examination date. All tests persons were between 19 and 25 years old, the average age was 21 (std.dev. = 1,34 years). Two female and 29 male participants took part in the study and each of them studied computer science for engineers.

Demographics

Independent and dependent Variables

First of all, the relationship between the independent and dependent variables was analyzed (cf. Figure 6.4). In order to determine how often and how intensively the students used the system for test preparation, the data logs of the *askMe!* system under test were investigated. This includes the number of logins, the approximate period of use as well as the amount of questions and tests completed. The *frequency of use* was determined by the number of logins, whereas the *extent of use* was determined by the number of questions completed. Both independent (sub-) variables were subject of a correlation analysis. However, no correlation between the frequency/extent of use and students' exam result could be identified. Hence, the first two hypotheses (cf. Table 6.5) have not been confirmed. However, having a look at the overall exam results, it could be notice that the average grade of students that used the *askMe!* system for test preparation (n = 21; average grade = 1,81) is much better than students who did not use the system (n = 60; average grade =

2,59). In addition, the hypothesis that students who use the *askMe!* system have more stable exam results turn out to be true. The analysis of the exam results showed that the standard deviation value of students that used the system was about 1,00 and the value of students who did not was about 1,38. In order to prove whether these results are statistically-robust, a significance test (t-test) was carried out. It confirmed that there is significant difference between both groups (n = 41; mean difference = 0,779; degree of freedom = 39; significance = 0,030). For that reason, this group can be considered as a representative sample. In addition, similar results can be seen when looking at the failure rates. The failure rate of students that did not use the system was about 19%. However, the failure rate of students that used the system for test preparation was about 5%. This is a remarkable difference, especially from the fact that the overall failure rate (n = 81; failure rate = 15%) is higher than those of the previous year (n = 63; failure rate = 12,4%).

Intervening Variables

The intervening variables were also analyzed in order to determine whether they have a connection to the independent and dependent variables. As a result, a relationship between the answer to the question *"I prefer content that arouses curiosity even it is difficult to learn"* (m2) and the exam result were found. This means, the higher the students agreed on this statement, the better were their exam results (n = 25; r = 0,476; p = 0,016). Due to the few significant correlations between the different variables, the initial correlation analysis was followed by a descriptive analysis of the questionnaire results.

Attitude towards ICT

The first questionnaire aimed at assessing students' attitude towards ICT. The analysis of the results showed that the students were mostly *experienced* users (23x). In addition, five students can be categorized as *playful* and one students as *interested amateur*. The remaining two students did not provide any data. Experienced users belong to the classes with highest competence, exploration and interest in combination with lowest anxiety and skepticism.

Learning Motivation

Following the ICT questionnaire, the students were asked to express their motivation towards learning. Here, it could be noticed that the majority of test persons is open to challenging and difficult content if something new can be learned or the content arouses curiosity. This confirms the need for personalization as provided by the *askMe!* system.

askMe! Questionnaire

The next items in the questionnaire concerned the *askMe!* system itself. It was very pleasing to see that the test persons almost always responded very positive to the items in this questionnaire. A more detailed look at the different questionnaire results is provided in Appendix H.

User Experience

Just as in the first user study, the user experience (even from the students' point of view in this respect) was measured using the UEQ. Table 6.6 gives an overview about the internal consistency reliability of the different scales. In comparison with Table 6.3, it can be seen that reliability value of attractiveness is almost the same in both user studies. In contrast, the other scales have slightly different values. For example, perspicuity has a substantially higher value, whereas the reliability value of efficiency has fallen by half. Table 6.7 lists and Figure 6.5 depicts the mean and standard deviation values for each scale. Even though some values are slightly below the ones obtained from the first user study, the overall result is quite good. Important to note is the fact that the students rated the system as more exciting, faster, inventive and innovative than the authors. Due to the fact that the number of data sets is more than twice as much, this result is much more significant. More details about the different scales and their items can be found in Appendix H.

SCALE	ITEMS	RELIABILITY
Attractiveness	6	0,80
Perspicuity	4	0,38
Efficiency	4	0,21
Dependability	4	0,46
Stimulation	4	0,48
Novelty	4	0,76

Table 6.6: Reliability of the user experience scales (second user study)

SCALE	N	MEAN	STD. DEV.
Attractiveness	21	1,08	0,66
Perspicuity	21	1,21	0,70
Efficiency	21	0,89	0,63
Dependability	21	1,14	0,67
Stimulation	21	0,81	0,67
Novelty	21	0,55	0,86

Table 6.7: Mean and standard deviation value per user experience scale (second user study)

Student Comments

After the test persons filled out the questionnaires, they had the opportunity to leave comments on the *askMe!* system (e.g., problems encountered or desirable improvements). Some of the notes and comments were implemented immediately (e.g., spelling errors), others need more development efforts and were integrated into future development activities (e.g., math/formula editor or automatic test generation). Moreover, the students really appreciated the existing questions and test, however, they mentioned that they would like to have some more.

6.3.6 *Summary*

This section has presented the design, the procedure and the data analysis of the second user study. The aim of this study was getting feedback about the

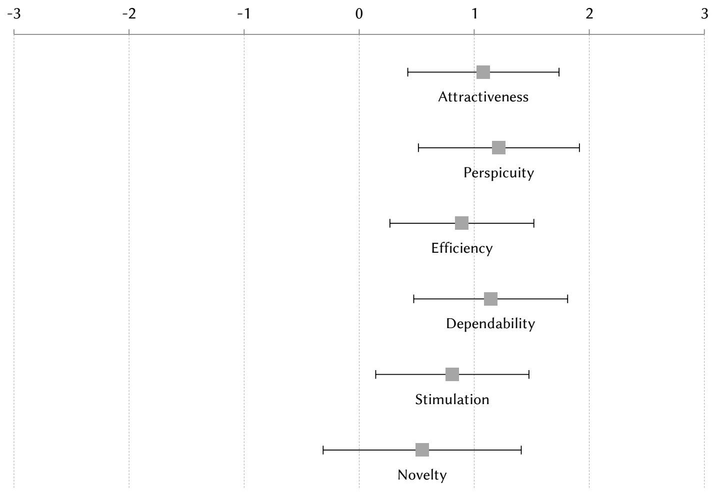


Figure 6.5: Mean and standard deviation value per user experience scale (second user study)

ability of the *askMe!* system to support students' learning as well as about the user experience from the students' point of view. Consequently, the system was tested in a real-life setting at the Ilmenau University of Technology in a course called digital systems design, which is provided for bachelor students in the fourth semester. In doing so, the system was made available for students preparing for their final exams. They were freely to decide on which test they could start first or which tests they re-try or leave out. In order explain the causal connections presented in the research model, the study made use of different data collection methods (i.e., questionnaires, data log analysis and document-based analysis). As a result, 31 test persons took part in the second study, whereas 21 actively used the system for test preparation. Unfortunately, no correlation between the frequency/extent of use and students' exam result could be identified. However, the average grade of students that used the *askMe!* system for test preparation is much better than students who did not use system. Moreover, it is also remarkable that the failure rate of students that did not use the system was four times higher than students that used the system for test preparation. Just as in the first user study, the user experience was rated very well by the test persons. These results as well as the notes/comments given by students serve to further improve and strengthen the functioning of the *askMe!* system.

6.4 SUMMARY

This chapter described two user studies performed to evaluate the usability, user experience and educational benefits of the *askMe!* system. The first user study focused on the usability and user experience of the authoring process of adaptive tests. Consequently, people with teaching context were selected as test persons. They carried out specified tasks that are typical of the tasks, for which the system was designed (e.g., creating questions, grouping those to adaptive tests and showing user statistics). After they completed the tasks, they were asked to fill out four questionnaires (i.e., e-learning experience questionnaire, usability questionnaire, user experience questionnaire and a demographic questionnaire). In total, eight test persons (six male and two female) took part in the study and each of them holds an academic degree. The responses to the e-learning experience questionnaire show that all test persons are interested in ICT and that half of the test persons already had experiences with e-learning or e-assessment systems. Thus, the reliability of results of the subsequent usability and user experience questionnaire could be considered as sufficient high. In order to rate the usability of the *askMe!* system, the ISONORM 9241/110 questionnaire was used. It addresses seven dimensions which are in accordance with ISO 9241-110. As a result, a uniform positive agreement to the items of the different scales could be noticed. Important to note is that the scales *suitability for the task* and *suitability for learning* show the best ratings and thus mostly influence the usability rating of the system. The user experience from the authors' point of view was measured using the UEQ. The scales of the questionnaire cover a comprehensive impression of user experience (i.e., attractiveness, efficiency, perspicuity, dependability, stimulation and novelty) and result in 26 pairs of contrasting attributes. The results showed that the user experience from the authors' point of view is rated as very well. Besides, it has been seen that the scale *perspicuity* has the highest influence on the user experience of the *askMe!* system. In addition to the different questionnaires, the thinking aloud method was used in order to get to know what the test persons think, expect and feel while performing the predefined tasks. The notes and comments obtained from this method will be used for further developments and improvements of the *askMe!* system.

The second user study focused on the user experience from the students' point of view, but also on evaluating the educational benefits (learning support) of the *askMe!* system. For that reason, the system was tested in a real-life setting at the Ilmenau University of Technology in a course called digital systems design for bachelor students. The system was made available for students preparing for their final exams and they could freely decide to use the system and to what extent. In total, 31 students (29 male and two female) took part in the study, from which 21 students actively used the system for exam preparation. All in all, 101 tests were completed and 714 question were answered (including repetitions) by the different students over a period of 12 weeks. The analysis of the usage statistics showed that only a few students tested and used the system after they got login accounts, instead the most students used the system shortly before the examination date. After the final exam, the test persons had to fill out a questionnaire that aimed at identifying students' attitude towards ICT, their individual learning motivation as well as their opinions about the robustness, navigation and accessibility of the *askMe!* system. The analysis of the ICT questionnaire results showed that the students were mostly experienced users. Experienced users belong to the classes with highest competence, exploration and interest in combination with lowest anxiety and skepticism. With respect to the learning motivation, it could be noticed that the majority of test persons is open to challenging and difficult content if something new can be learned or the content arouses curiosity. This confirms the need for personalization as provided by the *askMe!* system. In addition to the questionnaires, the second user study also made use of a data log and document-based analysis method. This allowed investigating a potential relationship between the use of the *askMe!* system and students' exam results. Unfortunately, no correlation between the frequency/extent of use and students' exam result could be identified. Thus, the first two research hypotheses were not confirmed. However, the average grade of students that used the *askMe!* system for test preparation is almost one point better than students who did not use system. Furthermore, the hypothesis that students who use the *askMe!* system have more stable exam results turn out to be true. The standard deviation value of students that used the system was much lower than of students who did not. Moreover, also important to note is the fact that the failure rate of students that did not use the system was four times higher than students that used the system for test preparation. A statistical-robust difference between both groups could be determined by a significance test (t-test). In addition to the learning support of the *askMe!* system, students' feelings, impressions and attitudes resulting from using the system (user experience) was also evaluated. Just as in the first user study, the user experience was rated very well by the test persons. Both ratings show considerable similarities, for example, the scale *perspicuity* also has the highest influence on the user experience followed by the scales *dependability* and *attractiveness*. These results as well as the notes/comments given by students will also serve to further improve and strengthen the functioning of the *askMe!* system.

Summarized it can be stated that the architectural model and its implementation (*askMe!* system) is well applicable for its intended use, both in terms of educational benefits as well as in terms of usability and user experience.

7 CONCLUSION

7.1 INTRODUCTION

This chapter summarizes the work carried out within this thesis. Section 7.2 discusses the goals of this thesis and how they were achieved. Following up on this, Section 7.3 identifies the contribution this work has made to the state-of-the-art of (personalized) e-assessment systems. Finally, Section 7.4 concludes with a discussion of future work that describes research areas in which this work could be carried forward.

7.2 RESEARCH GOALS AND ACHIEVEMENTS

The research question driving this thesis was how interactive e-assessment can be enhanced with personalization aspects. In answering this question, the primary objective of this Ph.D. project was to provide an architectural model for implementing personalized and interactive e-assessment systems and tools that allow integrating and using interactive and immersive tools (e.g., simulations or animations) into questions and tests, and enable tailoring them to students' individual characteristics. It has been proposed that this objective could be achieved by the development of a novel approach for implementing advanced e-assessment systems able to consider the needs and characteristics of students and allow for more creativity in answering by interacting with digital tools in a variety of ways.

Architectural Model

The architectural model is based on three pillars: a consistent user modeling, a generic domain modeling and a flexible adaptation modeling, which represent the fundamental basis for the adaptive behavior. This approach advocates the separation of the elements of adaptivity into discrete models (i.e., User Model (UM), Domain Model (DM) and Adaptation Model (AM)) that are brought together at run-time through a generic and extensible adaptive testing engine. Additionally, the architectural model is complemented by a question modeling component responsible for representing questions and tests, responses, etc. The separate and discrete modeling of the different elements of the architectural model not only increases the maintainability and extensibility of the future e-assessment system, but also facilitate adaptability and customization through component exchange.

Question Modeling Component

One of the most significant components in the architectural model is the question modeling component because it is responsible for authoring questions and interactive tasks to be used in adaptive tests. The analysis of the state-of-the-art of established e-assessment standards and specifications has shown that the Question Models (QMs) provided lack opportunities to integrate and interact with interactive and simulative tools (Interactive Content

Objects (ICOs)) during e-assessment. As a result, a new QM called MICO has been designed. It enables describing and initializing ICOs as well as processing students' responses resulting from interacting with ICOs. Additionally, the processing of responses in turn requires a mechanism for passing information back and forth. Consequently, not only a QM, but also a Communication Schema (CS) and a Communication Mechanism (CM) has been proposed. Due to the fact that the proposed model and mechanism are based on established standards and specifications (i.e., IMS QTI and xAPI), they can also be applied to other e-assessment systems and tools with little effort.

Adaptation Modeling Component

Another key component in the architectural model is the adaptation modeling component because it is responsible for maintaining the AM. The AM can be regarded as the core of the adaptive system because it defines what can be adapted, as well as when and how it is to be adapted. In order to realize personalization, the AM allows tailoring the sequence, presentation and selection of questions, the question difficulty as well as the selection and presentation of feedback to the individual student. The information that can be used to provide adaptations encompass all information that are stored in the user profile (e.g., students' knowledge, interests, goals, etc.) as well as information, which are obtained during testing (e.g., score achieved, time required, etc.). For proposing and aiding decisions about alternatives, the AM uses the rule-based reasoning approach. The analysis of state-of-the-art has revealed that existing adaptive (e-assessment) systems lack the ability of the author to directly and easily influence the adaptation process. Due to the fact that rules provide this kind of flexibility as well as a high level of comprehensibility, they were used as basis for the reasoning approach.

In summary, the main goals of this Ph.D. project were:

1. Analyze how existing e-assessment concepts, methods and systems can support the design of such a model.

2. Design of a conceptual model that integrates personalization and didactic interactivity.

3. Apply the conceptual model by implementing it in an appropriate scenario.

4. Perform an evaluation to validate the educational benefits of the model and its implementation.

5. Examine how integration with established LMSs/LCMSs can be achieved.

The first goal aimed at identifying and surveying the concepts, methods and systems relevant to the design and implementation of the architectural model. A comprehensive analysis of the state-of-the-art in the field of e-assessment and e-learning has been performed in the course of this project. In particular, the knowledge about strengths and weaknesses of existing approaches as well as about established standards and specifications in this field has significantly influenced the architectural design and its implementation. The results have been presented in Chapter 3. The second and third goal has been achieved by the design of an architectural approach for interactive and personalized e-assessment systems and its implementation by the web-based e-assessment system *askMe!*. The system fully conforms to the model and allows creating, editing and presenting interactive and personalized questions and tests. The design and implementation of the overall model have been described in Chapter 4 and Chapter 5, respectively. The fourth goal was to validate the educational benefits of the model and its implementation. The achievement of this goal has been demonstrated by two user studies described in Chapter 6. As a result, it can be concluded that the model/system is well applicable for its intended use, both in terms of educational benefits as well as in terms of usability and user experience. Finally, it has been proven that the model and its implementation, respectively can successfully be integrated with established LMSs/LCMSs such as Moodle. The integration can also contribute to a prompt and widespread adoption of the architectural model. Details about the integration have been shown in Section 4.8.

7.3 CONTRIBUTION TO THE STATE-OF-THE-ART

The architectural approach, as an approach for the development of interactive and personalized e-assessment systems, is the primary contribution to the state-of-the-art made by this thesis and the work described in it. This approach is significantly different to that used in the development of current e-assessment systems.

Personalization

Firstly, the approach is based on an adaptive system consisting of a UM, DM and AM whose interplay allows tailoring the questions and tests to the individual students. This ensures that there is, whatever their starting point, continuity and progression at all stages of their learning process and that each of them meets their full potential. The UM enables representing a variety of user features ranging from user's knowledge to their individual characteristics (e.g., interests, goals or background), supports complex assumptions and reasoning and allows dealing with uncertain and imprecise information. The DM supports the creation of arbitrary domain structures by defining hierarchical or associative relationships between the concepts. The AM is different from current adaptive assessment systems in that it allows authors directly and easily influence the adaptation process by creating adaptation rules. In order to support the authors in creating the rules needed for the adaptive tests, a novel graphical adaptation modeling approach based on Finite State Machines (FSMs) has been developed. In addition, particular importance has been attached to feedback personalization because it not only plays an important role in the question and test evaluation, but is also essential when striving for the assessment of students' HOTS. In this context, a new three-dimensional feedback classification has been proposed.

Didactic Interactivity

Secondly, the approach is not limited to creating traditional question types, but allows creating ICO-enhanced questions/tasks. They do not require students to simply choose an answer from a list of options, but allow them to carry out experiments by manipulating data (i.e., real or virtual objects) and observing the effects of change, and create and test hypotheses. This significantly increases the quality and efficiency of assessment because it allows students to be assessed in the same environment in which they learn. It also allows combining theory and practice and sets the basis for sophisticated learning methods such as competency-based, task-directed or problem-based learning. Due to the fact that all interactions can be recognized by the system, the teacher is always informed about how the student reached this answer or solution. In this way, teachers are able to assess lower (LOTS), but also higher levels of knowledge (HOTS). This is a significant contribution to the state-of-the-art where almost all e-assessment systems are limited to the lower levels. This has been made possible by the design of a new QM and an accompanied CS and CM. The QM as well as all models integrated in the adaptation process are kept separate and discrete. The separation of the models and components is fundamental to the architectural approach and benefits the state-of-the-art by providing an approach that is highly reusable and extensible.

The influence of this approach on the state-of-the-art has been witnessed by its direct contribution to 13 peer-reviewed publications in national and international conferences and journals such as:

- International Conference on Computer Supported Education (CSEDU)
- International Conference on User Modeling, Adaptation and Personalization (UMAP)
- European Journal of Open, Distance and E-learning (EURODL)
- International Journal of Emerging Technologies in Learning (iJET)

The research and development performed in this thesis open up new opportunities for advanced e-assessment systems, so-called Assessment 3.0 systems, which are able to consider the needs and characteristics of students and allow for more creativity in answering by interacting with digital tools in a variety of ways. The combination of both aspects will move the purpose of assessment from assessment *of* learning to assessment *for* learning and can in turn improve the quality and outcomes of learning in general. However, interactive and personalized e-assessment is still a relatively young research field and it is hoped that the work described in this thesis will have a beneficial and significant impact upon it.

7.4 FUTURE WORK

Students Features

There are many areas in which the work described in this thesis could be taken forward. This concerns on the one hand the inclusion of new students' features. There are many different forms of personalization and characteristics of the student that may be adapted to. First ideas were already provided in [HSL14], where students' volition has been proposed as a beneficial feature for adaptations. The term volition is defined as the cognitive process by which an individual decides on and commits to a particular course of action. It is generally agreed [GMTR98, TNB+05] that the consideration and stimulation of students' volition has a positive effect on learning efficiency and that it plays a mediating role between the intention to learn (motivation) and goal-directed behavior (the use of learning strategies).

Learning Analytics

The next area for future work concerns the emerging field of learning analytics. It tries to measure, collect, analyze and report use data about students and their contexts for the purpose of understanding and optimizing their learning. It is generally recognized that it offers promising possibilities for education and assessment. A first study [SW14] has shown that the approach and its implementation already provides a decisive contribution to this field, however, it is recognized that there are much more potential and challenges to address. This includes, for example, analyzing student' performances using machine learning algorithms in order to predict which students are at risk of failure.

Authoring of Adaptivity

Another area for future work is the authoring of adaptivity. Although this thesis proposes a novel graphical adaptation modeling approach based on FSMs, there is still much work to be carried out. There are several tools to support authors in the creation of questions and tests, but there are only a few to aid them in the creation of adaptive tests. Such a tool should actively support the author in creating pedagogically sound tests. This could be done by giving recommendations based on the analysis of the students and their prior results, or by compiling new adaptive tests based on established test patterns.

Finally, there are many areas outside of traditional e-assessment/e-learning where the architectural approach could be applied. This includes any ICT system where users' individual knowledge or characteristics are of particular importance, and/or need to adapt accordingly (e.g., recommendation systems).

[AC01] L. Ardissono and L. Console. An adaptive system for the personalized access to news. *Artificial Intelligence Communications*, 14:129–147, 2001.

[ACW95] J.C. Alderson, C.M. Clapham, and D. Wall. *Language Test Construction and Evaluation.* Cambridge University Press, Cambridge, 1995.

[AD04] J. Aitchison and S. Dextre Clarke. The Thesaurus: A Historical Viewpoint, with a Look to the Future. *Cataloging & Classification Quarterly*, 37(3-4):5–21, 2004.

[AGK+09] Á. Agea, R.C. García, C.D. Kloos, I. Gutiérrez, D. Leony, and A. Pardo. ICOPER D6.1 - Analysis of existing specifications and standards for assessment and evaluation and their usage in Europe. Technical report, ICOPER, 2009.

[AKA+01] L.W. Anderson, D.R. Krathwohl, P.W. Airasian, K.A. Cruikshank, R.E. Mayer, P.R. Pintrich, J. Raths, and M.C. Wittrock. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives.* Addison Wesley Longman, Inc, 2001.

[Ala14] N. Alaqraa. *Design and Implementation of an Adaptation-Engine for a personalized E-Assessment System.* Master thesis, Ilmenau University of Technology, 2014.

[Ame06] A. Amer. Reflections on Bloom's revised Taxonomy. *Electronic Journal of Research in Educational Psychology*, 4(1):213–230, 2006.

[Ang95] T.A. Angelo. Reassessing (and Defining) Assessment. *AAHE Bulletin*, 48(3):7, 1995.

[Ast93] A.W. Astin. *Assessment for Excellence: The Philosophy and Practice of Assessment and Evaluation in Higher Education.* Oryx Press, 1993.

[Bat90] A.W. Bates. Interactivity as a Criterion for Media Selection in Distance Education. In *Proceedings of the Annual Conference of the Asian Association of Open Universities*, page 16, 1990.

[BC82] J.B. Biggs and K.F. Collis. *Evaluating the Quality of Learning: the SOLO taxonomy.* Academic Press Inc., New York, 1982.

[BC99] P. Brusilovsky and D.W. Cooper. ADAPTS: Adaptive hypermedia for a Web-based performance support system. In *Proceedings of Second Workshop on Adaptive Systems and User Modeling on WWW at the 8th International Word Wide Web Conference and 7-th International Conference on User Modeling*, pages 41–47, 1999.

[BE94] C. Boyle and A.O. Encarnacion. Metadoc: An Adaptive Hypertext Reading System. *User Modelling and User-Adapted Interaction*, 4(1):1–19, 1994.

[BEF+56] B.S. Bloom, M.D. Engelhart, E.J. Furst, W.H. Hill, and D.R. Krathwohl. *Taxonomy of Educational Objectives, Handbook 1: Cognitive Domain*. Longman, 1956.

[Ben98] R.E. Bennett. Reinventing Assessment. Speculations on the Future of Large-Scale Educational Testing. Technical report, Policy Information Center, 1998.

[BIO88] C.V. Bunderson, D.K. Inouye, and J.B Olsen. The Four Generations of Computerized Educational Measurement. Technical report, Educational Testing Service, 1988.

[BKI03] C. Beierle and G. Kern-Isberner. *Methoden wissensbasierter Systeme: Grundlagen, Algorithmen, Anwendungen*. Vieweg+Teubner Verlag, 2003.

[BM03] J. Bull and C. McKenna. *A Blueprint for Computer-assisted Assessment*. Routledge, 2003.

[BM07] P. Brusilovsky and E. Millan. *User Models for Adaptive Hypermedia and Adaptive Educational Systems*, chapter 1, pages 3–53. Berlin Heidelberg New York: Springer-Verlag, 2007.

[Bou95] D. Boud. *Enhancing Learning Through Self-assessment*. Routledge, 1995.

[Boy05] A. Boyle. Sophisticated Tasks in E-Assessment: What are they? And what are their benefits? In *Proceedings of the 9th CAA Conference*, page 16, June 2005.

[Bru96] P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, July 1996.

[Bru01] P. Brusilovsky. Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 11(1):87–110, 2001.

[Bru03] P. Brusilovsky. Developing Adaptive Educational Hypermedia Systems: From Design Models to Authoring Tools. In T. Murray, S. Blessing, and S. Ainsworth, editors, *Authoring Tools for*

Advanced Technology Learning Environments, pages 377–409. Kluwer, 2003.

[BTP+12] P. Broadfoot, S. Timmis, S. Payton, A. Oldfield, and R. Sutherland. Rethinking Assessment - Learning analytics and technology enhanced assessment (TEA). Technical report, University of Bristol, 2012.

[Bur82] R.R. Burton. Diagnosing Bugs in a Simple Procedural Skill. In D.H. Sleeman and J.S. Brown, editors, *Intelligent Tutoring Systems*, pages 157–184. Academic Press, New York, 1982.

[CC03] A. Cristea and L. Calvi. The three Layers of Adaptation Granularity. In *Proceedings of the 9th International Conference on User Modeling*, pages 4–14. Springer, 2003.

[CCPM04] J. Coldwell, A. Craig, T. Paterson, and J. Mustard. Online Students: Relationships between Participation, Demographics and Academic Performance. *Electronic Journal of e-Learning*, 6(1):19–30, 2004.

[CdM03] A.I. Cristea and A. de Mooij. LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators. In *Proceedings of the 12th International World Wide Web Conference*, page 10, 2003.

[CFLGP03] O. Corcho, M. Fernández-López, and A. Gómez-Pérez. Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data & Knowledge Engineering*, 46(1):41–64, July 2003.

[Cha07] L.M. Chan. *Cataloging and Classification: An Introduction.* Scarecrow Press, 2007.

[Cho00] C. Chou. Constructing a Computer-Assisted Testing and Evaluation System on the World Wide Web - The CATES Experience. *IEEE Transactions on Education*, 43(3):266–272, 2000.

[Con05] O. Conlan. *The Multi-Model, Metadata Driven Approach to Personalised eLearning Services.* Phd thesis, Trinity College Dublin, 2005.

[Cri06] G. Crisp. Interactive e-Assessments. In *Proceedings of the EDUCOM International Conference*, pages 144–153, 2006.

[Cri10] G. Crisp. Interactive E-Assessment - Practical Approaches to Constructing More Sophisticated Online Tasks. *Journal of Learning Design*, 3(3):10, 2010.

[Cro51] L.J. Cronbach. Coefficient alpha and the internal Structure of Tests. *Psychometrika*, 16(3):297–334, 1951.

[Cro60] N.A. Crowder. Automatic teaching by intrinsic programming. In *Teaching machines and programmed learning: A source book*, pages 286–298. National Education Association of the United States, Washington, 1960.

[CSD05] A. Cristea, D. Smits, and P. De Bra. Writing MOT, Reading AHA! -converting between an authoring and a delivery system for adaptive educational hypermedia. In *Proceedings of the International Workshop on Authoring of Adaptive & Adaptable Educational Hypermedia (A3EH)*, pages 36–45, 2005.

[CW05] G. Conole and B. Warburton. A review of computer-assisted assessment. *Alt-J, Research in Learning Technology*, 13(1):17–31, 2005.

[CZQ+05] Z. Chengling, S. Zhimei, L. Qingtang, S. Chaowang, and S. Dandan. Research on initializing student model. In *Proceedings of the 5th International Conference on Advanced Learning Technologies (ICALT'05)*, pages 691–693, 2005.

[dA08] R. J. de Ayala. *The Theory and Practice of Item Response Theory*. Guilford Press, New York, 2008.

[Dav93] B.G. Davis. *Tools for Teaching*. Jossey-Bass, San Francisco, 1993.

[DB06] N.-P. Deibler and W.-H. Blackmon. Best Practices Guide for the Design and Development of SCORM Assessments. Technical report, Carnegie Mellon University, 2006.

[DBK08] H.L. De Penning, E.W. Boot, and B. Kappe. Integrating Training Simulations and e-Leaming Systems: The SimSCORM platform - Final report. Technical report, TNO Defence, Security and Safety, 2008.

[DC98] P. De Bra and L. Calvi. AHA! An open adaptive hypermedia architecture. *New Review of Hypermedia and Multimedia*, 4(1):115–139, 1998.

[DFSP08] B.W. Dargue, G. Frank, B. Smith, and K. Peplow. SCORM - Simulation Interface Standards Study Group - Final Report. Technical report, Simulation Interoperability Standards Organization (SISO), 2008.

[DHI78] W. Dzida, S. Herda, and W.D. Itzfeld. User-perceived quality of interactive systems. *IEEE Transactions on Software Engineering*, SE4(4):170–276, 1978.

[DHW99] P. De Bra, G.J. Houben, and H. Wu. AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In *Proceedings of the 10th ACM Conference on Hypertext and Hypermedia*, pages 147–156, Darmstadt, February 1999.

[Die04] B. Diegner. Erwartungen der Elektrotechnik- und Elektronikindustrie an die Absolventen der Bachelor- und Master-Studiengänge. Technical report, ZVEI, 2004.

[DMKSH93] H. Dieterich, U. Malinowski, T. Kühme, and M. Schneider-Hufschmidt. State of the Art in Adaptive User Interfaces. In *Adaptive User Interfaces: Principle and Practice*, pages 13–48. Elsevier Science & Technology, Amsterdam, 1993.

[DR99] J.S. Dumas and J.C. Redish. *A Practical Guide to Usability Testing*. Intellect Ltd, 1999.

[Duf91] J.A. Duffield. Designing computer software for problem-solving instruction. *Educational Technology Research and Development*, 39(1):50–62, March 1991.

[Dur94] J. Durkin. *Expert Systems Design and Development*. Prentice Hall, 1994.

[DW08] M. Danson and J. Winkley. e-Assessment - An overview of JISC activities. Technical report, Joint Information Systems Committee (JISC), 2008.

[Ell07] B. Elliott. Assessment 2.0 - Assessment in the age of Web 2.0. Technical Report September, Scottish Qualifications Authority, 2007.

[Ell08] B. Elliott. Assessment 2.0 - Modernising assessment in the age of Web 2.0. Technical report, Scottish Qualifications Authority, 2008.

[Erw91] T.D. Erwin. *Assessing Student Learning and Development: A Guide to the Principles, Goals, and Methods of Determining College Outcomes*. Proquest Info & Learning, 1991.

[ES03] C. Evans and K Sabry. Evaluation of the Interactivity of Web-Based Learning Systems: Principles and Process. *Innovations in Education and Teaching International*, 40(1):89–99, 2003.

[FHP07] C. Fadel, M. Honey, and S. Pasnik. Assessment in the Age of Innovation. *Education Week*, 26(38):34–40, 2007.

[FM93] R. Fogarty and J. McTighe. Educating Teachers for Higher Order Thinking: The Three-Story Intellect. *Theory into Practice - Special Issue: Teaching for Higher Order Thinking*, 32(3):161–169, 1993.

[Fos05] C.T. Fosnot. *Constructivism: Theory, Perspectives, and Practice*. Teachers College Press, New York, 2nd edition, 2005.

[FPC11] J. Fontenla, R. Pérez, and M. Caeiro. Using IMS Basic LTI to Integrate Games in LMSs. In *Proceedings of the Global Engineering Education Conference (EDUCON)*, pages 299–306, 2011.

[Gag85] R.M. Gagné. *The Conditions of Learning and Theory of Instruction*. Wadsworth Publishing Company, 4th edition, 1985.

[GC04] E. Guzmán and R. Conejo. A Brief Introduction to the New Architecture of SIETTE. In *Proceedings of the 3th International Conference on Adaptive Hypermedia and Adaptive web-based Systems (AH 2004)*, pages 405–408, 2004.

[GCGH05] E. Guzmán, R. Conejo, and E. García-Hervás. An Authoring Environment for Adaptive Testing. *Educational Technology & Society*, 8(3):66–76, 2005.

[Gen11] J. Genthner. *Konzeption und Implementierung eines Autorenwerkzeugs zur benutzerorientierten Modellierung von Domänen für ein webbasiertes System zur adaptiven Lernerfolgskontrolle*. Diploma thesis, Ilmenau University of Technology, October 2011.

[GHJV94] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.

[GIK99] S. Garlatti, S. Iksal, and P. Kervella. Adaptive on-line information system by means of a task model and spatial views. In *Proceedings of Second Workshop on Adaptive Systems and User Modeling on the World Wide Web*, pages 59–66, 1999.

[GMTR98] T. Garcia, E.J McCann, J.E Turner, and L. Roska. Modeling the Mediating Role of Volition in the Learning Process. *Contemporary Educational Psychology*, 23:392–418, 1998.

[GPG02] E. Gouli, K. Papanikolaou, and M. Grigoriadou. Personalizing Assessment in Adaptive Educational Hypermedia Systems. In *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 153–163, 2002.

[GPKM01] M. Grigoriadou, K. Papanikolaou, H. Kornilakis, and G. Magoulas. INSPIRE: An INtelligent System for Personalized Instruction in a Remote Environment. In *Proceedings of 3rd Workshop on Adaptive Hypertext and Hypermedia*, pages 13–24, 2001.

[Gra07] S. Graf. *Adaptivity in Learning Management Systems Focussing on Learning Styles*. Phd thesis, Vienna University of Technology, 2007.

[Gre08] D. Greer. The Art of Separation of Concerns, January 2008.

[Gru95] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(4-5):907–928, 1995.

[Hey12] K. Heyder. *Konzeption eines Adaptivitätsmodells und dessen Implementierung in einem benutzerorientierten Autorenwerkzeug für die personalisierte Lernerfolgskontrolle.* Diploma thesis, Ilmenau University of Technology, March 2012.

[HMU06] J.E. Hopcroft, R. Motwani, and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 2006.

[HS94] F. Halasz and M. Schwartz. The Dexter Hypertext Reference Model. *Communications of the ACM*, 37(2):30–39, February 1994.

[HSR91] R.K. Hambleton, H. Swaminathan, and H.J. Rogers. *Fundamentals of Item Response Theory.* Sage Publications, Inc., 1991.

[IEE04] IEEE. 1484.11.1 - IEEE Standard for Learning Technology - Data Model for Content to Learning Management System Communication. Technical Report January, IEEE Computer Society, 2004.

[Imr84] B.W. Imrie. In Search of Academic Excellence: Samples of Experience. In *Proceedings of the 10th International Conference on Improving University Experience*, pages 160–183, 1984.

[Imr95] B.W. Imrie. Assessment for Learning : quality and taxonomies. *Assessment & Evaluation in Higher Education*, 20(2):175–189, August 1995.

[ISO97] ISO. ISO 9241-14:1997 Ergonomic requirements for office work with visual display terminals (VDTs) – Part 14: Menu dialogues, 1997.

[ISO98] ISO. ISO 9241-11:1998 Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability, 1998.

[ISO99] ISO. ISO 9241-16:1999 Ergonomic requirements for office work with visual display terminals (VDTs) – Part 16: Direct manipulation dialogues, 1999.

[ISO06] ISO. ISO 9241-110:2006 Ergonomics of human-system interaction - Part 110: Dialogue principles, 2006.

[ISO08] ISO. ISO 9241-151:2008 Ergonomics of human-system interaction – Part 151: Guidance on World Wide Web user interfaces, 2008.

[ISO10] ISO. ISO 9241-210:2010 Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems, 2010.

[ISO12] ISO. ISO 9241-143:2012 Ergonomics of human-system interaction - Part 143: Forms, 2012.

[Jac93] P. Jackson. *Introduction To Expert Systems.* Addison Wesley, 1993.

[JC92] L.C. Jacobs and C.I. Chase. *Developing and Using Tests Effectively: A Guide for Faculty.* Jossey-Bass, San Francisco, 1992.

[Jen96] F.V. Jensen. *An Introduction to Bayesian Networks.* Springer, London, 1996.

[JFM97] T. Joachims, D. Freitag, and T. Mitchell. WebWatcher: A tour guide for the World Wide Web. In *Proceedings of 15th International Joint Conference on Artificial Intelligence*, pages 770–775, 1997.

[Joh12] A. Johnson. An ADL Perspective on Next Generation SCORM Requirements as Derived from Project Tin Can. Technical Report January, Advanced Distributed Learning (ADL) Co-Laboratories, 2012.

[KGR98] F.J. King, L. Goodson, and F. Rohani. Higher Order Thinking Skills. Technical report, Center for Advancement of Learning and Assessment, 1998.

[KHL+07] A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, and E. Giannopoulou. Ontology Visualization Methods - A Survey. *ACM Computing Surveys*, 39(4):43, November 2007.

[KK99] R.D. Kellough and N.G. Kellough. *Secondary School Teaching: A Guide To Methods And Resources; Planning For Competence.* Upper Saddle River, New York, 1999.

[KT65] C.H. Kepner and B.B. Tregoe. *The Rational Manager: A Systematic Approach to Problem Solving and Decision-Making.* McGraw-Hill Book Company, 1965.

[KW02a] N. Koch and M. Wirsing. The Munich Reference Model for Adaptive Hypermedia Applications. In *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 213–222. Springer, 2002.

[KW02b] M.A. Kozloff and N.C. Wilmington. Three Requirements of Effective Instruction: Providing Sufficient Scaffolding, Helping Students Organize and Activate Knowledge, and Sustaining High Engaged Time. Technical report, Watson School of Education, 2002.

[KW09] J.P. Kincaid and K.K. Westerlund. Simulation in Education and Training. In *Proceedings of the Winter Simulation Conference (WSC)*, pages 273–280, 2009.

[LB11] R. Lopes and R. Bidarra. Adaptivity Challenges in Games and Simulations: A Survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(2):85–99, June 2011.

[LCP11] C. Levesque, K.J. Copeland, and M.D. Pattie. Intrinsic and Extrinsic Motivation. In *Social and Emotional Aspects of Learning*, pages 15–20. Elsevier Ltd, 2011.

[Leu92] D. Leutner. *Adaptive Lehrsysteme: Instruktionspsychologische Grundlagen und experimentelle Analysen.* Psychologie Verlags Union, 1992.

[Lev97] D.A. Levy. *Tools of critical thinking: Metathoughts for psychology.* Allyn & Bacon, 1997.

[LGP09] F. Lazarinis, S. Green, and E. Pearson. Focusing on content reusability and interoperability in a personalized hypermedia assessment tool. *Multimedia Tools and Applications*, 47(2):257–278, July 2009.

[LGP10] F. Lazarinis, S. Green, and E. Pearson. Creating personalized assessments based on learner knowledge and objectives in a hypermedia Web testing application. *Computers & Education*, 55(4):1732–1743, 2010.

[LR06] F. Lazarinis and S. Retalis. Engineering an interoperable adaptive hypermedia testing tool supporting user adaptable strategies. In *Proceedings of the Artificial Intelligence Applications and Innovations (AIAI) Conference*, volume 204, pages 393–400, 2006.

[Lug01] G.F. Luger. *Künstliche Intelligenz - Strategien zur Lösung komplexer Probleme.* Pearson Education Deutschland, 2001.

[Mac05] D. Mackenzie. Online assessment: quality production and delivery for higher education. In *Proceedings of the Assessment Workshop Series No. 5 in Reflections on Assessment*, pages 22–29, 2005.

[Mag84] R.F. Mager. *Preparing instructional objectives.* Fearon Teacher Aids, 1984.

[McA02] M. McAlpine. Principles of Assessment. Technical Report 1, CAA Centre, 2002.

[MGH97] M. Montebello, W.A. Gray, and S. Hurley. A Personal Evolvable Advisor for WWW Knowledge-Based Systems. In *Proceedings of Workshop on Reuse of Web information at the Seventh International World Wide Web Conference,*, volume 98, pages 59–68. Citeseer, 1997.

[Mit07] S. Mitra. Web 3.0 = (4C + P + VS), 2007.

[MMHM04] V. Manikonda, P. Maloor, J. Haynes, and S. Marshall. An Architecture for Integrating SCORM-Compliant Instruction with HLA-Compliant Simulation. In *Proceedings of the SISO Fall Simulation Interoperability Workshop*, page 7, 2004.

[Moo89] M.G. Moore. Editorial: Three types of interaction. *American Journal of Distance Education*, 3(2):1–7, 1989.

[ND08] L. Nguyen and P. Do. Learner Model in Adaptive Learning. *World Academy of Science, Engineering and Technology*, 45(21):395–400, 2008.

[ND09] L. Nguyen and P. Do. Combination of Bayesian Network and Overlay Model in User Modeling. *International Journal of Emerging Technologies in Learning (iJET)*, 4(4):5–14, December 2009.

[Neg04] M. Negnevitsky. *Artificial Intelligence: A Guide to Intelligent Systems.* Addison Wesley, 2 edition, 2004.

[Nel65] T.H. Nelson. Complex information processing: a file structure for the complex, the changing and the indeterminate. In *Proceedings of the 1965 20th national conference*, pages 84–100. ACM, 1965.

[Nie93] J. Nielsen. *Usability engineering.* Morgan Kaufmann, 1993.

[NSHLB11] T. Navarrete, P. Santos, D. Hernández-Leo, and J. Blat. QTIMaps : A Model to Enable Web Maps in Assessment. *Educational Technology & Society*, 14(3):203–217, 2011.

[NSW08] NSW Department of Education and Training. Principles for Assessment and Reporting in NSW Government Schools, 2008.

[OPC+09] L. Oneto, M. Poggi, A. Cristea, M. Hendrix, C. Steiner, O. De Troyer, M. Harrigan, and V. Wade. GRAPPLE (D3.1a) - Design specification of a DM definition tool. Technical report, GRAPPLE, 2009.

[OS99] R. Oppermann and M. Specht. Adaptive Information for Nomadic Activities. A process oriented approach. In *Proceedings of the Software Ergonomie Conference*, volume 99, pages 255–264. Citeseer, 1999.

[Ost07] C. Ostyn. SCORM Interactions. Technical report, Ostyn Consulting, 2007.

[Par09] A. Paramythis. *Adaptive Systems: Development, Evaluation and Evolution.* Phd thesis, Universität Linz, 2009.

[PDP00] C.G. Parshall, T. Davey, and P.J. Pashley. Innovative Item Types for Computerized Testing. In *Computerized Adaptive Testing: Theory and Practice*, chapter 7, pages 129–148. Kluwer Academic Publishers, 2000.

[PFB+01] P.E. Parker, P.D. Fleming, S. Beyerlein, D. Apple, and K. Krumsieg. Differentiating Assessment from Evaluation as a continuous Improvement Tool. In *Proceedings of the Frontiers in Education Conference (FIE)*, pages T3A1–6. IEEE Computer Society, October 2001.

[PH07] J. Prentzas and I. Hatzilygeroudis. Categorizing Approaches Combining Rule-Based and Case-Based Reasoning. *Expert Systems*, 24(2):97–122, 2007.

[Phi07] D. Philpitt. Integrating Simulation Based Training into the JADL IPA - Final Report. Technical Report July, BBN Technologies, 2007.

[PR95] J.E. Pitkow and M.M. Recker. Using the Web as a Survey Tool: Results from the Second WWW User Survey. *Computer Networks and ISDN Systems*, 27(6):809–822, 1995.

[PSPT06] K. Pataki, K. Sachse, J. Prümper, and M. Thüring. ISONORM 9241/110-Short: Ein Kurzfragebogen zur Software-Evaluation. In *Proceedings of the 45. Kongress der Deutschen Gesellschaft für Psychologie*, pages 258–259, 2006.

[QCA07] QCA. Regulatory principles for e-assessment. Technical Report April, Qualifications and Curriculum Authority (QCA), 2007.

[Ree00] T.C. Reeves. Alternative Assessment Approaches for Online Learning Environments in Higher Education. *Journal of Educational Computing Research*, 23(1):101–111, 2000.

[RF07] M. Richter and M.D. Flückiger. *Usability Engineering kompakt: Benutzbare Software gezielt entwickeln.* Spektrum Akademischer Verlag, 2007.

[Ric79] E. Rich. User modeling via stereotypes. *Cognitive Science*, 3(4):329–354, October 1979.

[Rus01] C. Rust. *A Briefing on Assessment of Large Groups.* Learning and Teaching Support Network, 2001.

[Sah06] S. Sahin. Computer Simulations in Science Education: Implications for Distance Education. *Turkish Online Journal of Distance Education (TOJDE)*, 7(4):132–146, 2006.

[Sal75] G. Salomon. Heuristische Modelle für die Gewinnung von Interaktionshypothesen. In *Adaptiver Unterricht*, pages 127–145. Kösel, München, 1975.

[SB07] M. Specht and D. Burgos. Modeling Adaptive Educational Methods with IMS Learning Design. *Journal of Interactive Media in Education (JIME) - Special Issue: Adaptation and IMS Learning Design*, 2007(1):13, 2007.

[SB11] A. Schulz and M. Becker. Old School Meets New School: The Adaptive School Book Approach - Adaptivity Extends Conventional School Books with Digital Media. In *Proceedings of the 3rd International Conference on Computer Supported Education (CSEDU 2011)*, pages 130–139, Noordwijkerhout, May 2011.

[SD10] S. Sosnovsky and D. Dicheva. Ontological technologies for user modelling. *Metadata, Semantics and Ontologies*, 5(1):32–71, 2010.

[SG04] E. Shepherd and J. Godwin. Assessments through the Learning Process. Technical report, Questionmark, 2004.

[SHB04] G. Sim, P. Holifield, and M. Brown. Implementation of computer assisted assessment: lessons from the literature. *Alt-J, Research in Learning Technology*, 12(3):215–229, September 2004.

[Sit09] O. Sitthisak. *A competency model for semi-automatic question generation in adaptive assessment.* Phd thesis, University of Southampton, 2009.

[SK00] S.G. Schar and H. Krueger. Using new learning technologies with multimedia. *IEEE Multimedia*, 7(3):40–51, 2000.

[SK02] R. Strzebkowski and N. Kleeberg. Interaktivität und Präsentation als Komponenten multimedialer Lernanwendungen. In *Informationen und Lernen mit Multimedia und Internet*, pages 228–245. Psychologie Verlags Union, 2002.

[Ski54] B.F. Skinner. The Science of Learning and the Art of Teaching. *Harvard Educational Review*, 24(2):969–977, 1954.

[Spe98] M. Specht. *Adaptive Methoden in computerbasierten Lehr/Lernsystemen.* Phd thesis, University of Trier, 1998.

[SPK00] I. Schwab, W. Pohl, and I. Koychev. Learning to Recommend from Positive Evidence. In *Proceedings of the 5th International Conference on Intelligent User Interfaces*, pages 241–247. ACM, 2000.

[TAA+05] R. Thomas, H. Ashton, B. Austin, C. Beevers, D. Edwards, and C. Milligan. Cost Effective use of Simulations in Online Assessment. In *Proceedings of the 9th CAA Conference*, page 14, 2005.

[Tha07] W. Thalheimer. Measuring Learning Results : Creating fair and valid assessments fundamental learning research. Technical Report April, Questionmark, 2007.

[Tho01] R. Thomas. Interactivity & Simulations in e-Learning. Technical report, MultiVerse Solutions Limited, 2001.

[Tho03] R. Thomas. What Are Simulations? - The JeLSIM Perspective. Technical report, JeLSIM, 2003.

[TNB+05] S. Takahama, N. Nakamura, L. Barolli, A. Koyama, A. Durresi, and K. Sugita. An E-learning System for Improving Learner Study Efficiency by Stimulating Learner Volition. In *Proceedings of the 11th International Conference on Parallel and Distributed Systems (ICPADS'05)*, 2005.

[TRP04] A. Tzanavari, S. Retalis, and P. Pastellis. Giving More Adaptation Flexibility to Authors of Adaptive Assessments. In *Proceedings of the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 549–568. Springer, August 2004.

[TV02] V. Tsiriga and M. Virvou. Initializing the Student Model using Stereotypes and Machine Learning. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 404–409, 2002.

[WB01] G. Weber and P. Brusilovsky. ELM-ART : An Adaptive Versatile System for Web-based Instruction. *International Journal of Artificial Intelligence in Education*, 12:351–384, 2001.

[WB09] B. Woolf and W. Burleson. Affect-aware tutors: recognising and responding to student affect. *International Journal of Learning Technology*, 4(3-4):129–164, 2009.

[WH08] H.-D. Wuttke and K. Henke. LMS-Coupled Simulations and Assessments in a Digital Systems Course. In *Proceedings of the 10th IEEE International Symposium on Multimedia*, pages 726–731, 2008.

[WK84] D.J. Weiss and G.G. Kingsbury. Application of Computerized Adaptive Testing to Educational Problems. *Journal of Educational Measurement*, 21(4):361–375, 1984.

[Woo09] A.E. Woolfolk. *Educational Psychology.* Prentice Hall, 2009.

[WUHJ08] H.-D. Wuttke, R. Ubar, K. Henke, and A. Jutman. The Synthesis Level in Bloom's Taxonomy - a Nightmare for an LMS. In *Proceedings of the 19th EAEEIE Annual Conference*, pages 199–204, 2008.

[WWM12] B. Weiss, I. Wechsung, and S. Marquardt. Assessing ICT user groups. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction (NordiCHI 12)*, pages 275–283, 2012.

GLOSSAR

ADAPTATION RULE: An adaptation rule consists of conditions and an action. The action (e.g., selecting a certain question) will be triggered the conditions are fulfilled.

ADAPTIVE TEST: An adaptive test is a set of questions and adaptation rules.

QUESTION A question is more than a simple question in that it contains a question title, a question text (prompt), different answer options, a score and feedback.

STUDENT'S RESPONSE A student's response is a generic term for both student's answer or answers selected/given to a question and a solution given for a task.

USER MODEL: A user model is an explicit representation of the properties of an individual user.

USER PROFILE: A user profile is an instantiation of a user model representing a specific (real) user.

APPENDICES

APPENDIX A

In the following, the different question types provided by e-assessment systems and tools Questionmark Perception, Respondus, TATS, ILIAS, Moodle and OLAT are listed and explained.

Questionmark Perception

- *Drag and drop*: This question type requires students to drag and drop objects to certain areas.
- *Essay*: This question type requires students to write an answer in essay format.
- *Explanation*: This question type is used to provide some information about a subsequent group of questions.
- *File upload*: This question type requires students to upload a file.
- *Fill in blanks*: This question type requires students to complete the missing words within a statement.
- *Hotspot*: This question type requires students to select one or more areas on an image.
- *Likert scale*: This question type requires students to select one of several options that are weighted with numbers to aid analysis of the results.
- *Survey matrix*: This question type presents multiple rows of Likert questions.
- *Matching*: This question type requires students to match items with exactly one item in a list of choices.
- *Multiple-choice*: This question type requires students to check one choice in a list.
- *Knowledge matrix*: This question type presents several multiple-choice questions together.
- *Multiple-response*: This question type requires students to check one or more choices in a list.
- *Numeric*: This question type requires students to enter an answer for a numeric question.

- *Pull-down list*: This question type requires students to select an answer from a list of choices.
- *Ranking*: This question type requires students to rank a list of choices numerically.
- *Select a blank*: This question type requires students to complete the missing words within a statement, but the words can be selected from a list of choices.
- *True/false*: This question type requires students to select from two choices.
- *Yes/no*: This question type requires students to select from two choices.
- *Text match*: This question type requires students to type in a single word or a few words to indicate the answer.
- *Adobe Flash*: This question type allows embedding Adobe Flash applets.
- *Adobe Captivate*: This question type allows embedding Adobe Captivate simulations.
- *Spoken response*: This question type records a student's voice as the answer to a question.

Respondus

- *Multiple-choice*: This question type requires students to check one choice in a list.
- *True/false*: This question type requires students to select from two choices.
- *Essay*: This question type requires students to write an answer in essay format.
- *Short-answer*: This question type requires students to enter a word or phrase.
- *Description*: This question type is used to provide some information about a subsequent group of questions.
- *Embedded answer*: This question type allows combining short-answer, multiple-choice and numeric sub-questions into a single question.
- *Fill in blanks*: This question type requires students to enter a word, short phrase or string of characters.
- *Fill in multiple blanks*: This question type requires students to fill multiple blanks.

- *Ordering*: This question type requires students to put items in a correct order.
- *Arithmetic*: This question type requires students to apply a mathematical formula to answer the question.
- *Numerical answer*: This question type requires students to enter a number or a value within a range of numbers.
- *Long answer*: This question type requires students to enter a complete sentence or paragraph as answer.
- *Drop-down list*: This question type requires students choice one choice from a drop-down list.
- *Multiple-select*: This question type requires students to check one or more choices in a list.
- *Matching*: This question type requires students to match items with exactly one item in a list of choices.

TATS

- *Multiple-choice*: This question type requires students to check one choice in a list.
- *Multiple-response*: This question type requires students to check one or more choices in a list.
- *Order*: This question type requires students to put items in a correct order.
- *Associate*: This question type requires students to graphically associate pairs of items.
- *Match*: This question type requires students to match items with exactly one item in a list of choices.
- *Gap-match*: This question type is similar to match, but requires students to select items from a set of choices and use them to fill gaps.
- *Inline-choice*: This question type requires students to complete the missing words within a statement, but the words can be selected from a list of choices.
- *Text entry*: This question type requires students to enter a word or phrase.
- *Extended text*: This question type requires students to write an answer in a short text.
- *Hottext*: This question type is similar to multiple-choice, but requires students to select the options in the context of a surrounding text.

- *Slider*: This question type requires students to estimate a numerical value on a slider.

ILIAS

- *Cloze*: This question type requires students to fill blanks in a sentence or text with existing choices.
- *Error text*: This question type requires students to select errors in a sentence or text.
- *Essay*: This question type requires students to write an answer in essay format.
- *File upload*: This question type requires students to upload a file.
- *Flash*: This question type allows embedding Adobe Flash applets, whereas the applet offers the question and the logic of the question to communicate with ILIAS. After the student has finished the question, the applet has to send a well-defined number of parameters (e.g., user solution, reached points for the given solution) back to ILIAS for grading purposes.
- *Hotspot/image map*: This question type requires students to select one or more areas on an image.
- *Java applet*: This question type allows embedding Java applets, whereas the applet offers the question and the logic of the question to communicate with ILIAS. After the student has finished the question, the applet has to send a well-defined number of parameters (e.g., user solution or the reached points for the given solution) back to ILIAS for grading purposes.
- *Matching*: This question type requires students to match items with exactly one item in a list of choices.
- *Multiple-choice*: This question type requires students to check one or more choices in a list.
- *Numeric*: This question type requires students to enter an answer for a numeric question.
- *Ordering*: This question type requires students to put items in a correct order.
- *Text subset*: This question type requires students to enter a certain amount of answers for a question.

Moodle

- *Calculated*: This question type creates numerical questions by the use of wildcards, which are filled with individual values before the presentation.
- *Essay*: This question type requires students to write an answer in essay format.
- *Matching*: This question type requires students to match items with exactly one item in a list of choices.
- *Cloze*: This question type requires students to fill blanks in a sentence or text with existing choices.
- *Multiple-choice*: This question type requires students to check one or more choices in a list.
- *Short-answer*: This question type requires students to enter a word or phrase.
- *Numerical*: This question type requires students to enter an answer for a numeric question.
- *Random short-answer matching*: This question type is similar to matching, but the sub-questions are selected randomly from short-answer questions.
- *True/false*: This question type requires students to select from two choices.
- *Description*: This question type is used to provide some information about a subsequent group of questions.
- *Drag and drop*: This question type requires students to drag and drop objects to certain areas.
- *Molecular editor*: This question type requires students to design and submit a molecular structure.
- *Open Protocol for Accessing QUestion Engines* (OPAQUE): This question type allows using questions from other systems.
- *Regular expression short answer*: This question type requires students to answer an open question with a word or a short phrase. Regular expressions are used to analyze the students' answers.

OLAT

- *Single-choice*: This question type requires students to check one choice in a list.

- *Multiple-choice*: This question type requires students to check one or more choices in a list.
- *Kprim*: This question type requires students to decide for each of four answers if it is correct or not.
- *Cloze*: This question type requires students to fill blanks in a sentence or text.
- *Free text*: This question type requires students to insert a text of his choice in a field of any size.

APPENDIX B

In the following, an adaptive test specified according to the IMS QTI specification is presented. The example is taken from the diploma thesis of Kerstin Heyder [Hey12], which was supervised by the author of this thesis. In line 3 and 4 of Listing B.1, the variables TEST_SCORE and TEST_DIFFICULTY are defined, which hold the score of the student and the difficulty level of the question. Subsequently, line 5 to 59 defines and structures the test using the elements *testPart*, *assessmentSection* and *assessmentItem*. By setting the *submissionMode* of the *testPart* to *"individual"* (cf. Line 5), the student is required to submit the response for one question before answering any other item in the test part. Due to that fact, the *outcomeProcessing* is performed each time the student has submitted a question. As indicated in line 61 to 71, it calculates the score of the student each time. The rules for the multiple-choice question is realized in line 72 to 88. The difficulty level is set as a pre-condition (line 8 to 20). That means, only questions are presented that have the corresponding difficulty level. The rule for the slider question is realized using a *branchRule* element (line 51 to 56).

Listing B.1: XML representation of an assessmentTest

```
1  <?xml version="1.0" encoding="UTF-8"?>
 <assessmentTest xmlns="http://www.imsglobal.org/xsd/imsqti_v2p1"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
     schemaLocation="http://www.imsglobal.org/xsd/imsqti_v2p1 http://
     www.imsglobal.org/xsd/imsqti_v2p1.xsd" identifier="TEST_01"
     title="BeispielTest" toolName="askMe!" toolVersion="1.0">
  <outcomeDeclaration identifier="TEST_SCORE" cardinality="single"
      baseType="integer"/>
  <outcomeDeclaration identifier="TEST_DIFFICULTY" cardinality="
      single" baseType="string"/>
5  <testPart identifier="TESTPART_01" navigationMode="linear"
      submissionMode="individual">
   <assessmentSection identifier="SECTION_01" title="Section" visible
       ="false">
    <assessmentItemRef identifier="ITEM_01" href="ITEM_01.xml">
     <preCondition>
      <and>
10      <not>
        <isNull>
         <variable identifier="TEST_DIFFICULTY"/>
        </isNull>
       </not>
15      <match>
        <variable identifier="TEST_DIFFICULTY"/>
        <variable identifier="ITEM_01.DIFFICULTY"/>
       </match>
      </and>
20    </preCondition>
    </assessmentItemRef>
    <assessmentItemRef identifier="ITEM_02" href="ITEM_02.xml">
     <preCondition>
      <and>
25      <not>
        <isNull>
         <variable identifier="TEST_DIFFICULTY"/>
        </isNull>
       </not>
30      <match>
```

```
              <variable identifier="TEST_DIFFICULTY"/>
              <variable identifier="ITEM_02.DIFFICULTY"/>
            </match>
          </and>
        </preCondition>
      </assessmentItemRef>
      <assessmentItemRef identifier="ITEM_03" href="ITEM_03.xml">
        <preCondition>
          <and>
            <not>
              <isNull>
                <variable identifier="TEST_DIFFICULTY"/>
              </isNull>
            </not>
            <match>
              <variable identifier="TEST_DIFFICULTY"/>
              <variable identifier="ITEM_03.DIFFICULTY"/>
            </match>
          </and>
        </preCondition>
        <branchRule target="ITEM_01">
          <lt>
            <variable identifier="TEST_SCORE"/>
            <baseValue baseType="integer">70</baseValue>
          </lt>
        </branchRule>
      </assessmentItemRef>
    </assessmentSection>
  </testPart>
  <outcomeProcessing>
    <setOutcomeValue identifier="TEST_SCORE">
      <divide>
        <product>
          <variable identifier="SCORE"/>
          <baseValue baseType="float">100.0</baseValue>
        </product>
        <sum>
          <testVariables variableIdentifier="SCORE"/>
        </sum>
      </divide>
    </setOutcomeValue>
    <outcomeCondition>
      <outcomeIf>
        <and>
          <match>
            <variable identifier="ITEM_02.RESPONSE"/>
            <correct identifier="ITEM_02.RESPONSE"/>
          </match>
          <durationLT>
            <variable identifier="duration"/>
            <baseValue baseType="integer">300.0</baseValue>
          </durationLT>
        </and>
        <setOutcomeValue identifier="TEST_DIFFICULTY">
          <baseValue baseType="string">2</baseValue>
        </setOutcomeValue>
      </outcomeIf>
    </outcomeCondition>
  </outcomeProcessing>
</assessmentTest>
```

The rules for the true/false question are defined on question level and thus integrated into the *assessmentItem* element (cf. Listing B.2). For this purpose at first, the outcome variable FEEDBACK is defined in line 13. This variable is set by the response processing (line 47 to 49) with to the answer selected by the student and evaluated in order to show the feedback specified in line 23 and 26, respectively.

Listing B.2: XML representation of an assessmentItem

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <assessmentItem xmlns="http://www.imsglobal.org/xsd/imsqti_v2p1"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
      schemaLocation="http://www.imsglobal.org/xsd/imsqti_v2p1 http://
      www.imsglobal.org/xsd/imsqti_v2p1.xsd" identifier="ITEM_01"
      title="Kaefer" adaptive="false" timeDependent="false" toolName="
      askMe!" toolVersion="1.0">
3  <responseDeclaration identifier="RESPONSE" cardinality="single"
      baseType=" identifier">
4   <correctResponse>
5    <value>True</value>
6   </correctResponse>
7  </responseDeclaration>
8  <outcomeDeclaration identifier="SCORE" cardinality="single"
      baseType="integer"/>
9   <defaultValue>
10   <value>0</value>
11  </defaultValue>
12 </outcomeDeclaration>
13 <outcomeDeclaration identifier="FEEDBACK" cardinality="single"
      baseType="identifier"/>
14 <outcomeDeclaration identifier="DIFFICULTY" cardinality="single"
      baseType="string">
15  <defaultValue>
16   <value>1</value>
17  </defaultValue>
18 </outcomeDeclaration>
19 <itemBody>
20  <choiceInteraction responseIdentifier="RESPONSE" shuffle="false"
       maxChoices="1">
21   <prompt>Bugs are the world's biggest order from the class of
        insects.</prompt>
22   <simpleChoice identifier="True">Right
23    <feedbackInline outcomeIdentifier="FEEDBACK" identifier="True"
         showHide="show">Well done!</feedbackInline>
24   </simpleChoice>
25   <simpleChoice identifier="False">Wrong
26    <feedbackInline outcomeIdentifier="FEEDBACK" identifier="True"
         showHide="show">Are you sure?</feedbackInline>
27   </simpleChoice>
28  </choiceInteraction>
29 </itemBody>
30 <responseProcessing>
31  <responseCondition>
32   <responseIf>
33    <match>
34     <variable identifier="RESPONSE"/>
35     <correct identifier="RESPONSE"/>
36    </match>
37    <setOutcomeValue identifier="SCORE">
38     <baseValue baseType="float">1</baseValue>
39    </setOutcomeValue >
40   </responseIf>
41   <responseElse>
42    <setOutcomeValue identifier="SCORE">
43     <baseValue baseType="float">0</baseValue>
44    </setOutcomeValue>
45   </responseElse>
46  </responseCondition>
47  <setOutcomeValue identifier="FEEDBACK">
```

```
      <variable identifier="RESPONSE"/>
    </setOutcomeValue>
  </responseProcessing>
</assessmentItem>
```

APPENDIX C

Application Scenarios

In the following, three application scenarios are described, which illustrate how the *askMe!* system is envisaged to be used by the main actors (i.e., administrator, author and student). Each scenario is a narrated description of a causally connected sequence of (inter-)actions with the different components of the system.

Application Scenario #1 – Author

Tom is seminar leader of a course at a German university. The course is called computer engineering and the objectives of the course are getting acquainted with basic knowledge in the field of computer engineering, the design of digital systems and the understanding of computer internal information processing, computer architectures and organizations. The course consists of a lecture and an accompanying seminar. While in the lecture primarily theoretical basics are taught, the seminar focuses on exercising the application of the theoretical foundations. The course is completed with a final exam, which is usually characterized by a high failure rate. Due to that fact, Tom was searching for an opportunity that provides students with individual assistance and support, and helps them identifying their strengths and weaknesses so that they can address their deficits early (without failing the exam once). Especially, the theory taught in the lectures should be put into practice. Tom immediately thought of the use of the variety of interactive Java applets, which were created by several student projects. The professor uses some of them in the lectures for demonstration purposes. However, due to the tight time frame, more than just a short demonstration is mostly not possible. These applets enable students to actively discover concepts, conducting experiments by manipulating data as well as creating and testing hypotheses. Moreover, they allow this degree of interactivity that is needed to elicit intensive elaboration processes. Neither a seminar nor a lecture can provide this.

A colleague of Tom has called his attention to the *askMe!* system. Therefore, he asked the responsible administrator Chris to set up an account for him. Chris complied with this request and sent Tom his access data. After Tom has logged into the system, he was forwarded to a starting page on which he found several tips and guidelines how to use the different components of the system. After studying this information, he decided to create an interactive and personalized test. The first step is specifying the main topics that will be addressed by the questions included in the test. These topics are represented and connected to each other using a domain model. Each topic was represented by a concept and connected to each other using associative and/or hierarchical relationships. Finally, Tom got a comprehensive domain model that includes and clearly shows all topics (concepts), which can be used to associate questions. In addition to creating a new domain model, Tom is also able to subscribe to and use models created by other authors.

The next step is configuring the user model. A user model characterizes the students, which will be presented with the tests subsequently. A rough structure of this model is already provided by the system, it consists of a set of general user features namely *knowledge*, *interests*, *goals*, *background* and *demographic information*. Referring to these features, the system required Tom to make a series of inputs. With respect to knowledge, Tom was asked to define a lower and upper performance limitation value, which limit the extent to what the probability that a student knows a specific concept can rise/fall. With regard to interests, Tom was requested to select a domain model that students can use to specify their interests. Tom selected the model he just created. In addition, he had to define degrees with which the students can express their interests to the containing concepts. He had to choose between several input types such as single-select, multiple-select or free-text input. Tom defined a single select, where the students can choose between *weak*, *medium* and *strong*. This procedure is the same for goals and background, with the exception that the input type of students' goals is predefined by the system. Here, the user model makes use of the *competency taxonomy* defined once in the *askMe!* system. For students' background, Tom selected an additional domain model that also includes topics not covered by the lecture. Furthermore, he lets students specify their background using the degrees *novice*, *beginner*, *advanced* and *expert*. Regarding the demographic information, Tom did not need to select a domain model, but he had to refer to a predefined structure (i.e., IMS LIP by default). He chose elements from this structure and specified the way how to enter the inputs (e.g., for students' age a free-text input was defined). This completed the configuration of the user model.

Subsequently, Tom has asked Chris to set up an account for each student that participates in his seminar and to combine them into a group. The next task is creating questions for their use in adaptive tests. The creation of a new question requires the definition of a *title*, a *description*, a *score*, a *thematic relationship* to at least one concept of a domain and the *competencies addressed* by the question as well as the actual *interaction*. The interaction specifies the type of the question that means how the student answers/solves the question/task. For this, Tom could choose from a pool of traditional question types such as true/false, single- and multiple-choice, hotspot, graphical associate, etc., but could also make use of an advanced question type that allows integrating ICOs. This term refers to all the interactive Java applets that were created by students in the past. While the traditional question types only require the definition of answer options including at least one correct answer and optional feedback, the advanced question type requires a little more configuration effort. Before Tom can use ICOs for creating interactive questions, he has to add them to the system. For this purpose, he has to specify where the ICO is located as well as which variables/events he wants to handle. After that, he was able to use the ICOs when creating questions.

In contrast to traditional questions, an interactive question also required Max to initialize the external object as well as to define a state an interactive question/task is deemed to be completed. This is done using the variables and events he specified before. Max created a set of questions using both traditional as well as advanced question types.

The next step is grouping the created questions into adaptive tests. The first step in creating a new test is selecting the questions to be added to the test. Then, these questions are presented in a sequential order. Tom changed the order of the questions using drag and drop according to his needs. Afterward, he dedicated himself to the individual questions and defined associated *adaptation rules.* In order to do that, he firstly determined the *trigger point* for the rule, which means the time when the rule will be checked. The definition of trigger points is done using a graph-based visualization, which allows specifying trigger points before and after presenting a question as well as during the interaction (e.g., when selecting a specific answer option). After that, Tom was able to configure the adaptive rule in detail. This includes the definition of *conditions* and associated *actions* to be executed. For defining conditions, Tom could make use of students' performance in the test (e.g., a score of a question is greater than a specific value) as well as their individual profiles stored in the system (e.g., the existence of a specific interest). More than one condition can be linked using both disjunctive (*or*) and conjunctive (*and*) relationships. With respect to the actions to be executed, Tom had the choice between a large variety of options including retrying, skipping as well as filtering subsequent questions.

After finalizing the adaptive tests, Tom assigned the newly created tests to his seminar group. From then on, these tests were available for each student of his seminar group. As soon as the students completed the test, Tom gained a detailed insight into their results (e.g., number of correct answered questions, conceptual relationships or competency relationships) separated and aggregated. This information helped them to tailor the subsequent seminars for students' needs.

Application Scenario #2 – Student

Max is student of engineering computer sciences. During his study, he was faced with the course that is supported by Tom. Max had problems of understanding with this course. Although he understood the most part of the exercises done in the seminars, however, he did not feel quite confident in dealing with these topics. He doubted that he did entirely understand and was afraid that he will fail in the final exam. The exam does not only require students to reproduce knowledge, but to apply them in design, analysis and synthesis tasks. For that reason, Max was strongly interested in knowing his deficits in order to actively address them. Once in a seminar, Max told his concerns the seminar leader Tom. Although he understood his concerns, an individual support was not feasible due the big seminar groups and the tight time frame. However, he recommended Max testing the *askMe!* system. Tom told him that he had recently imputed a set of questions and tests specially designed for this course. Max followed his recommendation and asked Chris for an account to the *askMe!* system.

There, Max found several tests chronologically sorted according to the topics of the course. During the first login, he was asked to initialize his individual user profile. He was requested to give his interests, goals and background. Therefore, a list of topics relevant for the course was presented to Max. Using this list, he could specify his interests and goals by selecting a topic and a corresponding level. When specifying the background, Max could also select topics that were not related to the course. Finally, some demographic information was asked to specify. After initializing his profile, he was able to start the tests provided. During the course of the first test, he noticed that this test was totally different than initially expected. He already knew web-based tests from the LMS Moodle, which was occasionally used in other courses. Instead of being primarily presented with multiple-choice questions, Max was surprises to come across with application-oriented tasks. There, he was faced with interactive animations and simulations, which required the practical application of the knowledge that was taught in the lectures, instead of selecting an answer from a list of choice. Some of the tools looked quite familiar to him. Within the course of the tests presented by the system, Max was faced with questions and tasks, which were always challenging, but never overstraining. But when he stuck with a certain question/task, he immediately got feedback by the system. The feedback did not solve the question/task at all, instead hints were given to Max that guided him to the correct answer/solution. When Max has completed the test, he was not presented with an abstract score, but got a detailed overview about his knowledge level according to the topics addressed by the test. In this way, Max obtained direct feedback on his strengths and weaknesses, and could efficiently address specific deficits. During the next months, Max has taken all further tests and finally successfully passed the final exam.

Application Scenario #3 – Administrator

Chris is administrator of the *askMe!* system. He is responsible for both configuring structural settings such as the competency taxonomy and students' demographic information as well as for managing user account and groups. For the competencies, Chris defined Anderson and Krathwohl's taxonomy of intellectual skills in the cognitive domain as standard taxonomy for competency definitions in the *askMe!* system. The second predefined structure is the structure used for specifying students' demographic information. Here, Chris used the IMS LIP specification, which is a common standard for exchanging student information between different systems. These structures are the same for each author and intended to be used for configuring student features. In contrast, users/user groups are specific for each author. As requested by Tom, Chris created a number of student accounts and grouped them together. Then, he associated this group to Max. Adding a new user required Chris to input the first and last name of the student and an assignment to a group, but also a login and password to the *askMe!* system. Here, Chris used the last name of the student. However, these login credentials can later be changed by the respective student.

APPENDIX D

Functional System Requirements

Table D.1, D.2, D.3 and D.4 list all functional requirements related to the user modeling, question modeling, domain modeling and adaptive testing engine component.

ID	REQUIREMENT	PRIORITY
UM-1	Administrators are able to create, edit and delete user accounts.	1
UM-2	Administrators are able to create, edit and delete user groups and associate users to them.	1
UM-3	Administrators are able to assign the supervision of user groups to specific users.	1
UM-4	Administrators are able to specify data structures for storing/organizing student information.	2
UM-5	The system provides a predefined set of student features (i.e., knowledge, interests, goals, background and demographic information).	1
UM-6	Authors are able to individually configure student features.	1
UM-7	Authors are able to limit the extent to what the probability that a student knows a specific concept can increase/decrease.	2
UM-8	Authors are able to select a domain model that students can use to specify their interests, goals and background.	1
UM-9	Authors are able to define input options, which students can use to express their interests, background and demographic information.	1
UM-10	Students are able to specify their interests, goals, background and demographic information.	1
UM-11	Students are able to edit their accounts settings (i.e., login and password).	2

Table D.1: Functional requirements for the user modeling component

ID	Requirement	Priority
QM-1	Authors are able to create, edit and delete questions.	1
QM-2	Authors are able to specify the type of a question using different question types.	1
QM-3	The system provides traditional types such as true/false, single- and multiple-choice, hotspot and graphical associate.	2
QM-3	The system provides an advanced question type that allows integrating ICOs.	1
QM-4	Authors are able to configure the integration of and the communication with the ICO.	1
QM-5	Authors are able to thematically relate questions with domain concepts.	1
QM-6	Authors are able to specify the competencies addressed by the question.	1
QM-7	Authors are able to define feedback to be presented to the students.	2
QM-8	The system allows importing and exporting questions.	4

Table D.2: Functional requirements for the question modeling component

ID	Requirement	Priority
DM-1	Authors are able to create, edit and delete domain models.	1
DM-2	Authors are able to create, edit and delete concepts and assign them to domain models.	1
DM-3	Authors are able to create, edit and delete concept relationships.	1
DM-4	The system provides a set of relationship types (e.g., is-a or instance-of).	1
DM-5	Authors are able to create additional relationship types.	3
DM-6	Authors are able to share domain models with other authors.	2
DM-7	Authors are able to subscribe to domain models created by other authors.	2
DM-8	The system graphically visualizes the domain models created.	1

Table D.3: Functional requirements for the domain modeling component

ID	REQUIREMENT	PRIORITY
ATE-1	Students are able to take adaptive tests.	1
ATE-2	Authors are able to gain a detailed insight into students' test results.	2
ATE-3	Students are able to obtain detailed information about their strengths and weaknesses derived from the test results.	1

Table D.4: Functional requirements for the adaptive testing engine component

Non-Functional System Requirements

General Dialog Principles

Part 110 of International Organization for Standardization (ISO) 9241 [ISO06] deals with the ergonomic design of interactive systems and describes general dialog principles. They are based on the seven factors of *user-perceived quality* defined by Dzida, Herda and Itzfeld [DHI78]. The seven principles described by the standard served as general goals for the design and implementation of dialogs in the *askMe!* system:

- Conformity with user expectations
- Suitability for the task
- Self-descriptiveness
- Suitability for learning
- Controllability
- Error tolerance
- Suitability for individualization

For each of these principles, a set of recommendations are given to illustrate these principles. Several of them were considered in the implementation of the *askMe!* system. This includes providing users with information about the expected input, assistance in detecting and avoiding errors as well as guidance, feedback and status information for completing dialogs. Moreover, the system provides immediate and suitable feedback on user actions and explanation and request confirmation before carrying out the specific action (e.g., deleting elements). Furthermore, the dialog behavior is consistent within tasks and across similar tasks (e.g., the save and cancel button are always located at the same position) and the system feedback and explanations support the user in building a conceptual understanding of the system.

Guidance on Menu Dialogs

Part 14 of ISO 9241 [ISO97] provides recommendations for the ergonomic design of menus used in dialogs (e.g., pop-up, pull-down or text-based menus). The recommendations provided by the standard cover the following aspects:

- Menu structures
- Menu option selection and execution
- Menu navigation
- Menu presentation

Several recommendations included in this standard were considered in the implementation of the *askMe!* system for building logical menu categories, grouping, ordering and presenting of menu items as well as for providing navigation support. The aim is to support users in navigating and searching through *askMe!*'s menu structures and inform them, which items were selected and which actions will be carried out, respectively.

Guidance on Direct Manipulation Dialogs

Part 16 of ISO 9241 [ISO99] provides recommendations for the ergonomic design of direct manipulation dialogs. With direct manipulation, the user acts directly on the objects on the screen, for example, by dragging and dropping an object to change its position in a sequence. The recommendations provided by the standard address the following aspects:

- General information
- Manipulation of objects
- Direct manipulation of text objects, windows and control icons

Several recommendations included in this standard were considered in the implementation of the *askMe!* system for pointing, manipulating and selecting elements as well as to arrange them in a specific order using drag and drop.

Guidance on Form-filling Dialogs

Part 143 of ISO 9241 [ISO12] provides recommendations for the ergonomic design of form-filling dialogs that require users to enter textual input. The recommendations provided by the standard encompass the following aspects:

- Form filling structure
- Feedback
- Input considerations

- Navigation

Several recommendations included in this standard were considered in the implementation of the *askMe!* system for creating user-friendly form-filling structures, tailoring the form fields to input types (e.g., alphanumeric text entry or list of choices), validating inputs, preparing and presenting input errors, setting cursor and pointer positions as well as for moving between different fields and tabs.

Guidance on World Wide Web User Interfaces

Part 151 of ISO 9241 [ISO08] provides guidance on the human-centered design of user interfaces for the Web. Due to the fact that the *askMe!* system was realized as Web application, this standard is of particular importance. The recommendations provided by the standard focus on following aspects:

- High-level design decisions/strategy
- Content design
- Navigation and search
- Content presentation

Although several recommendations in this standard can also be found in other parts of ISO 9241, however, it also includes several new recommendations. Some of them were considered in the implementation of the *askMe!* system. This includes a consistent page layout and title location, page lengths, the minimizing of vertical and the avoidance of horizontal scrolling as well as the easily identification of links and self-explanatory link cues.

APPENDIX E

Question Modeling Component Design and Implementation

This section focuses on the design and implementation of the question modeling component. Figure 4.35 depicts the component based on the structure presented in Section 4.2.4. In the following sections, the different layers are described in detail.

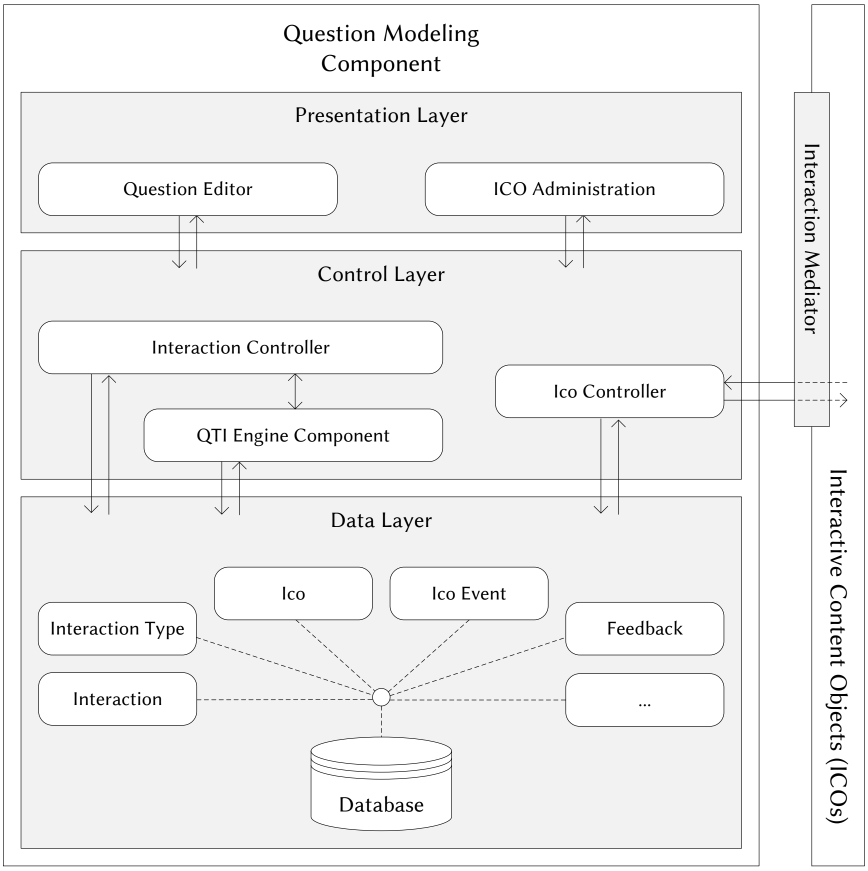


Figure E.1: Architectural structure of the question modeling component

Data Layer

The data layer of the question modeling component is based on a relational database consisting of a set of database tables. All of them are represented by specific models. The question modeling component uses models originally managed by other components such as the competency taxonomy dimension and level model, which are administered by the settings component as well as the domain or concept model originally managed by the domain modeling component. In addition, the component maintains several own models:

- *Interaction*: This model represents the information of the questions or tasks managed by the component.

- *Interaction Type*: This model represents the type of the question (e.g., a multiple-choice question or a task using an ICO).
- *Feedback*: This model represents the feedback associated to the questions/tasks.
- *Ico*: This model represents the ICOs used by the component.
- *Ico Event*: This model represents the information about students' interactions with the ICOs.

The different models are linked together using different associations (cf. Figure E.2). An interaction type is obligatory to an interaction, however, an association to a domain or an associated concept as well as to a knowledge level and type (2x competency taxonomy dimension and level) is optional. In addition, an interaction can make use of an ICO, but an ICO can be used by many interactions. This requires that an ico event belongs to a specific interaction.

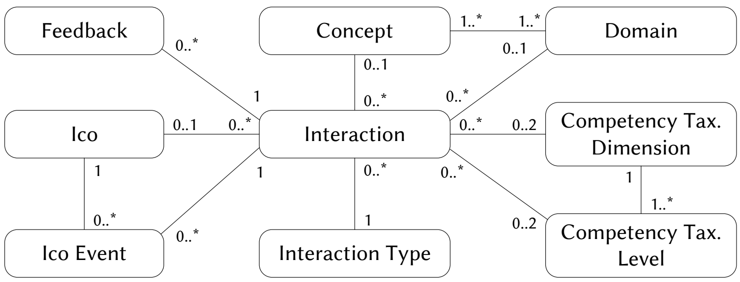


Figure E.2: Relations between the question modeling component models

Control Layer

The control layer of the question modeling component consists of the *interaction* and *ico controller* as well as the *QTI engine component*. They provide the functionality associated with the control and workflow of the question modeling component.

The *interaction controller* provides all actions for creating, editing and deleting interactions. The term *interaction* comes from the IMS QTI specification (cf. Section 3.3.4.3), which classifies a question or task according to the interaction (e.g., choice interaction, order interaction or custom interaction) provided. In the *askMe!* system, a question or task only includes one interaction. The interaction controller also makes use of the QTI engine component (cf. Section 4.7.4.3) in order to allow authors previewing their questions/tasks created.

The *ico controller* is responsible for importing, editing and deleting references to ICOs. An ICO is accompanied by a manifest, an Extensible Markup Language (XML) schema that describes which attributes (variables) and events the external object provides. When importing the ICO, the ico controller reads the values of the manifest and prepares the ICO to be used in

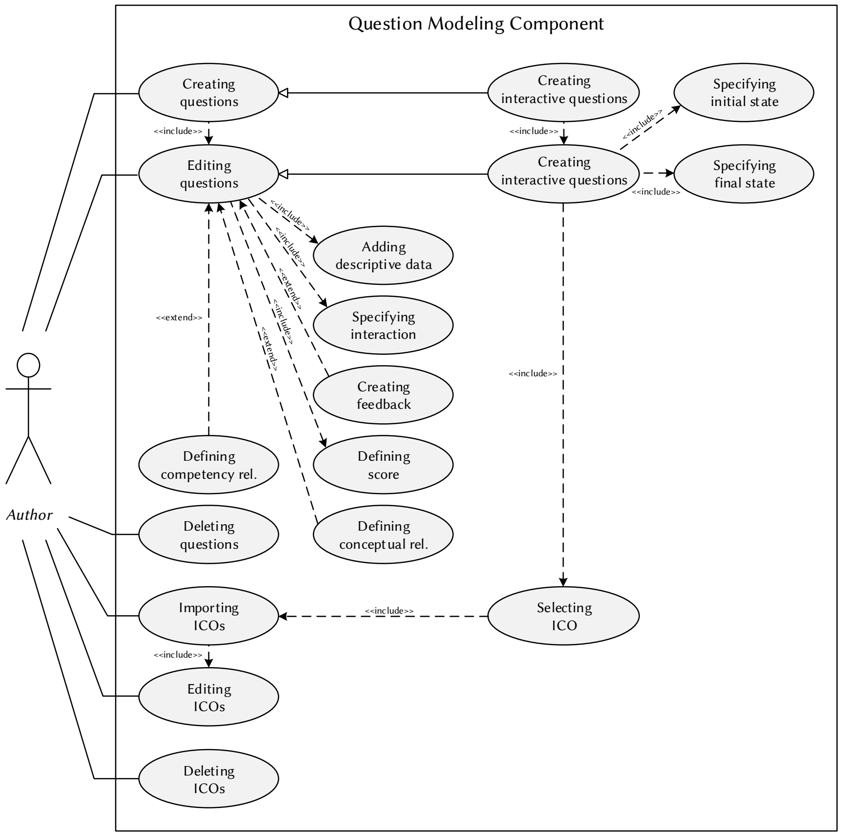


Figure E.3: Use case diagram of the question modeling component

interactive questions/tasks. Due to the fact that the xAPI specification has been selected as communication mechanism (cf. Section 4.3.4.2) between the ICO and the *askMe!* system, all ICOs have to send variable changes or events using xAPI statements. In case an ICO does not implement the xAPI specification, an *interaction mediator* has been developed. It mediates between the *askMe!* system and the ICO by sending and receiving xAPI statements, but also generates the code to set up and initialize the ICO.

Presentation Layer

The presentation layer of the question modeling component consists of two main views (cf. Figure E.1), each encapsulates certain functionality. The views or the component as such are exclusively designed for being used by authors (content creators). A graphical representation of the various ways an author can interact with the question modeling component is depicted in Figure E.3.

The *question editor* concentrates all functions concerning the creation, editing and deletion of questions. The starting point of this editor is a list of questions created by the author (cf. Figure E.4). A question is specified by a title, a question type and a concept this question is related to. Using the icons on the right, a specific question can be previewed, edited or deleted immediately. This fulfills requirements QM-1. New questions can be created by clicking on the *Add Question* button below the table. Then, the author is required to select the question (interaction) type he or she wishes to create. The list not only contains traditional question types such as multiple-choice or text-entry, but also an advanced question type that allows integrating ICOs. Consequently, this fulfills requirements QM-2 and QM-3. When the author has selected the question type, several input fields according to the question type will be required to fill. This includes general information such as the title and a short description, a prompt, answer options and the score. When choosing the advanced question type, the author is also required to select an ICO as well as to set the initial and final state. When the student reaches the final state, the interactive question is regarded as completed. Furthermore, in order to select questions best suited for the individual students, a conceptual (thematic) relationship as well as a competency relationship can be set. The former one allows specifying a domain and concept the question is related to and the later one a level and type of knowledge the question addresses. In this way, requirement QM-5 and QM-6 are satisfied. In addition, the question editor also allows creating feedback to provide students with elaborative and/or verificative information (cf. Section 4.3.4.3). Thus, requirement QM-7 is also met. At the time of this writing, the import and export of questions is not yet fully implemented and thus not yet included in the question editor. Therefore, requirement QM-8 is not yet fulfilled. However, it is only of low priority.

The *ico administration* enables the actual administration of the ICOs handled by the system. This includes importing, editing and deleting references to the external objects. The overview of the references created by the author is presented in a table. The table entries provide information about the name of the ICO, but also the date when the reference was established. When adding a new reference, the author can select the ICO locally and upload it to the system, but can also import the ICO by providing an URI. Not the object itself has to be referenced, but a meta-data file that describes where the ICO is located, which variables (attributes) it holds and which events it provides. After that, the author has to select, which variables/events he or she wants to handle/evaluate later on. Finally, the reference to the ICO will be saved and the ICO can be used to create interactive questions. Thus, the remaining requirement QM-4 is also met.

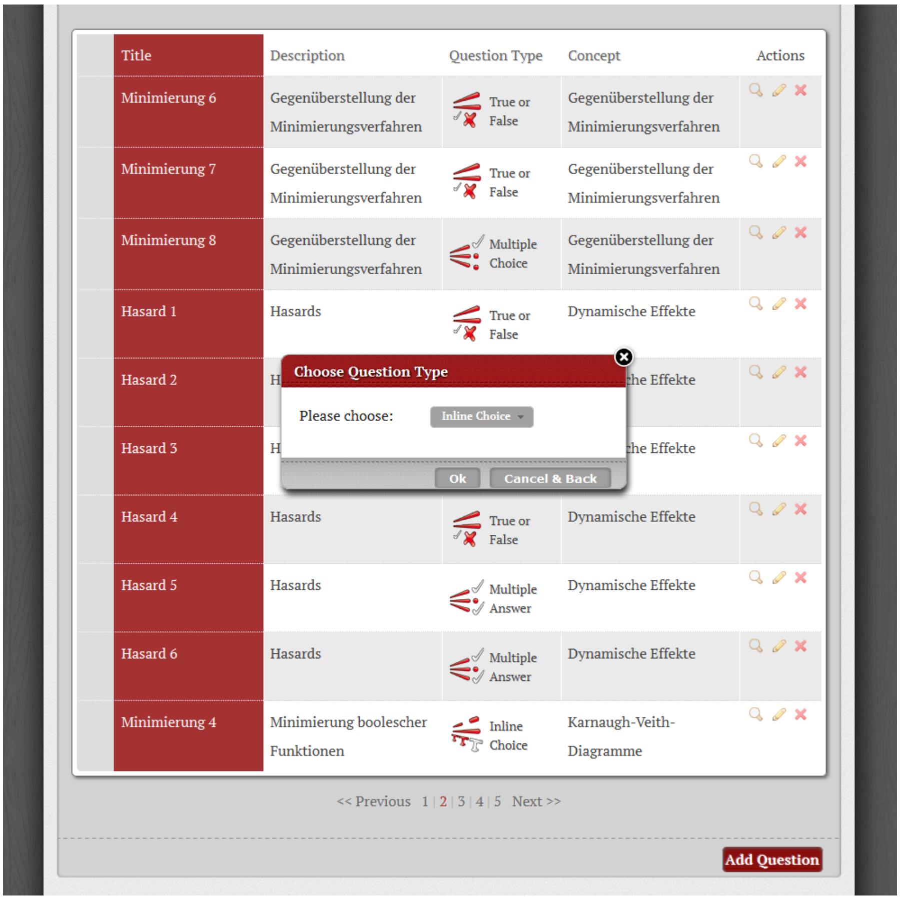


Figure E.4: askMe! question editor

User Modeling Component Design and Implementation

This section focuses on the design and implementation of the user modeling component. Figure E.5 depicts the component based on the structure presented in Section 4.2.4. In the following sections, the different layers are described in detail.

Data Layer

The data layer of the user modeling component is based on a relational database consisting of a set of database tables. They are represented by the following models:

- *User*: This model represents the users of the system.
- *Group*: This model represents the groups, users belong to. Each group is accompanied with specific rights to, for example, access controllers or perform actions.
- *Knowledge*: This model represents the configurations that authors can define to limit the increase and decrease of students' knowledge (probability).

- *Background*: This model represents the configurations that authors can define to allow students specifying their background.
- *Interest*: This model represents the configurations that authors can define to allow students specifying their interests.
- *Goal*: This model represents the configurations that authors can define to allow students specifying their goals.
- *Demographic information*: This model represents the configurations that authors can define to allow students input their demographic information.
- *Student information*: This model represents the data structures that authors can use to store/organize students' demographic information.
- *Student background*: This model represents students' background.
- *Student interest*: This model represents students' interests.
- *Student goal*: This model represents students' goals.
- *Student demographic information*: This model represents students' demographic information.

The different models do not stand for their own, but they are linked to each other using different associations (cf. Figure E.6). Figure E.6a shows the general overview of the different models. Each user belongs to exactly one group. Example groups could be *administrators*, *authors* and *students*. If a user is assigned to the author group, he or she can define configurations that allow students specifying their features. While background, interests and goals are generally configured, demographic information can be subdivided into different categories. This explains the zero to many relationships of user and demographic information. In contrast, if a user is a student, he or she can specify its background, interests, goals and demographic information. Figure E.6b shows a more detailed presentation of the models related to the authors' point of view. It includes the models input type and domain as well as competency taxonomy dimension and level. Although they are maintained separately (e.g., domain by the domain modeling component), however, they are used by this component to configure students' features. Finally, Figure E.6c focuses on the models relevant for the students' point of view. It shows that students use the feature configurations defined by the authors as well as concepts from a domain (model) to express their background, interests, goals and demographic information.

Control Layer

The control layer of the user modeling component consists of a set of controllers and components, which provide the functionality associated with the control and workflow of the component. These are the *user*, *group*, *student feature* and *student data* controller as well as the *user data handler component*.

In general, the *user, group and user feature controller* are responsible for receiving and preparing user inputs from the corresponding views and process them. Furthermore, they initiate the rendering of the views, which are presented to the users. While the user controller provides all actions for creating, editing and deleting of user accounts, the group controller provides all actions for creating, editing and deleting user groups and makes actions for restricting controllers and actions as well as associating users to groups available. User accounts are associated to one of these groups and whose rights (e.g., for accessing specific menu items) are automatically assigned. The student feature controller provides all actions for configuring and specifying students' features and the student data controller provides all actions for building data structures for storing and organizing information about students.

The *user data handler component* encapsulates all core functions, which are shared between the user modeling component controllers. This includes reading student information data structures or parts of, saving new and deleting existing structures.

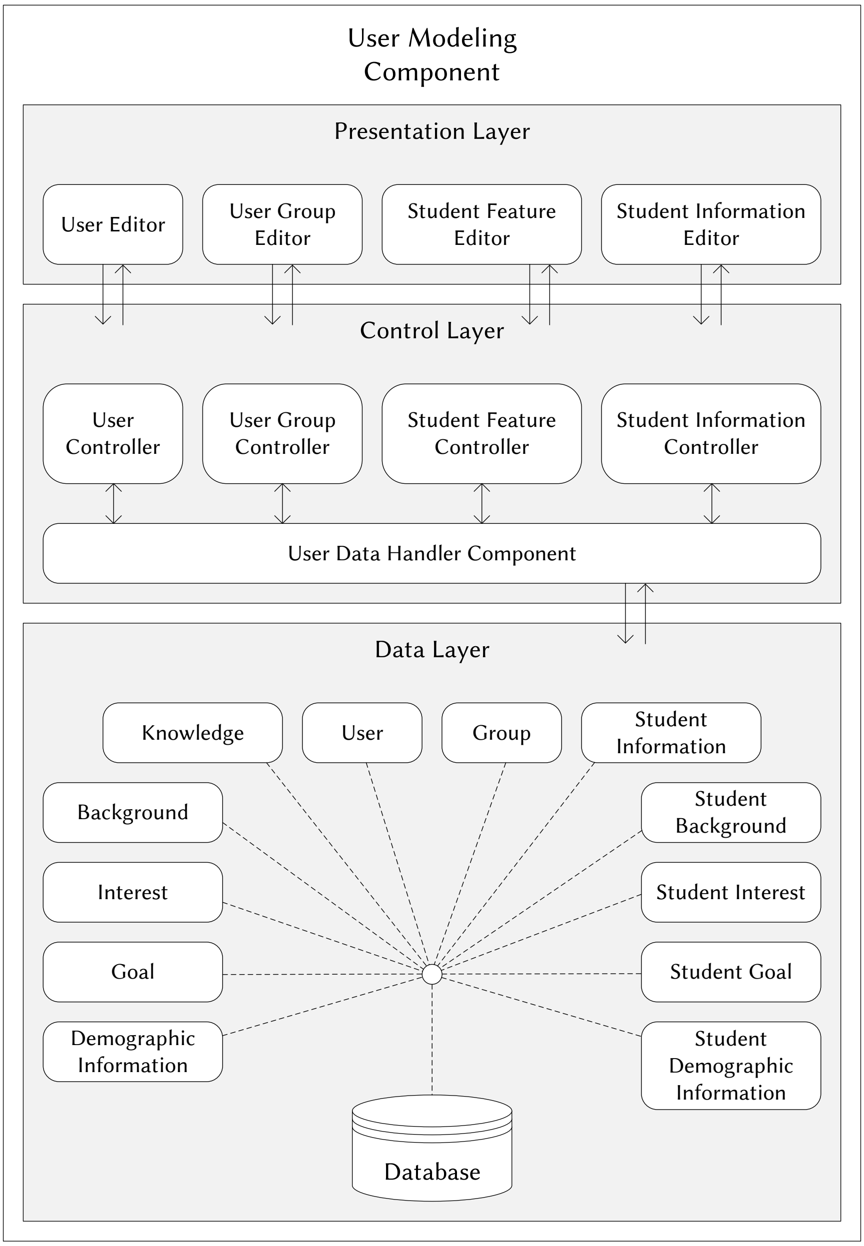
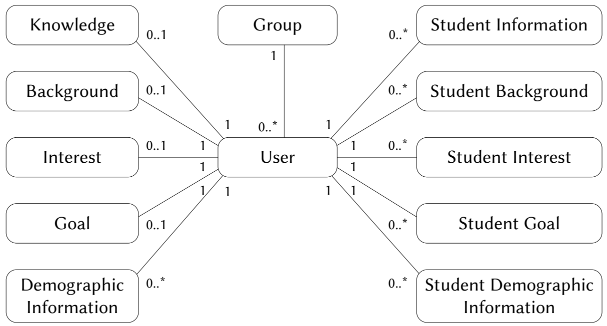
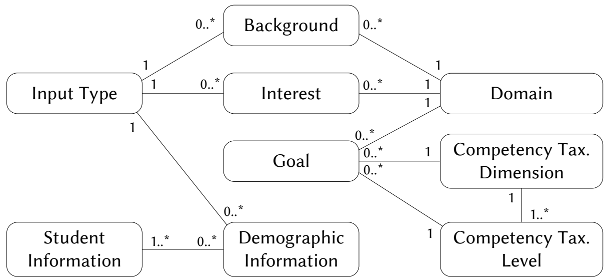


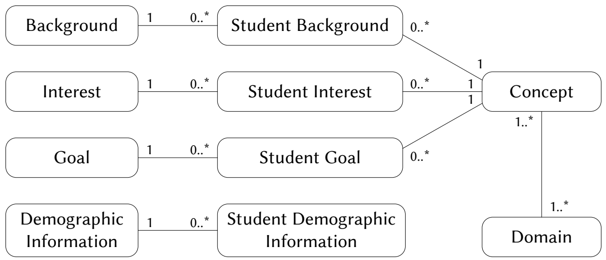
Figure E.5: Architectural structure of the user modeling component



(a) General overview of the model relations



(b) More detailed presentation of the model relations from the authors' point of view



(c) More detailed presentation of the model relations from the students' point of view

Figure E.6: Relations between the user modeling component models

Presentation Layer

The presentation layer of the user modeling component consists of four main views (cf. Figure E.5), each encapsulates certain functionality. It should be noted that the terms view and component are used here in a general way and not referring to CakePHP. The views or the component as such are designed for being used by administrators, authors and students. In what ways the different actors are able to interact with the user modeling component is depicted in Figure E.7.

The *user editor* enables administrators of the *askMe!* system managing user accounts. This includes creating, editing and deleting users. At first, the editor shows a list of accounts already created. Adding a new account requires the administrator to input the first and last name of the user as well as an assignment to a group (e.g., student or author). Additionally, also a login and password must be defined, with which the user can log in. These values can later be changed by the respective user. If a user account needs to be deactivated for any reason, this can also be done by unchecking the option *active*. In this way, the requirements UM-1, partly UM-2 and UM-9 derived from application scenario #3 are met.

The *user group editor* is also provided for administrators and allows managing user groups. This includes creating, editing and deleting groups. Initially, the editor shows a graphical overview of the different groups already created. Groups can be organized hierarchically in any way the administrator desires. This allows creating a group for all students taking part in a seminar group. By specifying a parent relationship to the student group, this group additionally inherits all rights prior assigned to the student group. When a new group was created, the next step is assigning rights to each group. Rights mainly permit users accessing specific menus and sub-items (e.g., the student group is not allowed to access the user or user group editor). The third task of the user group editor is associate user groups to specific users. The basic idea behind this is to assign the supervision of (student) groups to specific users. In doing so, authors are able to assign tests to specific user groups and to gain access to their results. Thus, requirements UM-2 and UM-3 are also satisfied.

The *student information editor* allows administrators specifying data structures for storing and organizing of student information. Hereby, the administrator can define any complex structure as basis for configuring students' demographics by the authors. Category structures can also be defined arbitrarily deep. Section 4.5.4.2 mentioned that an established standard or specification should be used as basis for this kind of information in order to promote their syntactic and semantic interoperability. Currently, the *askMe!* system implements the categories of the IMS LIP 1.0.1 specification for structuring students' demographics (cf. Figure E.8). Consequently, this fulfills requirement UM-4.

The *student feature editor* is mainly provided for authors and allows configuring the user model of the *askMe!* system. The system provides by default a predefined set of user (student) features namely *knowledge*, *interests*, *goals*,

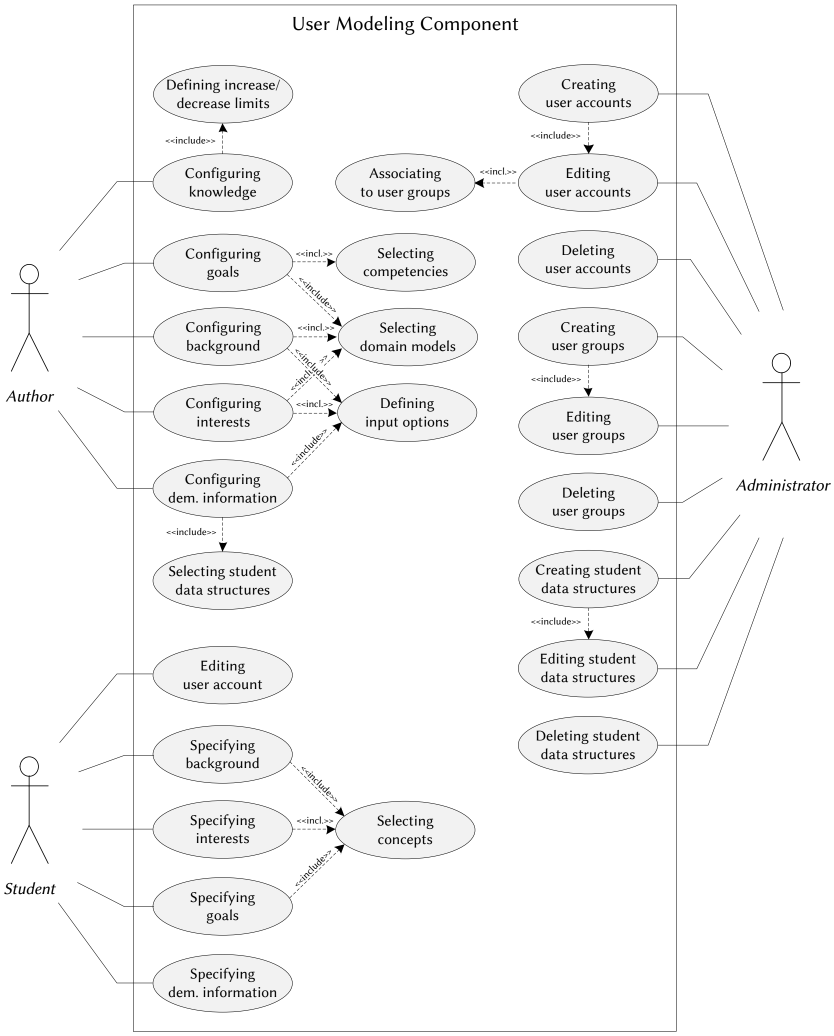


Figure E.7: Use case diagram of the user modeling component

background and *demographic information*. In terms of knowledge, the author needs to define two values between 0 and 1 that specify the extent to what the probability of students' knowledge can increase and decrease, respectively (cf. Section 4.5.4.2). These values are used by the adaptive testing engine component to compensate varying knowledge levels (e.g., caused by incorrect or unintended interactions) during testing. For that reason, the changes should be limited to a maximum and to a minimum. Basically, the *askMe!* system uses a positive approach that means that the value for the increase is greater than the value for the decrease. Regarding interests, authors need to select a domain model that students can use to specify their interests. This also applies to goals and background. In contrast, for specifying students' demographics, the author makes use of the data structure created with the student information editor. In addition, authors have also to define input options, which students can use to express their interests, background and demographic information. The input types currently implemented in the *askMe!* system are select and check boxes, radio buttons, input fields and text

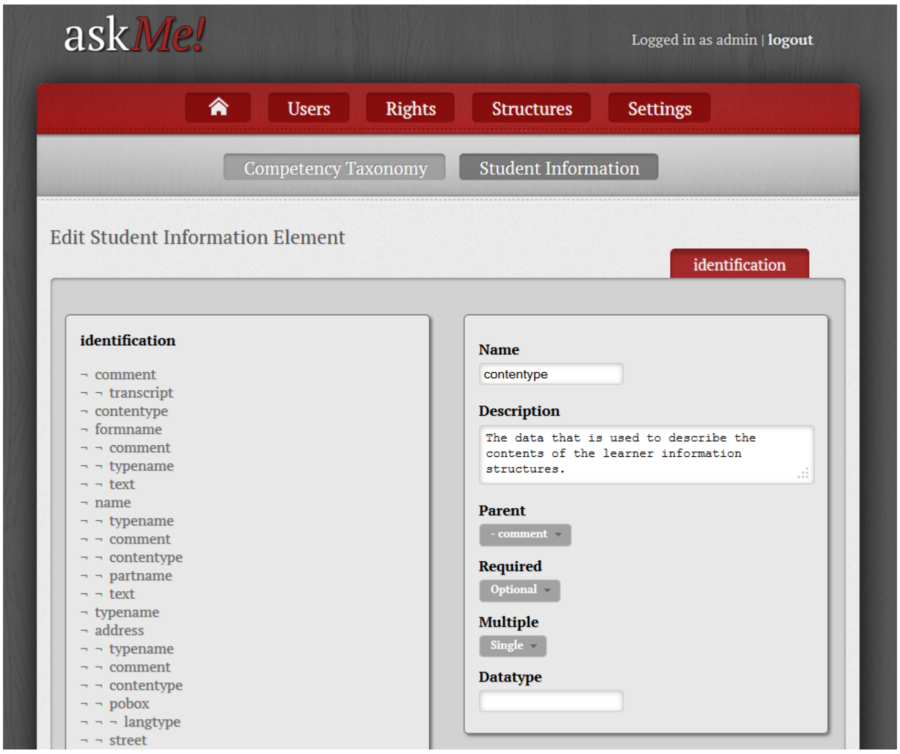


Figure E.8: askMe! student information editor

areas. For expressing goals, authors do not specify input options, instead, students make use of a predefined structure to specify the competencies they want to achieve. This structure (*competency taxonomy*) is defined once in the *askMe!* system and consists of dimensions and corresponding levels. By default, the system implements Bloom's taxonomy of the cognitive domain revised by Anderson and Krathwohl (cf. Section 2.2.6.1). Students also use this editor, however, they are provided with a varying (and limited) functionality. They are only able to initialize or update their interests, goals, background and demographic information using the domains or inputs options provided by the author (cf. Figure E.9). In this way, the remaining requirements UM-5 to UM-8 are also satisfied.

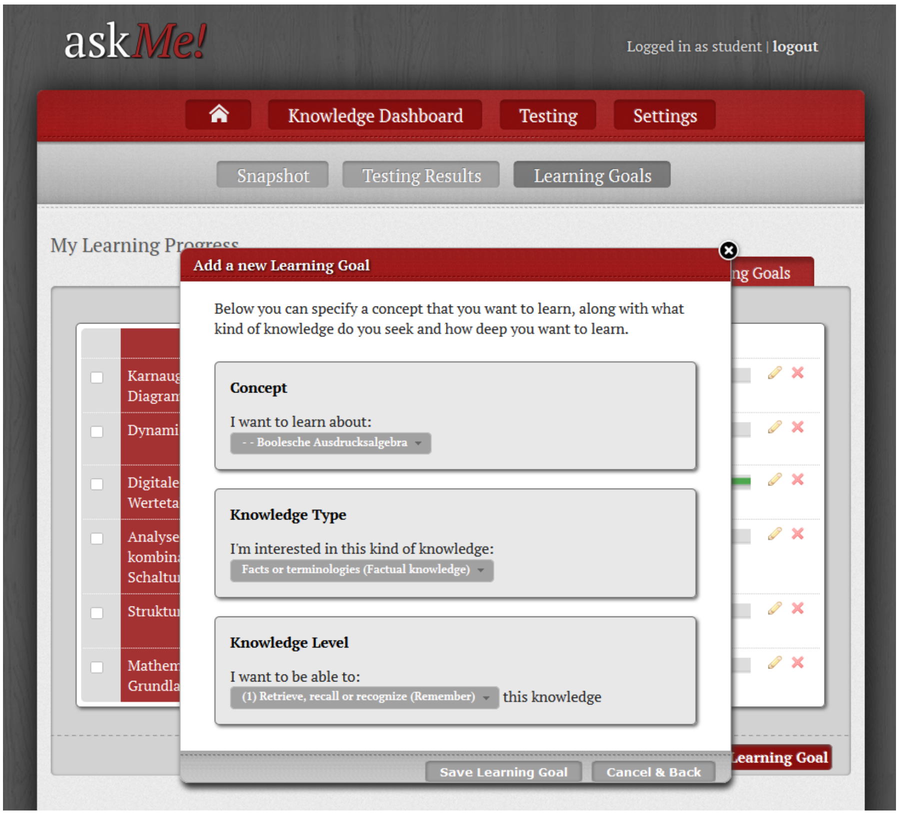


Figure E.9: askMe! student feature editor for administering learning goals

Domain Modeling Component Design and Implementation

This section focuses on the design and implementation of the domain modeling component. Figure E.10 depicts the component based on the structure presented in Section 4.2.4. In the following sections, the different layers are described in detail. Parts of the implementation were supported by Johannes Genthner in the course of his diploma thesis [Gen11].

Data Layer

The data layer of the domain modeling component is based on a relational database consisting of a set of database tables. They are represented by the following models:

- *Domain*: This model represents the information relevant for the domains managed by the component.
- *Concept*: This model represents the concepts underlying a domain.
- *Concept relationship*: This model represents all uni-directional relationships between two concepts.
- *Concept subscription*: This model represents subscriptions of domains.

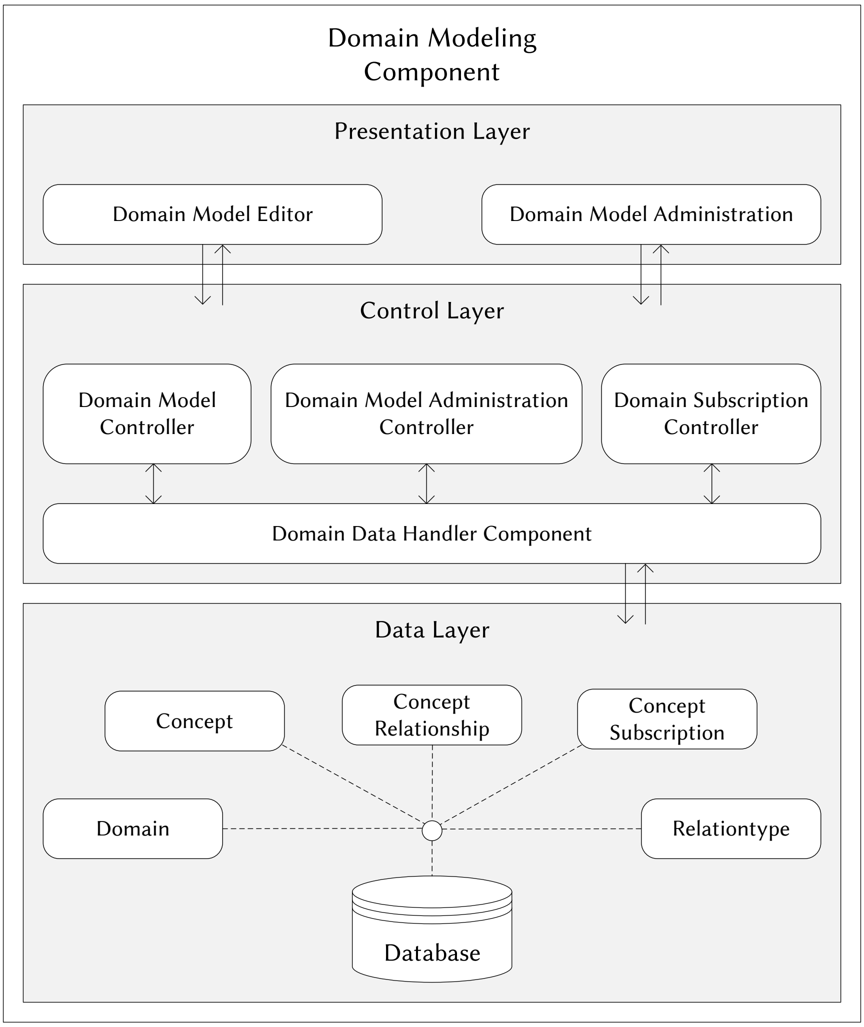


Figure E.10: Architectural structure of the domain modeling component

- *Relationtype*: This model represents the types of the relationships managed by the component.

The different models do not stand for their own, but they are linked together using different associations (cf. Figure E.11). Each domain (model) contains one or more concepts, whereas each concept belongs to one or more domains. A concept relationship exists between two different concepts and has a specific relation type. Furthermore, a relation type is not limited to one, but can be used by many concept relationships. Finally, each domain model can be subscribed by different users. The data structure of the *User* model is maintained by the user modeling component.

Control Layer

The control layer of the domain modeling component consists of a set of controllers and components, which provide the functionality associated with the control and workflow of the component. These are the *domain model*,

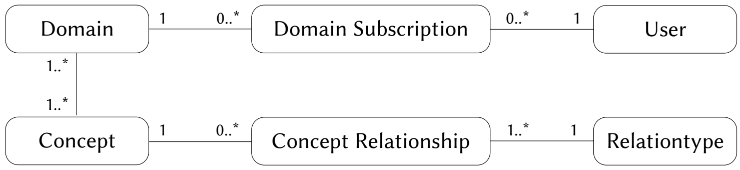


Figure E.11: Relations between the domain modeling component models

domain model administration and *domain subscription controller* as well as the *domain data handler component.*

In general, the *domain model, domain model administration* and *domain subscription controller* are responsible for receiving and preparing user inputs from the corresponding views so that the data handler component is able to process them. Furthermore, they initiate the rendering of the views, which are presented to the users. While the domain model controller provides all actions for creating, editing and deleting domain concepts, the domain model administration controller provides all actions for the overall administration of the domains such as creating and deleting of domain models. The domain subscription controller provides all actions for subscribing to domain models created by other authors.

The *domain data handler component* encapsulates all core functions, which are shared between the domain modeling component controllers. This includes reading domain models or parts of, saving new and deleting existing domain models. In doing this, the data handler instantiates all related models and make them available by a set of standardized data access functions. A further significant function of the data handler is the representation of fields of knowledge based on domain models. It uses the data and relationships of the different models to represent them in the form of an ontology. This is realized by a *top-down algorithm* that takes a *start concept* as input and searches for all child concepts, which have a *hierarchical* relation to this parent concept. This step is repeated until an optional *end concept* is reached or no child concept can be found anymore. By using the existing model relationships, the representation can also be enriched with *associative* relations. This representation can then be visualized by the own or integrated in another component.

Presentation Layer

The presentation layer of the domain modeling component consists of two main views (cf. Figure E.10), each encapsulates certain functionality. However, the views or the component as such are exclusively designed for being used by authors (domain experts). A graphical representation of the various ways an author can interact with the domain modeling component is depicted in Figure E.12.

The *domain model administration* enables the actual administration of the domain models. This includes editing the name and description of own models as well as deleting of own and subscribing to shared models. Furthermore,

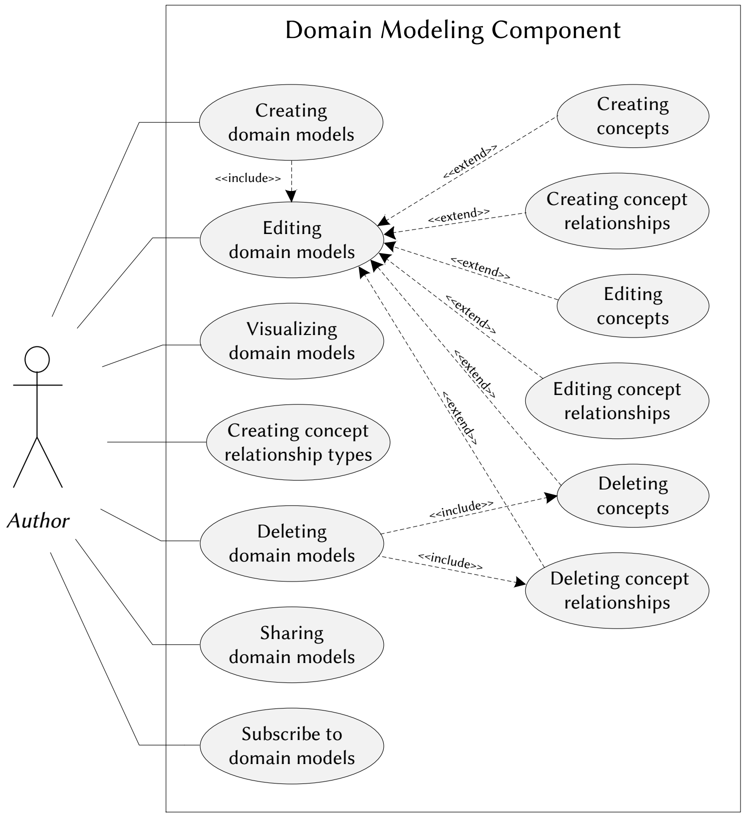


Figure E.12: Use case diagram of the domain modeling component

creating a new domain model is also initiated here. The overview of the models created by the author is presented as a table and located in the upper part of the page. The table entries provide information about the *title*, *description*, *creation* and *modification date*. Moreover, the fields *published* and *shared* allow the author to publish the domain model to be used by other components or to be subscribed by other authors, respectively. In this way, the requirements DM-6 and DM-7 derived from application scenario #1 are met (cf. Section 4.7.2). An overview about the models subscribed by the author is located in the lower part of the page. Here, the author can search and subscribe to new domain models or cancel existing subscriptions. In this way, the author can use domain models created by other authors without the need to create its own. Besides, this avoids duplication of domain models.

The *domain model editor* represents the core view of the domain modeling component. The functionality provided by this view allows creating and editing of a domain model and its elements. In order to intuitively support the user in dealing with complex domain structures, a user-friendly *tree view* visualization is provided by the editor. It corresponds to the intended list group selected as best suited (cf. Section 4.4.4.5). The editor is subdivided into three areas (cf. Figure E.13). The first area is the tree view located on the left hand side. It presents the concepts of the domain model and their *hierarchical* relationships and allows adding as well as deleting of concepts.

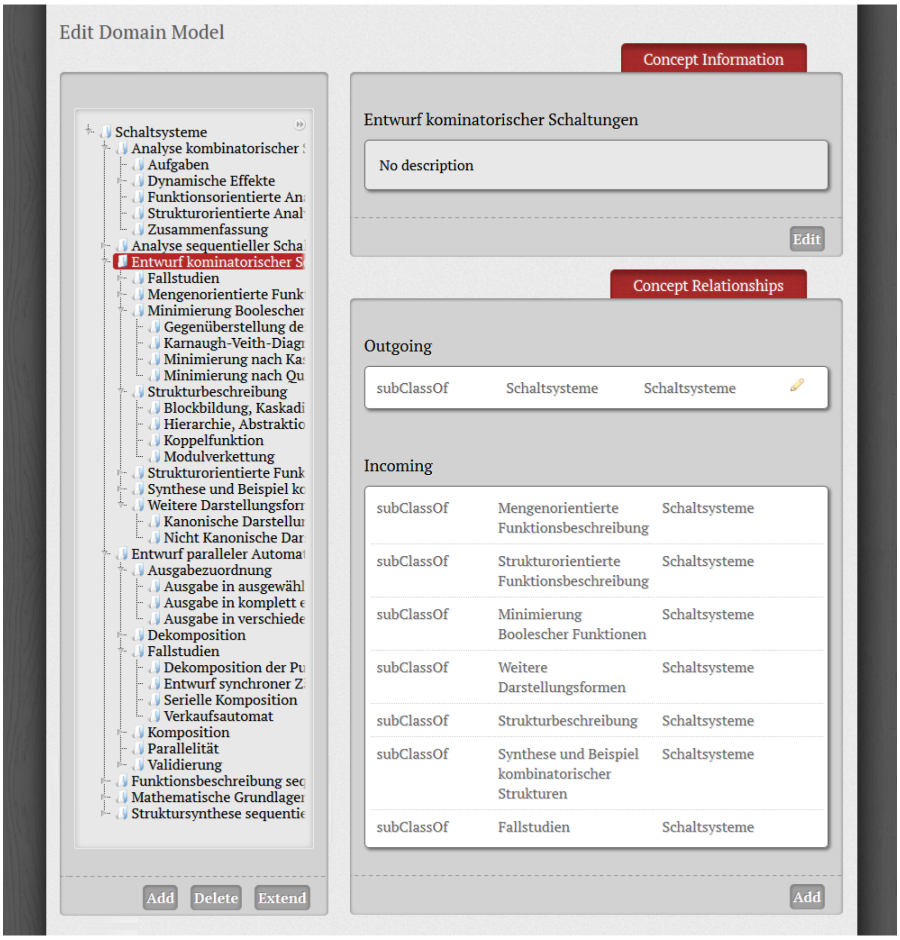


Figure E.13: askMe! domain model editor

Moreover, concepts in the list can be moved intuitively using drag and drop and the tree view can be extended to the full width of the page. This allows presenting even deep domain hierarchies in a clearly manner. The second area is located on the upper right hand side and shows information about a concept prior selected in the tree view. The information include a title and a description, whereas the description is entered using a rich-text editor. In this way, a description can also be enriched with images, videos and audios. The third area is directly below the concept information area. It allows managing *associative* relationships of the selected concept to other concepts. Besides, authors can freely define any type of relation to describe whose fields of knowledge sufficiently. Thus, the remaining requirements (i.e., DM-1 to DM-5 and DM-8) regarding the domain modeling component are also satisfied.

APPENDIX F

Integration with established Learning Environments

Launching the Test

When the test has been launched by the Learning Environment (LE), it needs to initialize its connection with the Adaptive Assessment System (AAS). Open Protocol for Accessing QUestion Engines (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.

Listing F.1: OPAQUE connection request message

```
<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:part name="cachedResources" 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 in order to adapt the assessment to these individuals. This kind of information can be submitted from the LE to the AAS using IEEE PAPI or 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. This identifier can be regarded as the key for further communication between the LE and the AAS. The following web service message corresponding to a connection request is defined by OPAQUE:

Listing F.2: OPAQUE connection response message

```
<wsdl:message name="startResponse">
    <wsdl:part name="questionSession" type="soapenc:string"/>
    <wsdl:part name="XHTML" type="soapenc:string"/>
    <wsdl:part name="CSS" type="soapenc:string"/>
    [...]
</wsdl:message>
```

The message part *questionSession* can be used to accommodate the session identifier of the test.

Communicating between AAS and LE

When the LE has established a connection with the AAS and 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:

Listing F.3: OPAQUE processing response message

```
<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 predetermined 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 *resources* 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. The web service message used to return the answer(s) to the AAS for further processing is as follows:

Listing F.4: OPAQUE processing request message

```
<wsdl:message name="processRequest">
    <wsdl:part name="questionSession" type="soapenc:string"/>
    <wsdl:part name="names" type="impl:ArrayOf_soapenc_string"/>
    <wsdl:part name="values" type="impl:ArrayOf_soapenc_string"/>
</wsdl:message>
```

The message part *questionSession* is purposed to accommodate the session identifier of the test. The answer(s) can be included in the second and third message part. Afterward, the AAS has to compare the received answer(s) with the correct answer(s) and decide how to proceed. For example, in case of an incorrect answer, the AAS could provide personalized feedback to guide the student to the correct solution or a slightly easier question addressing the same topic could instead be posed. It may happen that the student interrupts the assessment process for any reason. In this case, the LE has to inform the AAS that the test session is no longer needed. This can be realized using the OPAQUE *stopRequest* message, which only includes the session identifier. After receiving this message, the AAS can discard the corresponding test session.

Completing the Test

Eventually, the student answered 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 can be used by the LE, for example, for reporting and computing overall test scores. In order to inform the LE that the test has been completely answered, the AAS can use the *questionEnd* message part included in each *processResponse* message (cf. Listing F.3). 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 kind of information. The *results* message part is a complex structure and consists of the following elements:

Listing F.5: OPAQUE result message

```
<complexType name="Results">
    <sequence>
        <element name="actionSummary" type="soapenc:string"/>
        <element name="attempts" type="xsd:int"/>
        <element name="scores" type="impl:ArrayOfScore"/>
        [...]
    </sequence>
</complexType>
```

The element *actionSummary* can be used to accommodate the actions the student took in getting the answers. This can include the sequence of questions presented as well as the number of attempts per question. Although the element *attempts* is predetermined to record the number of attempts, it is only intended to hold the attempts of one question. In addition, the element *scores* is intended to accommodate the question scores of the test.

appendix g

First User Study – Tasks

Fraunhofer IDMT

Aufgaben – *askMe!* Evaluation (Autoren)

Lieber Teilnehmer,

Bitte lösen Sie nun folgende Aufgaben. Die Reihenfolge der Aufgaben ist festgelegt und soll wie vorgegeben eingehalten werden. Sie können jedoch eine Aufgabe jeder Zeit abbrechen und zur nächsten übergehen. Wenn Sie eine Aufgabe/Unteraufgabe beendet oder abgebrochen haben, informieren Sie bitte den Testleiter, bevor Sie zur nächsten übergehen.

Bei den nachfolgenden Aufgaben geht es darum, die Oberfläche kennenzulernen und erste Eindrücke zu äußern. Daher bitten wir Sie, all das auszusprechen, was Sie gerade denken während Sie parallel dazu die Aufgabe bearbeiten. Dabei liegt der Schwerpunkt auf Ihrem Empfinden bezüglich der Oberfläche. Sprechen Sie bitte laut aus, was Sie erwarten, was Ihnen gefällt und was Sie sich anders wünschen würden. Scheuen Sie sich dabei nicht, Kritik zu äußern und die Oberfläche kritisch zu bewerten.

1. Bitte loggen Sie sich mit folgenden Daten ein:

   Nutzername: *author*
   Passwort: *$askMe123*

2. Welche Hauptkategorien stellt Ihnen das System zur Verfügung?
   ☐ erledigt

Wenden Sie sich nun den im System hinterlegten Domänen-Modellen[1] (*Domains*) zu:

3. Nennen Sie die Anzahl der im System erstellten sowie abonnierten (*Subscribed*) Domänen-Modelle.
   ☐ erledigt
4. Bitte klicken Sie nun auf das Domänen-Model „Schaltsysteme“ und schauen Sie sich die Seite an, die nach dem Klick geöffnet wurde. Kehren Sie anschließend zur tabellarischen Übersicht zurück, ohne etwas zu bearbeiten.
   ☐ erledigt

Wenden Sie sich nun den im System hinterlegten Fragen (*Questions*) zu.

5. Verschaffen Sie sich einen Überblick über die im System erstellten Fragen und Fragetypen.
   ☐ erledigt
6. Schauen Sie sich nun die Bearbeitungsmöglichkeiten (Actions) rechts an. Was vermuten Sie hinter den einzelnen Funktionen?
   ☐ erledigt

[1] Domänen-Modelle können vereinfacht als eine Menge von Begrifflichkeiten und der zwischen ihnen bestehenden Beziehungen in einem bestimmten Gegenstandsbereich bezeichnet werden.

Seite 1

Fraunhofer IDMT

7. Erstellen Sie nun eine Lückentext-Frage (*Text Entry*) über „Boolesche Algebra"
    a. Nennen und beschreiben Sie die Frage beliebig.
    b. Erstellen Sie folgenden (*Fill-in*)Text:

    „Die boolesche Algebra ist nach George Boole benannt" und definieren Sie „Boole" als Platzhalter bzw. Lücke.

    c. Definieren Sie die Punktzahl (*Score*) der Frage beliebig.
    d. Definieren Sie ein beliebiges Feedback.
    e. Ordnen Sie Frage der Domäne „Schaltsysteme" und dem Konzept der „Mathematischen Grundlagen" zu.
    f. Ordnen Sie die Frage der Kompetenzstufe (*Knowledge level*) „Erinnern" und dem Wissenstyp (*Knowledge type*) „Faktenwissen" zu.
    g. Speichern Sie anschließend die Frage.
    h. Suchen Sie anschließend die soeben erstellte Frage in der Liste und lassen Sie sie sich anzeigen.

    ☐ erledigt

Wenden Sie sich nun den im System hinterlegten Tests (*Tests*) zu.

8. Suchen Sie den Test „Mathematische Grundlagen" und fügen Sie die soeben erstellte Frage diesem Test hinzu. Bewegen/Verschieben Sie anschließend diese Frage an den Anfang des Tests.
    ☐ erledigt
9. Schauen Sie sich nun die Struktur des Tests an. Was stellen Sie fest, wenn Sie mit der Maus auf die Frage zeigen? Was sagt ihnen die Position der einzelnen Fragen und Regeln über deren Präsentations-/Ausführungszeitpunkt aus?
    ☐ erledigt
10. Gehen Sie nun zur Übersicht der Tests (*List*) zurück und weisen Sie mittels der *Assignments*-Oberfläche der Studentin „Kerstin Bergmann" den Test „Mathematische Grundlagen" zu.
    ☐ erledigt
11. Überprüfen Sie die Zuweisung durch den Aufruf der Nutzer-Statistik-Übersicht (*Users*).
    ☐ erledigt

Vielen Dank!

Seite 2

First User Study – Questionnaires

Fraunhofer IDMT

Fragebogen – *askMe!* Evaluation

Lieber Teilnehmer,

Lesen Sie sich bitte die unten stehenden Fragen durch und tragen Sie Ihre Einschätzungen ein. Beachten Sie dabei, dass es **keine richtigen oder falschen Antworten** gibt. Diese Befragung ist **kein Test**. Es geht um Ihre **persönliche Meinung**. Es geht auch nicht um Schnelligkeit. Lassen Sie sich Zeit, die Fragen und auch die Anweisungen zu den Fragen in Ruhe durchzulesen und zu beantworten. Diese Befragung dient rein wissenschaftlichen Zwecken. Ihre Antworten werden selbstverständlich **streng vertraulich und anonym** behandelt.

Erfahrungen im Lernen und Lehren mit digitalen Medien (E-Learning)

Zunächst haben wir ein paar allgemeine Fragen zu Ihrer Erfahrung mit E-Learning:

1. Wie schätzen Sie Ihre Kenntnisse beim Lernen und Lehren mit digitalen Medien ein?

keine	wenige	durchschnittliche	gute	sehr gute
☐	☐	☐	☐	☐
Kenntnisse	Kenntnisse	Kenntnisse	Kenntnisse	Kenntnisse

2. Haben Sie bereits Erfahrungen mit anderen E-Learning Systemen und/oder Systemen für die Lernerfolgskontrolle?

☐	☐
ja	Nein

Wenn ja, welche: ____

3. Wie sehr sind Sie interessiert an neuer und innovativer Informations-/Kommunikations- oder Unterhaltungstechnologie?

☐	☐	☐	☐	☐
sehr gering	gering	mittel	stark	sehr stark

4. Wie oft nutzen Sie den Computer durchschnittlich in Ihrer Arbeitszeit?

☐	☐	☐	☐	☐
sehr selten	selten	mittel	häufig	sehr häufig

Fraunhofer IDMT

Beurteilung des *askMe!*-Systems

Im Folgenden sehen Sie Aussagen über *askMe!*. Geben Sie bitte an, in wie weit Sie diesen zustimmen. Um Ihnen das Ausfüllen zu erleichtern, ist der Fragebogen in Themen unterteilt. Sie können jederzeit an entsprechender Stelle **in *askMe!* nachsehen**.

Beispiel:

askMe! …	---	--	-	-/+	+	++	+++	*askMe!* …
ist schlecht	☐	☐	☐	☐	☒	☐	☐	ist gut

In dieser Beurteilung wird gefragt, wie gut bzw. wie schlecht *askMe!* ist. In diesem Fall beurteilen Sie *askMe!* zwar als gut, sehen jedoch noch Verbesserungsmöglichkeiten.

Bitte geben Sie nun Ihre Beurteilung von *askMe!* ab. Kreuzen Sie bitte nur einen Feld pro Zeile an:

5. Aufgabenangemessenheit

Unterstützte *askMe!* die Erledigung der Arbeitsaufgaben, ohne Sie unnötig zu belasten?

	askMe! …	---	--	-	-/+	+	++	+++	*askMe!* …
a1	ist kompliziert zu bedienen.	☐	☐	☐	☐	☐	☐	☐	ist unkompliziert zu bedienen.
a2	bietet nicht alle Funktionen, um die geforderten Aufgaben effizient zu bewältigen.	☐	☐	☐	☐	☐	☐	☐	bietet alle Funktionen, um die geforderten Aufgaben effizient zu bewältigen.
a3	erfordert überflüssige Eingaben.	☐	☐	☐	☐	☐	☐	☐	erfordert keine überflüssigen Eingaben.
a4	ist schlecht auf die Anforderungen der Aufgaben zugeschnitten.	☐	☐	☐	☐	☐	☐	☐	ist gut auf die Anforderungen der Aufgaben zugeschnitten.

Unterstützung bei der Erledigung der Arbeitsaufgaben ist Ihnen …

a5	sehr unwichtig	☐	☐	☐	☐	☐	☐	☐	sehr wichtig

Seite 2

6. Selbstbeschreibungsfähigkeit

Gibt Ihnen *askMe!* genügend Erläuterungen und ist sie in ausreichendem Maße verständlich?

	askMe! …	---	--	-	-/+	+	++	+++	*askMe!* …
s1	bietet einen schlechten Überblick über ihr Funktionsangebot.	☐	☐	☐	☐	☐	☐	☐	bietet einen guten Überblick über ihr Funktionsangebot.
s2	verwendet schlecht verständliche Begriffe, Bezeichnungen, Abkürzungen oder Symbole in Masken und Menüs.	☐	☐	☐	☐	☐	☐	☐	verwendet gut verständliche Begriffe, Bezeichnungen, Abkürzungen oder Symbole in Masken und Menüs.
s3	liefert in unzureichendem Maße Informationen darüber, welche Eingaben zulässig oder nötig sind.	☐	☐	☐	☐	☐	☐	☐	liefert in zureichendem Maße Informationen darüber, welche Eingaben zulässig oder nötig sind.
s4	bietet auf Verlangen keine situationsspezifischen Erklärungen, die konkret weiterhelfen.	☐	☐	☐	☐	☐	☐	☐	bietet auf Verlangen situationsspezifische Erklärungen, die konkret weiterhelfen.
s5	bietet von sich aus keine situationsspezifischen Erklärungen, die konkret weiterhelfen.	☐	☐	☐	☐	☐	☐	☐	bietet von sich aus situationsspezifische Erklärungen, die konkret weiterhelfen.

(Situationsspezifische) Erläuterungen in ausreichendem Maße sind Ihnen …

s6	sehr unwichtig	☐	☐	☐	☐	☐	☐	☐	sehr wichtig

Fraunhofer IDMT

7. Erwartungskonformität

Kommt *askMe!* durch eine einheitliche und verständliche Gestaltung Ihren Erwartungen und Gewohnheiten entgegen?

	askMe! …	---	--	-	-/+	+	++	+++	*askMe!* …
e1	erschwert die Orientierung durch eine uneinheitliche Gestaltung.	□	□	□	□	□	□	□	erleichtert die Orientierung durch eine einheitliche Gestaltung.
e2	lässt einen im Unklaren darüber, ob eine Eingabe erfolgreich war oder nicht.	□	□	□	□	□	□	□	lässt einen nicht im Unklaren darüber, ob eine Eingabe erfolgreich war oder nicht.
e3	informiert in unzureichendem Maße über das, was es gerade macht.	□	□	□	□	□	□	□	informiert in ausreichendem Maße über das, was es gerade macht.
e4	reagiert mit schwer vorhersehbaren Bearbeitungszeiten.	□	□	□	□	□	□	□	reagiert mit gut vorhersehbaren Bearbeitungszeiten.
e5	lässt sich nicht durchgehend nach einem einheitlichen Prinzip bedienen.	□	□	□	□	□	□	□	lässt sich durchgehend nach einem einheitlichen Prinzip bedienen.

Eine einheitliche und verständliche Gestaltung ist Ihnen …

e6	sehr unwichtig	□	□	□	□	□	□	□	sehr wichtig

8. Lernförderlichkeit/Erlernbarkeit

Ist *askMe!* so gestaltet, dass Sie sich gut darin einarbeiten konnten und bietet sie auch dann Unterstützung, wenn Sie neue Funktionen lernen möchten?

	askMe! …	---	--	-	-/+	+	++	+++	*askMe!* …
l1	erfordert viel Zeit zum Erlernen.	□	□	□	□	□	□	□	erfordert wenig Zeit zum Erlernen.
l2	ermutigt nicht dazu, auch neue Funktionen auszuprobieren.	□	□	□	□	□	□	□	ermutigt dazu, auch neue Funktionen auszuprobieren.
l3	erfordert, dass man sich viele Details merken muss.	□	□	□	□	□	□	□	erfordert nicht, dass man sich viele Details merken muss.
l4	ist so gestaltet, dass sich einmal Gelerntes schlecht einprägt.	□	□	□	□	□	□	□	ist so gestaltet, dass sich einmal Gelerntes gut einprägt.

Fraunhofer IDMT

	askMe! …	---	--	-	-/+	+	++	+++	*askMe!* …
l5	ist schlecht ohne fremde Hilfe oder Handbuch erlernbar.	☐	☐	☐	☐	☐	☐	☐	ist gut ohne fremde Hilfe oder Handbuch erlernbar.

Leichte Einarbeitung und Unterstützung, wenn Sie neue Funktionen lernen möchten, ist Ihnen …

l6	sehr unwichtig	☐	☐	☐	☐	☐	☐	☐	sehr wichtig

9. Steuerbarkeit

Können Sie die Art und Weise, wie Sie mit *askMe!* arbeiten, beeinflussen?

	askMe! …	---	--	-	-/+	+	++	+++	*askMe!* …
st1	bietet keine Möglichkeit, die Arbeit an jedem Punkt zu unterbrechen und dort später ohne Verluste wieder weiterzumachen.	☐	☐	☐	☐	☐	☐	☐	bietet die Möglichkeit, die Arbeit an jedem Punkt zu unterbrechen und dort später ohne Verluste wieder weiterzumachen.
st2	erzwingt eine unnötig starre Einhaltung von Bearbeitungsschritten.	☐	☐	☐	☐	☐	☐	☐	erzwingt keine unnötig starre Einhaltung von Bearbeitungsschritten.
st3	ermöglicht keinen leichten Wechsel zwischen einzelnen Menüs oder Masken.	☐	☐	☐	☐	☐	☐	☐	ermöglicht einen leichten Wechsel zwischen einzelnen Menüs oder Masken.
st4	ist so gestaltet, dass der/die Benutzer/in nicht beeinflussen kann, wie und welche Informationen am Bildschirm dargeboten werden.	☐	☐	☐	☐	☐	☐	☐	ist so gestaltet, dass der/die Benutzer/in beeinflussen kann, wie und welche Informationen am Bildschirm dargeboten werden.
st5	erzwingt unnötige Unterbrechungen der Arbeit.	☐	☐	☐	☐	☐	☐	☐	erzwingt keine unnötigen Unterbrechungen der Arbeit.

Die Steuerbarkeit von *askMe!* ist Ihnen …

st6	sehr unwichtig	☐	☐	☐	☐	☐	☐	☐	sehr wichtig

Fraunhofer IDMT

10. Fehlertoleranz

Bietet Ihnen die Software die Möglichkeit, trotz fehlerhafter Eingaben das beabsichtigte Arbeitsergebnis ohne oder mit geringem Korrekturaufwand zu erreichen?

	askMe! …	---	--	-	-/+	+	++	+++	*askMe!* …
f1	ist so gestaltet, dass kleine Fehler schwerwiegende Folgen haben können.	□	□	□	□	□	□	□	ist so gestaltet, dass kleine Fehler keine schwerwiegenden Folgen haben können.
f2	informiert zu spät über fehlerhafte Eingaben.	□	□	□	□	□	□	□	informiert sofort über fehlerhafte Eingaben.
f3	liefert schlecht verständliche Fehlermeldungen.	□	□	□	□	□	□	□	liefert gut verständliche Fehlermeldungen.
f4	erfordert bei Fehlern im Großen und Ganzen einen hohen Korrekturaufwand.	□	□	□	□	□	□	□	erfordert bei Fehlern im Großen und Ganzen einen geringen Korrekturaufwand.
f5	gibt keine konkreten Hinweise zur Fehlerbehebung.	□	□	□	□	□	□	□	gibt konkrete Hinweise zur Fehlerbehebung.

Die Fehlertoleranz von *askMe!* ist Ihnen …

f6	sehr unwichtig	□	□	□	□	□	□	□	sehr wichtig

Seite 6

Fraunhofer IDMT

Erfahrungen bei der Interaktion mit *askMe!*

Im Folgenden finden Sie Gegensatzpaaren von Eigenschaften, die *askMe!* haben kann. Entscheiden Sie möglichst spontan, welcher Begriff am ehesten zu *askMe!* passt bzw. deren Ausprägung.

Beispiel: attraktiv ☐ ☒ ☐ ☐ ☐ ☐ ☐ unattraktiv

Mit dieser Beurteilung sagen Sie aus, dass Sie die Software eher attraktiv als unattraktiv einschätzen.

11. Bitte geben Sie nun Ihre Einschätzung zu *askMe!* ab. Kreuzen Sie bitte nur einen Feld pro Zeile an:

1	unerfreulich	☐	☐	☐	☐	☐	☐	☐	erfreulich
2	unverständlich	☐	☐	☐	☐	☐	☐	☐	verständlich
3	kreativ	☐	☐	☐	☐	☐	☐	☐	phantasielos
4	leicht zu lernen	☐	☐	☐	☐	☐	☐	☐	schwer zu lernen
5	wertvoll	☐	☐	☐	☐	☐	☐	☐	minderwertig
6	langweilig	☐	☐	☐	☐	☐	☐	☐	spannend
7	uninteressant	☐	☐	☐	☐	☐	☐	☐	interessant
8	unberechenbar	☐	☐	☐	☐	☐	☐	☐	voraussagbar
9	schnell	☐	☐	☐	☐	☐	☐	☐	langsam
10	originell	☐	☐	☐	☐	☐	☐	☐	konventionell
11	behindernd	☐	☐	☐	☐	☐	☐	☐	unterstützend
12	gut	☐	☐	☐	☐	☐	☐	☐	schlecht
13	kompliziert	☐	☐	☐	☐	☐	☐	☐	einfach
14	abstoßend	☐	☐	☐	☐	☐	☐	☐	anziehend
15	herkömmlich	☐	☐	☐	☐	☐	☐	☐	neuartig
16	unangenehm	☐	☐	☐	☐	☐	☐	☐	angenehm
17	sicher	☐	☐	☐	☐	☐	☐	☐	unsicher
18	aktivierend	☐	☐	☐	☐	☐	☐	☐	einschläfernd
19	erwartungskonform	☐	☐	☐	☐	☐	☐	☐	nicht erwartungskonform
20	ineffizient	☐	☐	☐	☐	☐	☐	☐	effizient
21	übersichtlich	☐	☐	☐	☐	☐	☐	☐	verwirrend
22	unpragmatisch	☐	☐	☐	☐	☐	☐	☐	pragmatisch
23	aufgeräumt	☐	☐	☐	☐	☐	☐	☐	überladen
24	attraktiv	☐	☐	☐	☐	☐	☐	☐	unattraktiv
25	sympathisch	☐	☐	☐	☐	☐	☐	☐	unsympathisch
26	konservativ	☐	☐	☐	☐	☐	☐	☐	innovativ

Seite 7

Fraunhofer
IDMT

Persönliche Angaben

12. Abschließend bitten wir Sie noch um ein paar personenbezogene Angaben.

Alter: ______ Jahre

Geschlecht: □ weiblich □ männlich

Beruf: ____

Seite 8

First User Study – Detailed Data Analysis

Usability – Suitability for the Task

This first block of items measured whether the *askMe!* system supports the author in fulfilling its tasks (cf. Table G.1). The first item (a1) asked for whether the system is easy to use. All test persons agreed on that statement, whereas seven strongly agreed (++) and one person agreed (+). The second item (a2) aimed at identifying whether the test persons felt that the system provides all functions that are relevant for the task. The answers to this item obviously show that the test persons were fully satisfied with the functionality provided by the system. Two test person agreed (+), five strongly agreed (++) and one person absolutely agreed (+++) on that statement. The next item (a3) asked the test persons for their opinion about whether the system requires too much or even unneeded inputs. Even though the majority of test persons strongly agreed on the opposite (1x +, 3x ++, 3x +++), one test person absolutely agreed (+++) on this statement. This extreme outliner could be a hint that the item was misinterpreted by the test person. The fourth item (a4) aimed at figuring out whether the system is precisely tailored to the tasks. The result is clear: four test persons strongly agreed (++) and the other four test persons absolutely agreed (+++) on this statement. Finally, the test persons were asked to express how import they regard the suitability for the task (a5). The results show that all test persons regard this aspect as important, whereas four test persons stated it as important (+) and four as highly important (++). Figure G.1 shows the calculated mean and standard deviation value for each item. It can be seen that the results are mainly clustered closely to the mean value with the exception of the third item. This can be explained by the statistical outlier (1x −−−).

Item No.	N	Answers per rating scale item						
		−−−	−−	−	−/+	+	++	+++
a1	8	0	0	0	0	1	7	0
a2	8	0	0	0	0	2	5	1
a3	8	1	0	0	0	1	3	3
a4	8	0	0	0	0	0	4	4
a5	8	0	0	0	0	4	4	0

Table G.1: Answers per rating scale (suitability for the task)

Usability – Self-descriptiveness

The next block of items measured whether the *askMe!* system describes itself (cf. Table G.2). The first item (s1) asked if the system provides a good overview of the functions offered. From the positive answers (2x +, 4x ++, 2x +++) it

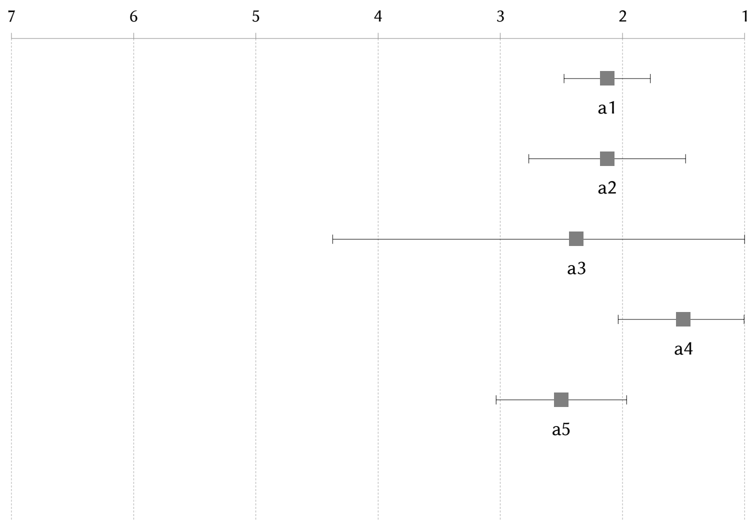


Figure G.1: Mean and standard deviation value per item (suitability for the task)

can be noted that the system is doing this in a satisfactory way. The second item (s2) concerned the statement whether the icons, symbols and words used are understandable. It seemed that it is the case because all test persons agreed on that statement (2x +, 5x ++, 1x +++). The next item (s3) aimed at figuring out whether the system provides sufficient feedback which inputs are allowed and necessary, respectively. The majority of test persons strongly agreed (++) on that statement and only one person agreed (+) and another person absolutely agreed (+++). The fourth and fifth item (s4, s5) concerned the provision of situation-specific explanations. The results show that the system provides good support if requested (1x −/+, 4x +, 2x ++, 1x +++), but better support on its own initiative (2x +, 4x ++, 2x +++). Finally, the test persons were asked to express how import they regard the suitability for the task (s6). From the results of this item, it can be shown that the test persons assessed this aspect as very important because two test persons stated it as important (+), four as very important (++) and two as highly important (+++). Figure G.2 reflects these results in the form of mean and standard deviation values.

Usability – Conformity with user expectations

The third block of items measured whether the *askMe!* system corresponds to predictable contextual needs of the authors and to commonly accepted conventions (cf. Table G.2). The first item (e1) aimed at figuring out whether the system is designed in a way that facilitates orientation. The current design obviously fulfilled the expectations of the majority of the test persons because all test persons provided positive answers (2x +, 3x ++, 3x +++) and no one responded in a negative way. The next item (e2) concerned the feedback to

ITEM NO.	N	ANSWERS PER RATING SCALE ITEM						
		−−−	−−	−	−/+	+	++	+++
s1	8	0	0	0	0	2	4	2
s2	8	0	0	0	0	2	5	1
s3	8	0	0	0	1	1	5	1
s4	8	0	0	0	1	4	2	1
s5	8	0	0	0	0	2	4	2
s6	8	0	0	0	0	2	4	2

Table G.2: Answers per rating scale (self-descriptiveness)

inputs and whether they were accepted or not. From the results, it can be seen that the system informs the user about the inputs they make (1x +, 4x ++, 1x +++). However, one negative and one undecided response indicated that there is some room for improvements. The third item (e3) also concerned the feedback of the system. Here, the test persons were asked whether it makes clear what it is doing in a respective situation. According to the results, a very positive result can be noticed. The results show that 75% of the test persons are satisfied with the feedback of the system and that the remaining two persons are undecided in this respect. The next item (e4) was created in order to check if the system reacts with unforeseen processing times. The test persons generally responded positively to this item. Most of the test persons marked the answer options + (1x), ++ (4x) and +++ (1x). Just two test persons responded in a negative way. This result can be explained by the fact that the *askMe!* system was running on a laptop due to organizational reasons. This laptop has limited hardware resources (cf. Section 6.2.4.1), which results in increased response times compared to a dedicated server. The fifth item (e5) aimed at figuring out whether the system can be used based on a uniform principle. It seems that the system fulfills this criterion very satisfactorily because almost all test persons strongly or absolutely agreed on this statement (1x +, 3x ++, 3x +++) and only one person remained undecided. Finally, the test persons were asked to express how import they regard the conformity with user expectations (e6). The results of this item show that the majority of test persons regard this aspect as highly important (1x −/+, 4x ++, 3x +++). Figure G.3 graphically represents the calculated mean and standard deviation value for each item. It can be seen that the mean value is always between 3 and 2, which means a positive agreement for the respective item. However, the few negative and undecided responses have led to slightly higher standard deviation values.

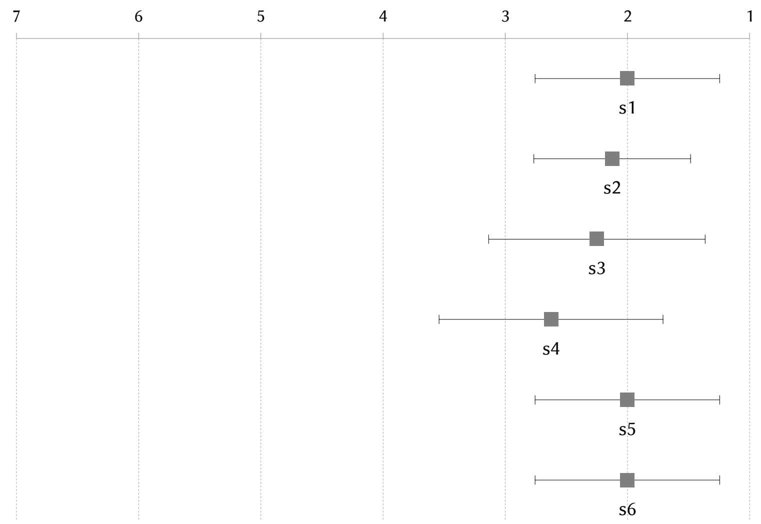


Figure G.2: Mean and standard deviation value per item (self-descriptiveness)

ITEM NO.	N	ANSWERS PER RATING SCALE ITEM						
		−−−	−−	−	−/+	+	++	+++
e1	8	0	0	0	0	2	3	3
e2	8	0	0	1	1	1	4	1
e3	8	0	0	0	2	3	2	1
e4	8	0	0	2	0	1	4	1
e5	8	0	0	1	0	1	3	3
e6	8	0	0	1	0	0	4	3

Table G.3: Answers per rating scale (conformity with user expectations)

Usability – Suitability for learning

The fourth block of items measured whether the *askMe!* system supports and guides the author in learning to use the system. The first item (l1) asked the test persons how much time it requires to get to know the system. From the results, it can be noted that it takes less time because all test persons provided positive answers (1x +, 6x ++, 1x +++). The second item (l2) concerned the ability of the system to encourage users to try out new functions. The results indicate that the system does not provide any hurdle because seven test persons provided positive results (2x +, 3x ++, 2x +++) and only one person remained undecided. The next item (l3) aimed at figuring out whether the system requires remembering a lot of details. The majority of test persons claimed the opposite (5x ++, 2x +++) and one only person had no opinion.

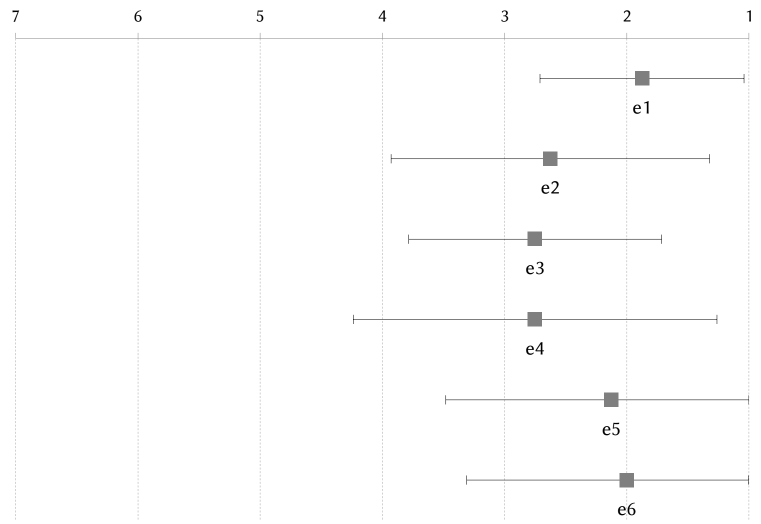


Figure G.3: Mean and standard deviation value per item (conformity with user expectations)

From the results, it can be stated that the system is mostly simple and intuitive to use and does not require remembering specific processes and procedures. The fourth item (l4) concerned the memorization of what the authors have once learned. The system obviously supports the memorization of facts and procedures because no test person stated the opposite (1x −/+, 1x +, 2x ++, 4x +++). The fifth item (l5) asked whether additional guidance is needed to learn to operate the system. The results of this item are also quite positive. 75% of all test persons stated that they do not need a manual or any other help to learn the system (4x ++, 2x +++), whereas the other test persons had no opinion (2x −/+). Finally, the test persons were asked to express how import they regard the suitability for learning (l6). The results show that all test persons regard this aspect as important, whereas two test persons stated it as extremely important (+++), four as highly important (++), one as important (+) and one test person remained undecided (−/+). Figure G.4 shows the calculated mean and standard deviation value for the respective item. It can be noticed that the mean value is always around 2, which means a uniform positive agreement for the respective item, but the few undecided responses have led to slightly higher standard deviation values.

Usability – Controllability

The fifth block of items measured whether the *askMe!* system allows authors to initiate and control the direction and pace of interactions. The first item (st1) asked the test persons whether the system allows to stop the work and to resume at the same point later on. Six out of eight test persons ascribe this possibility to the system (1x +, 4x ++, 1x +++) and the remaining two

item no.	n	answers per rating scale item						
		−−−	−−	−	−/+	+	++	+++
l1	8	0	0	0	0	1	6	1
l2	8	0	0	0	1	2	3	2
l3	8	0	0	0	1	0	5	2
l4	8	0	0	0	1	1	2	4
l5	8	0	0	0	2	0	4	2
l6	8	0	0	0	1	1	4	2

Table G.4: Answers per rating scale (suitability for learning)

persons had no opinion. The second item (st2) asked for opinions about the flexibility of the processing steps of the system. The results show that the system does not enforce any certain processing steps (2x −/+, 2x +, 3x ++, 1x +++). The third item (st3) was related to the switch between different menus and dialogs. According to the results, a very positive result can be noticed. It can be seen that this does not constitute a problem for any test persons (2x +, 2x ++, 3x +++). The fourth item (st4) concerned the influence of the authors on the presentation of information. The test persons generally responded positively. Most of the test persons marked the answer options + (2x), ++ (4x) and +++ (1x). Only one test persons remained undecided. The distribution of the answers indicated that the majority of test persons were able to easily influence the presentation of information according to their needs. The fifth item (st5) asked for unnecessary interruptions when using the system. It seems that the system does avoid this because no test person provided any indication for that. All test persons agreed on the statement that the system does not provide any unnecessary interruptions (1x +, 5x ++, 1x +++) and only one person had no opinion to this item. Finally, the test persons were asked to express how import they regard the controllability (st6). The results show that all test persons regard this aspect as highly important, whereas three test persons stated it as extremely important (+++), four as highly important (++) and one as important (+). Figure G.5 reflects these results in the form of mean and standard deviation values.

Usability – Error tolerance

The last block of items in this usability questionnaire focused on the ability of the *askMe!* system to achieve, despite erroneous inputs, the intended result with either no or minimal corrective actions by the author. The first item (f1) asked the test persons whether minor mistakes can cause serious consequences. The results indicate that the system is regarded as very robust towards minor mistakes because all test persons state the contrary (1x +, 5x ++, 1x +++). The second item (f2) concerned erroneous inputs. Five out of

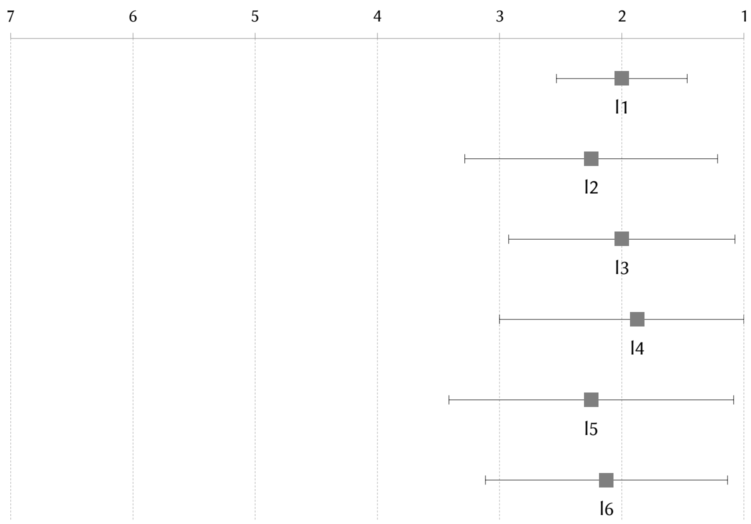


Figure G.4: Mean and standard deviation value per item (suitability for learning)

ITEM NO.	N	ANSWERS PER RATING SCALE ITEM						
		−−−	−−	−	−/+	+	++	+++
st1	8	0	0	0	2	1	4	1
st2	8	0	0	0	2	2	3	1
st3	8	0	0	0	1	2	2	3
st4	8	0	0	0	1	2	4	1
st5	8	0	0	0	1	1	5	1
st6	8	0	0	0	0	1	4	3

Table G.5: Answers per rating scale (controllability)

eight test person responded that the system immediately informs the user about erroneous inputs (1x +, 1x ++, 3x +++). In contrast, the remaining three persons had no opinion. The next item (f3) was created in order to check if the system provides understandable error messages. Even though four test persons agreed on this statement (2x ++, 2x +++), the other four persons remained undecided. As a result, it seems that the messages are not confusing, but they could be improved to make the meanings behind a little more clearly. The fourth item (f4) aimed at figuring out how much corrections are needed when imputing erroneous data. The results are similar the previous item. The half of the test persons provided a positive answer meaning that it requires less corrections when making erroneous inputs (3x ++, 1x +++), however, the other 50% of the test persons provided no clear answer. The same goes to the fifth item (f5), which concerned the instructions for troubleshooting.

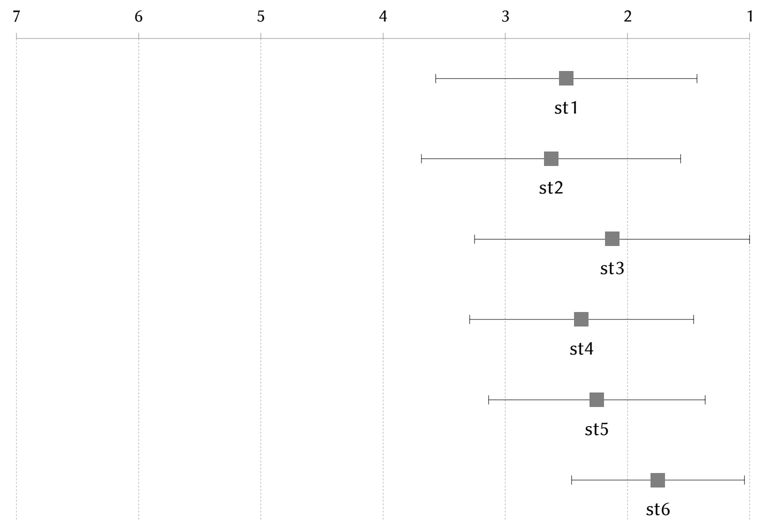


Figure G.5: Mean and standard deviation value per item (controllability)

Just as the last two items, four test persons provided positive answers (2x ++, 2x +++) and the other four persons were undecided. One reason for the results could be the fact that the tasks done by test persons may not address the error tolerance of the system sufficiently. The more erroneous inputs the authors make, the more they get a clear opinion about the error tolerance of the system. Finally, the test persons were asked to express how import they regard the error tolerance (f6). The results show that the majority of test persons regard this aspect as highly important (1x +, 1x ++, 3x +++) and that three test persons were not able to make a clear statement about this. Figure G.6 reflects these results in the form of mean and standard deviation values. It can be seen that the mean value is always between 3 and 1, which represents a very positive agreement for the respective item. The relatively high standard deviation values can be explained by the many undecided responses given by the test persons.

item no.	n	answers per rating scale item						
		−−−	−−	−	−/+	+	++	+++
f1	8	0	0	0	1	1	5	1
f2	8	0	0	0	3	1	1	3
f3	8	0	0	0	4	0	2	2
f4	8	0	0	0	4	0	3	1
f5	8	0	0	0	4	0	2	2
f6	8	0	0	0	3	1	1	3

Table G.6: Answers per rating scale (error tolerance)

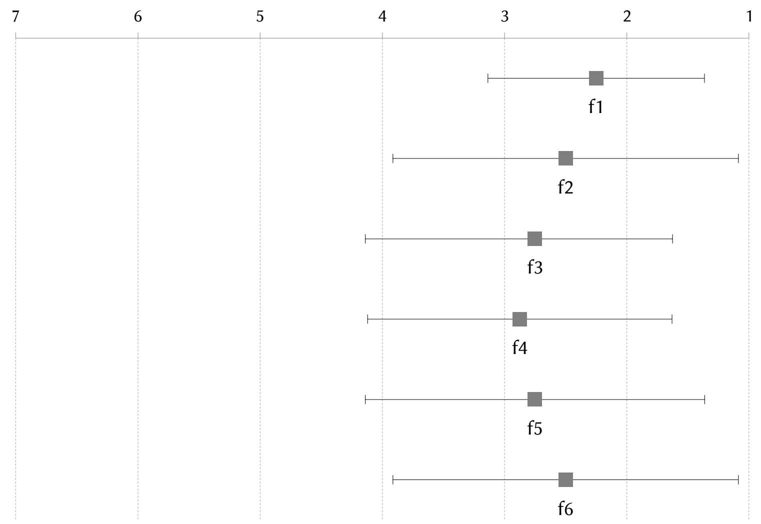


Figure G.6: Mean and standard deviation value per item (error tolerance)

User Experience – Attractiveness

The first UEQ scale expresses the general impression towards the *askMe!* system. In contrast to the other scales, attractiveness is measured using six items. Table G.7 lists the answers per rating scale. The first and second item show similar results. The authors mostly strongly agreed (++) that the system is giving joy (item no. 1) and makes a good impression (item no. 12). The third item (no. 14) asked whether the system is pleasing or unlikable. The results show that the most test persons rated the system as very (1x +) or highly pleasing (4x ++) and none of them as unlikable. However, three test persons remained undecided (-/+). The results of the fourth item (no. 16) show a similar picture compared with the first two items. All test persons rated the working with the system as pleasant (3x +, 5x ++). The last two items (no. 24 and 25) of this scale provided exactly the same results. Both, three test persons agreed (+) and three strongly agreed (++) that the working with the system is attractive and friendly, respectively. Besides, two persons had no clear opinions on these items.

ITEM NO.	N	ANSWERS PER RATING SCALE ITEM						
		−−−	−−	−	−/+	+	++	+++
1	8	0	0	1	1	1	5	0
12	8	0	0	0	0	2	5	1
14	8	0	0	0	3	1	4	0
16	8	0	0	0	0	3	5	0
24	8	0	0	0	2	3	3	0
25	8	0	0	0	2	3	3	0

Table G.7: Answers per rating scale (attractiveness)

User Experience – Perspicuity

The second scale measures whether it is easy to get familiar with the *askMe!* system. Table G.8 lists the authors' answers per item and rating scale. The results of the first two items (no. 2 and 4) clearly show that the system is understandable and easy to learn. All test persons agreed on the positive attributes and no one disagreed or was undecided. The results of the third item (no. 13) followed the previous results. Two test person agreed (+), five strongly agreed (++) and one person absolutely agreed (+++) that the working with the system is easy. Finally, the last item (no. 21) asked the test persons whether the system is clear or confusing. The results provided by the test persons are very good (1x +, 3x ++, 3x +++), even though one test person rated the system as little confusing (−). This outlier as well as the few undecided responses clouded the very positive picture a little. Nevertheless, this scale shows the highest mean value (cf. Table 6.4) of all user experience scales.

ITEM NO.	N	ANSWERS PER RATING SCALE ITEM						
		−−−	−−	−	−/+	+	++	+++
2	8	0	0	0	0	3	5	0
4	8	0	0	0	0	1	7	0
13	8	0	0	0	0	2	5	1
21	8	0	0	1	0	1	3	3

Table G.8: Answers per rating scale (perspicuity)

User Experience – Efficiency

The third scale aims at figuring out whether it is possible to use the *askMe!* system fast and efficient. Table G.9 presents the answers of the authors to each item. The first item (no. 9) asked the test persons whether the system is fast or slow. Surprisingly, the most people rated it as slow (1x −−−, 1x −−, 2x −) and only two persons stated the opposite (1x +, 1x ++). This result confirms the result of the usability questionnaire with respect to the processing times (self-descriptiveness). As mentioned before, one reason could be the laptop on which the tasks were performed. The second item (no. 20) was created in order to check whether the system works efficiently or not. The results show a clear picture. 88% of all test persons considered the system as efficient (3x +, 4x ++) and the remaining person had no clear opinion. The next item (no. 22) provides similar results. Six out of 8 test persons agreed on the statement that the system is very practical to handle (1x +, 4x ++, 1x +++) and the remaining people were undecided. The fourth and last item (no. 23) asked if the system provides an uncluttered user interface. That seems to be the case because 100% of the test persons provided a strong agreement with this statement (1x +, 3x ++, 4x +++).

ITEM NO.	N	ANSWERS PER RATING SCALE ITEM						
		−−−	−−	−	−/+	+	++	+++
9	8	1	1	2	2	1	1	0
20	8	0	0	0	1	3	4	0
22	8	0	0	0	2	1	4	1
23	8	0	0	0	0	1	3	4

Table G.9: Answers per rating scale (efficiency)

User Experience – Dependability

The fourth scale measures the feeling of being in control of the interaction provided by the *askMe!* system. Table G.10 presents the answers of the authors to each item. The first item (no. 8) aimed at figuring out whether the system reacts predictable or not. According to the results, this seems to be the case because all test persons provided positive answers (5x +, 2x ++, 1x +) to this item. The second item (no. 11) concerned the ability of the system to give help. The system obviously provides assistance to the users because the results are unanimous. One test person agreed (+), six strongly agreed (++) and one person absolutely agreed (+++) to this statement. The third item (no. 17) asked for opinions on the security of the system. No test person expressed any doubt, instead six test person had a strong confidence in the security of the system (2x +, 4x ++). The fourth item (no. 19) asked the test persons whether the system meets their expectations. The results show that for the majority of test persons, the system met their expectations in a very satisfactory manner (2x +, 4x ++, 1x +++). No one stated the opposite and only one test persons provided an unbiased answer.

ITEM NO.	N	ANSWERS PER RATING SCALE ITEM						
		−−−	−−	−	−/+	+	++	+++
8	8	0	0	0	0	5	2	1
11	8	0	0	0	0	1	6	1
17	8	0	0	0	2	2	4	0
19	8	0	0	0	1	2	4	1

Table G.10: Answers per rating scale (dependability)

User Experience – Stimulation

The fifth scale measures whether the user feel motivated to further use the *askMe!* system. Like most of the other scales, stimulation is measured using four items. Table G.11 shows the answers per rating scale. The first item (no. 5) asked if the system is valuable or not. From the results, it can be seen that almost all test persons went along with that (4x +, 3x ++) and that only one person was undecided. The second item (no. 6) asked whether the system excites or bores. Although the number of positive answers (2x +, 1x ++) is slightly larger than those of negative answers (1x −), the most test persons had no opinion or have chosen an unbiased answer (4x −/+). This result could be explained by the fact that the most test persons had no experience with this kind of e-learning/assessment software and thus no opportunity for comparison. The third item (no. 7) aimed at figuring out whether the system is interesting or not. The results of this item are very good because 5 out of 8 test persons found it interesting to absolutely interesting (2x +, 1x ++, 2x +++). The remaining test persons remained undecided (3x −/+). The last item (no. 18) concerned the motivational aspects of the *askMe!* system. For this item, the number of positive answers (1x +, 1x ++, 2x +++) is significantly larger than those of negative answers (1x −). It seems that the system is interesting, even though three undecided answers were also recognized.

User Experience – Novelty

The sixth scale measures whether the *askMe!* system is innovative and creative. Table G.12 presents the answers of the authors to each item. The first item (no. 3) asked the test persons whether the system is creative or dull. 75% of all test persons provided positive answers to this statement (4x +, 2x ++) and only one person stated the opposite (−). The second item (no. 10) concerned the originality of the system. The number of positive answers (3x +, 1x ++) is equal to the sum of negative (2x −−) and undecided answers (2x −/+). The originality provided by the system thus seems to be undoubtedly

Item No.	N	Answers per rating scale item						
		−−−	−−	−	−/+	+	++	+++
5	8	0	0	0	1	4	3	0
6	8	0	0	1	4	2	1	0
7	8	0	0	0	3	2	1	2
18	8	0	0	1	3	1	1	2

Table G.11: Answers per rating scale (stimulation)

available, but should be more highlighted. The third item (no. 15) asked the test persons whether the system is usual or cutting edge (technology). 75% of all test persons were of the opinion that the *askMe!* system represents a high level of development in terms of e-assessment technology. Only one person expressed doubts (−) and another person was undecided. Item no. 26 aimed at figuring out whether the system (from the authors' point of view) is innovative or conservative. The number of positive answers (1x +, 2x ++) is slightly larger than those of negative answers (1x −, 1x −−), but the most test persons (3x) had no opinion or have chosen an unbiased answer. From the results, it can be said that the system provides innovative approaches, which were obviously recognized by the test persons. However, due to the fact that the majority of test persons were undecided, it seems that the innovative character of the system needs to be more emphasized. Summarized it can be stated that with respect to the novelty the test persons did not provide a consistent picture, even though the positive responses outweigh the negative ones. This inconsistency is also reflected in the high standard deviation value (cf. Table 6.4). Finally, Figure G.7 graphically shows the mean and standard deviation value per each user experience item.

Item No.	N	Answers per rating scale item						
		−−−	−−	−	−/+	+	++	+++
3	8	0	0	1	1	4	2	0
10	8	0	2	0	2	3	1	0
15	8	0	0	1	1	4	2	0
26	8	0	1	1	3	1	2	0

Table G.12: Answers per rating scale (novelty)

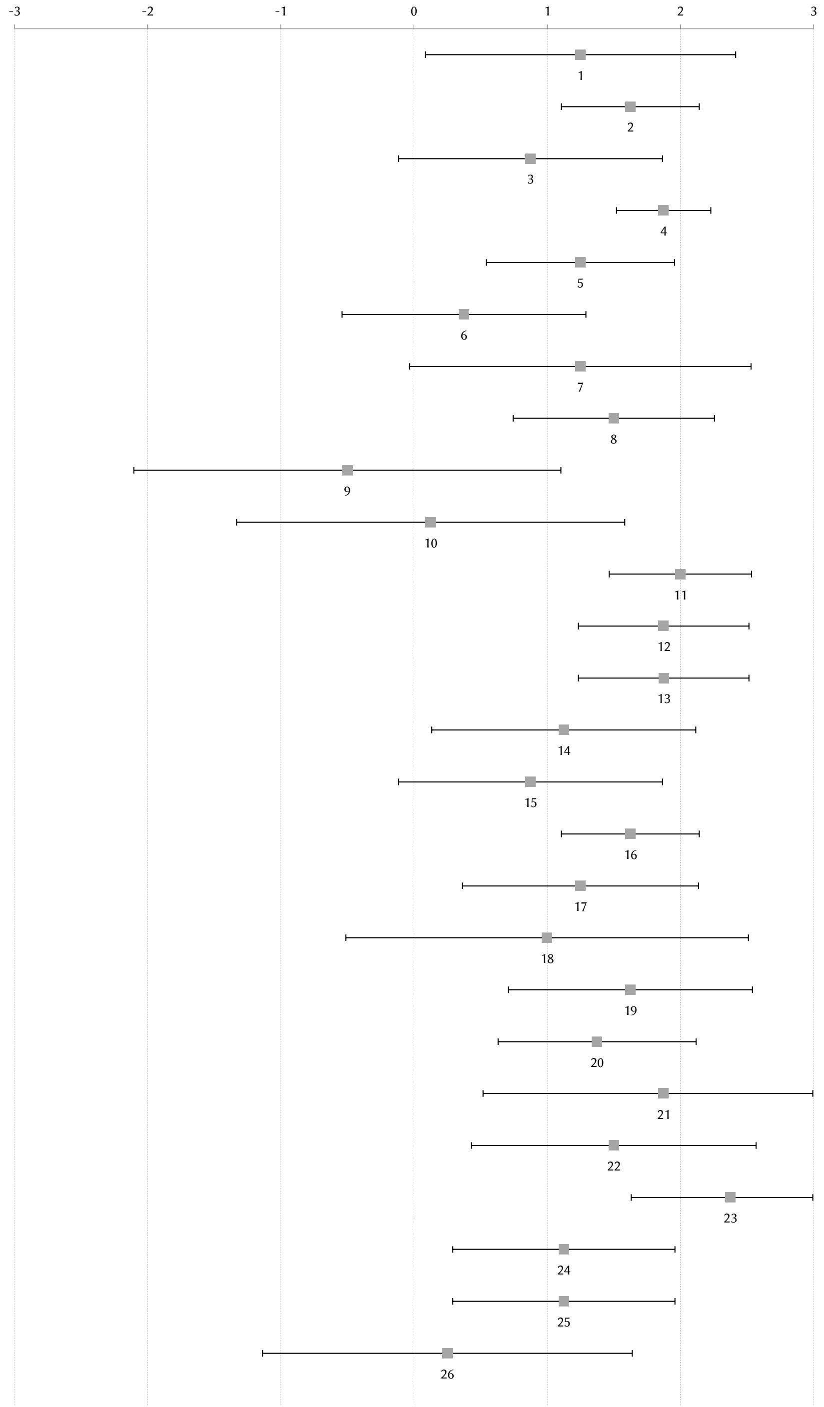


Figure G.7: Mean and std. deviation value per user experience item (first user study)

First User Study – Thinking Aloud Comments

TASK NO.	COMMENTS
1	—
2	• (+) The elements are clearly arranged • (+) The buttons are big and clearly visible • (▷) Use the whole width of the screen • (▷) Move settings on the top next to login
3	• (+) I like the AJAX loading • (▷) Show subscribe/unsubscribe button only when domain models are selected
4	• (–) The column for the concept hierarchy is too narrow • (▷) Use a progress indicator to bridge the load time
5	• (+) Icons for the question types are very easy to read and attractively designed • (+) Sorting functionality is helpful and intuitive • (▷) Provide search functionality • (▷) Provide filtering functionality (e.g., according to the question type or description) • (▷) A preceding selecting of a domain or concept would be helpful • (▷) Show the page selection also above the table
6	—
7	• (+) Fill-in blanks have the same width and don't reveal the length of the correct answer • (+) Loading icon after saving the question very helpful

7	• (▷) Rename the button "Add question" to "Create a new question" • (▷) Vertical align all input fields • (▷) Enlarge the fill-in-text input field • (▷) A click outside the pop-up dialog should also close the dialog • (▷) Allow setting a maximum score to ensure consistency • (▷) Provide an additional "Save" button on top of the page • (▷) Show the feedback created when previewing questions
8	• (+) Interactivity of the visualization is clearly identifiable • (+) Drag and drop is intuitive and easy • (▷) An extended sorting (e.g., after creation date) when adding questions would be helpful • (▷) Align the list when adding questions with the questions list seen before • (▷) Question preview when selecting questions for adding would be good • (▷) A click on the row should also select the checkbox • (▷) Show recent created questions
9	• (+) Structure of the test is clearly identifiable • (▷) Preview of the whole tests would be good (put the author in the position of the student)
10	• (+) Assigning tests to students is easy • (▷) "Select all" button when assigning tests to students would be helpful
11	• (▷) Show the message "not yet started" instead of "−−"

Table G.13: Thinking aloud comments from the first user study

+ = Positive comment
− = Critical comment
▷ = Suggestion for improvement

APPENDIX H

Second User Study - Questionnaires

Fraunhofer IDMT

Fragebogen – *askMe!* Evaluation

Lieber Teilnehmer,

Lesen Sie sich bitte die unten stehenden Fragen durch und tragen Sie Ihre Einschätzungen ein. Beachten Sie dabei, dass es **keine richtigen oder falschen Antworten** gibt. Diese Befragung ist **kein Test**. Es geht um Ihre **persönliche Meinung**. Es geht auch nicht um Schnelligkeit. Lassen Sie sich Zeit, die Fragen und auch die Anweisungen zu den Fragen in Ruhe durchzulesen und zu beantworten. Diese Befragung dient rein wissenschaftlichen Zwecken. Ihre Antworten werden selbstverständlich **streng vertraulich und anonym** behandelt.

Allgemeine Einstellung gegenüber Informations- und Kommunikationstechnik

Zunächst haben wir ein paar allgemeine Fragen zu Ihrer Einstellung gegenüber Informations- und Kommunikationstechnik:

1= stimme voll zu; 2= stimme eher zu; 3= teils/teils; 4= stimme eher nicht zu; 5= stimme gar nicht zu

		1 (++)	2 (+)	3 (0)	4 (-)	5 (--)
is1	Im Internet zu bezahlen ist riskant.					
ii1	Es ist für mich interessant, wie einzelne Teile eines technischen Gerätes zusammenwirken, damit es seine Aufgabe erfüllt.					
io1	Ich mag Design im Allgemeinen.					
ie1	Ich erlerne neue technische Geräte durch ausprobieren.					
ic1	Ich habe keine (ausreichende) Schulung mit Computern erhalten.					
ia1	Es frustriert mich, technische Geräte/Computer zu benutzen.					
io2	Ich erwarte einen guten, persönlichen Service, wenn ich ein Gerät oder Produkt kaufe.					
ii2	Ungewöhnliche Geräte finde ich spannend und interessant.					
ie2	Ich probiere alle Funktionen und Möglichkeiten aus, die technische Geräte bieten.					
ia2	Ich fühle mich verängstigt, wenn ich einen Computer bedienen muss.					
io3	Bei technischen Geräten achte ich auf solide Verarbeitung.					
ic2	Informationen von Anleitungen oder Mitarbeitern im technischen Service verstehe ich nicht.					
ie3	Es gefällt mir, neue technische Geräte/Computer-anwendungen zu erlernen.					

Fraunhofer IDMT

		1 (++)	2 (+)	3 (0)	4 (-)	5 (--)
io4	Für mich sind bekannte Marken beim Kauf eines Gerätes oder Produktes wichtig.					
is2	Ich benutze kein Online-Banking.					
io5	Das Aussehen eines Gerätes spielt für mich eine große Rolle.					
ia3	Ich fühle mich hilflos, wenn ich technische Geräte benutzen muss.					
ia4	Es frustriert mich, Computerprogramme zu benutzen.					
ie4	Es macht mir Spaß, neue Geräte / Programme zu erkunden.					
ic3	Bis jetzt hatte ich keine Möglichkeit, technische Geräte bedienen zu lernen.					
io6	Ich kaufe keine No-Name-Produkte.					
ia5	Bisher habe ich mich ängstlich gefühlt, wenn ich Computerprogramme benutzen musste.					
is3	Ich kaufe nur im Internet ein.					
ie5	Ich probiere sehr gerne neue technische Geräte aus.					
ic4	So genannte Organizer (Kalender, Adressen usw. auf einem mobilen Gerät) nutze ich nie.					
ii3	Technik hat mich schon immer fasziniert.					

Lernmotivation

Des Weiteren ein paar allgemeine Fragen zu Ihrer Einstellung gegenüber dem Lernen:

1= stimme voll zu; 2= stimme eher zu; 3= teils/teils; 4= stimme eher nicht zu; 5= stimme gar nicht zu

		1 (++)	2 (+)	3 (0)	4 (-)	5 (--)
m1	Ich bevorzuge Lernstoff, der mich wirklich herausfordert, so dass ich Neues kennen lerne.					
m2	Ich bevorzuge Lernstoff, der meine Neugier weckt, auch wenn er schwierig zu erlernen ist.					
m3	Am meisten stellt es mich beim Lernen zufrieden, wenn ich versuche, den Inhalt des Lernmaterials so gründlich wie möglich zu verstehen.					
m4	Wenn ich die Möglichkeit habe, Lernaufgaben selbst auszuwählen, bevorzuge ich solche, von denen ich viel lernen kann, auch wenn ich damit vielleicht keine gute Beurteilung erziele.					

Fraunhofer IDMT

Bewertung des *askMe!*-Systems

1. Wie würden Sie *askMe!* System beurteilen? Sie können Schulnoten bis 5 vergeben:

1= sehr gut; 2= gut; 3= befriedigend; 4= ausreichend; 5= mangelhaft

		1	2	3	4	5
a1	Stabilität / Funktionieren der Technik					
a2	Zugänglichkeit					
a3	Geschwindigkeit					
a4	Informations- und Textdesign					
a5	Navigation					
a6	Bedienung / Benutzerfreundlichkeit					
a7	Aktualität der Inhalte					
a8	Qualität der Inhalte					

2. Wie wichtig waren Ihnen die folgenden Methoden für die Prüfungsvorbereitung?

1= am wichtigsten; 2= sehr wichtig; 3= wichtig; 4= nicht sehr wichtig; 5= unwichtig

		1 (++)	2 (+)	3 (0)	4 (-)	5 (--)
p1	Studentische Arbeitsgruppen					
p2	Lehrbücher					
p3	Lernen mit Vorlesungsskript (des Lehrenden)					
p4	Lernen mit eigenem Script (Mitschrift aus der Vorlesung)					
p5	Lernen mit Mitschriften anderer Studierender					
p6	Weiterführende Literatur					
p7	Nutzung von *askMe!*					
p8	Besuch der Vorlesung					
p9	Übungsklausuren / Klausuren aus anderen Semestern					

Fraunhofer
IDMT

3. Wie oft haben Sie *askMe!* genutzt?

h1 ☐ sehr häufig ☐ häufig ☐ manchmal ☐ selten ☐ sehr selten

4. Wie viel der Inhalte von *askMe!* haben Sie zur Prüfungsvorbereitung verwendet?

h2 ☐ 0 – 20% ☐ 20 – 40% ☐ 40 – 60% ☐ 60 – 80% ☐ 80 – 100%

5. Wie stark stimmen Sie folgenden Aussagen zum Einsatz von *askMe!* in der Lehrveranstaltung zu?

1= stimme voll zu; 2= stimme eher zu; 3= teils/teils; 4= stimme eher nicht zu; 5= stimme gar nicht zu

		1 (++)	2 (+)	3 (0)	4 (-)	5 (--)
l1	Der Einsatz von *askMe!* in dieser Lehrveranstaltung war mit eindeutigen Zielen verbunden.					
l2	Die in dieser Lehrveranstaltung im Rahmen von *askMe!* zu bearbeitenden Aufgaben waren für dieses Medium geeignet.					
l3	Ich wünsche mir, dass in vergleichbaren Lehrveranstaltungen in Zukunft verstärkt *askMe!* eingesetzt wird.					
l4	Der Einarbeitungsaufwand in *askMe!* war nicht zu groß.					

6. Wie stark stimmen Sie folgenden Aussagen zum Einsatz von *askMe!* im Vergleich zu Veranstaltungen ohne *askMe!* zu?

1= stimme voll zu; 2= stimme eher zu; 3= teils/teils; 4= stimme eher nicht zu; 5= stimme gar nicht zu

		1 (++)	2 (+)	3 (0)	4 (-)	5 (--)
l5	Insgesamt habe ich durch die Kombination von Präsenzterminen und *askMe!* mehr gelernt als bei einer vergleichbaren reinen Präsenz-Veranstaltung.					
l6	Durch den Einsatz von *askMe!* habe ich mich in dieser Lehrveranstaltung intensiver mit den Inhalten auseinandergesetzt.					
l7	Durch den Einsatz von *askMe!* in dieser Lehrveranstaltung habe ich selbständiger gelernt.					
l8	Durch den Einsatz von *askMe!* konnte ich meinen Lernfortschritt in dieser Lehrveranstaltung besser überprüfen.					

Fraunhofer IDMT

Erfahrungen bei der Interaktion mit *askMe!*

Im Folgenden finden Sie Gegensatzpaaren von Eigenschaften, die *askme!* haben kann. Entscheiden Sie möglichst spontan, welcher Begriff am ehesten zu *askMe!* passt bzw. deren Ausprägung.

<u>Beispiel:</u> attraktiv ☐ ☒ ☐ ☐ ☐ ☐ ☐ unattraktiv

Mit dieser Beurteilung sagen Sie aus, dass Sie *askMe!* eher attraktiv als unattraktiv einschätzen.

7. Bitte geben Sie nun Ihre Einschätzung zu *askMe!* ab. Kreuzen Sie bitte nur einen Feld pro Zeile an:

1	unerfreulich	☐	☐	☐	☐	☐	☐	☐	erfreulich
2	unverständlich	☐	☐	☐	☐	☐	☐	☐	verständlich
3	kreativ	☐	☐	☐	☐	☐	☐	☐	phantasielos
4	leicht zu lernen	☐	☐	☐	☐	☐	☐	☐	schwer zu lernen
5	wertvoll	☐	☐	☐	☐	☐	☐	☐	minderwertig
6	langweilig	☐	☐	☐	☐	☐	☐	☐	spannend
7	uninteressant	☐	☐	☐	☐	☐	☐	☐	interessant
8	unberechenbar	☐	☐	☐	☐	☐	☐	☐	voraussagbar
9	schnell	☐	☐	☐	☐	☐	☐	☐	langsam
10	originell	☐	☐	☐	☐	☐	☐	☐	konventionell
11	behindernd	☐	☐	☐	☐	☐	☐	☐	unterstützend
12	gut	☐	☐	☐	☐	☐	☐	☐	schlecht
13	kompliziert	☐	☐	☐	☐	☐	☐	☐	einfach
14	abstoßend	☐	☐	☐	☐	☐	☐	☐	anziehend
15	herkömmlich	☐	☐	☐	☐	☐	☐	☐	neuartig
16	unangenehm	☐	☐	☐	☐	☐	☐	☐	angenehm
17	sicher	☐	☐	☐	☐	☐	☐	☐	unsicher
18	aktivierend	☐	☐	☐	☐	☐	☐	☐	einschläfernd
19	erwartungskonform	☐	☐	☐	☐	☐	☐	☐	nicht erwartungskonform
20	ineffizient	☐	☐	☐	☐	☐	☐	☐	effizient
21	übersichtlich	☐	☐	☐	☐	☐	☐	☐	verwirrend
22	unpragmatisch	☐	☐	☐	☐	☐	☐	☐	pragmatisch
23	aufgeräumt	☐	☐	☐	☐	☐	☐	☐	überladen
24	attraktiv	☐	☐	☐	☐	☐	☐	☐	unattraktiv
25	sympathisch	☐	☐	☐	☐	☐	☐	☐	unsympathisch
26	konservativ	☐	☐	☐	☐	☐	☐	☐	innovativ

Fraunhofer
IDMT

Persönliche Angaben

8. Abschließend bitten wir Sie noch um einige personenbezogene Angaben:

d1 Alter: _____ Jahre

d2 Geschlecht: ☐ weiblich ☐ männlich

d3 Studiengang: ______________________________

d4 Fachsemester: ______

Second User Study – Detailed Data Analysis

Attitude towards ICT

The first questionnaire aimed at assessing students' attitude towards ICT. Table H.1 gives an overview about the internal consistency reliability of the different scales. With the exception of competence, all scales can be considered as sufficient consistent. A factor analysis was performed in order to explain the negative reliability value. It could be noticed that one item (ic4) was mostly answered in a positive way, even though the item was formulated in a negative way. It could be happened that the students had problems with this double negative formulation as occurred with negations in combination with a Likert scale (totally disagree). As a consequence, the items of competence scale need to be interpreted very carefully. The analysis of the ICT questionnaire results showed that the students were mostly *experienced* users (23x). In addition, five students can be categorized as *playful* and one students as *interested amateur*. The remaining two students did not provide any data. Experienced users belong to the classes with highest competence, exploration and interest in combination with lowest anxiety and skepticism due to its willingness to take a risk inherent to exploratory learning. In contrast, playful users are defined as exploratory and interested as the experienced user, but quite skeptical and users classified as interested amateurs have a moderate exploration, anxiety and competence as well as less skepticism.

SCALE	ITEMS	RELIABILITY
Exploration	5	0,50
Skepticism	3	0,54
Anxiety	5	0,64
Interest	3	0,57
Competence	4	-0,19
Surface	6	0,48

Table H.1: Reliability of the ICT questionnaire scales

Learning Motivation

Following the ICT questionnaire, the students were asked to express their motivation towards learning. Table H.2 and Figure H.1 show the distribution of students' answers to each item. The first item (m1) asked the test persons whether they prefer content that is challenging in order to learn something new. The results show a clear picture, almost all test persons agreed on this statement (22x +, 3x ++) and only four test persons had no opinion. The second item (m2) also concerned the content. The results are even more clearly than those of the previous item. Almost all students agreed on the statement that they prefer content that arouses curiosity even it is difficult to learn (14x +, 12x ++). Summarized it can be stated that the majority of test persons is open to challenging and difficult content if something new can be learned or the content arouses curiosity. Besides, no student stated the opposite, only a few were undecided in this respect. This confirms the need for personalization as provided by the *askMe!* system. The third item (m3) addressed the satisfaction while learning and asked the test persons whether they are satisfied while learning when they try to understand the content as much as possible. It seems that not all test persons agree on this statement because two persons provided negative answers and nine had no opinion to this item. However, the remaining test persons provided positive answers to this item (9x +, 9x ++). The last item addressing the learning motivation (m4) asked the test persons whether they would select content they could learn quite a lot from them even if they might not achieve the best grading. As a result, the number of test persons provided positive answers (15x +, 4x ++) is much higher than those of negative answers (4x −). Besides, six test persons remained undecided.

ITEM NO.	N	ANSWERS PER RATING SCALE ITEM				
		−−	−	−/+	+	++
m1	29	0	0	4	22	3
m2	29	0	0	3	14	12
m3	29	0	2	9	9	9
m4	29	0	4	6	15	4

Table H.2: Answers per rating scale (learning motivation)

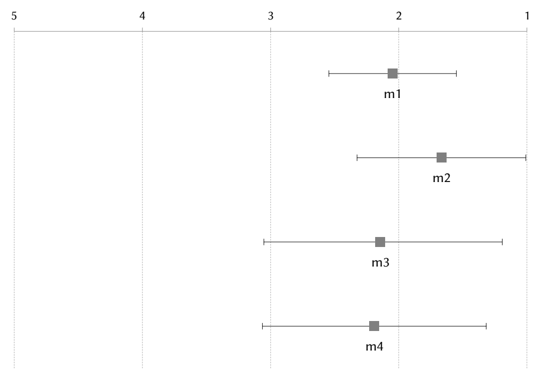


Figure H.1: Mean and std. deviation values per item (learning motivation) *askMe! Questionnaire*

The next items in the questionnaire concerned the *askMe!* system itself. The first set of items (a1-a8) aimed at identifying students' opinion about the robustness, navigation and accessibility of the *askMe!* system. Figure H.2 shows the mean and standard deviation value for each item. It can be seen that the mean value is always between 2 (strongly agree) and 3 (absolutely agree), which represents a uniform positive agreement to the respective item.

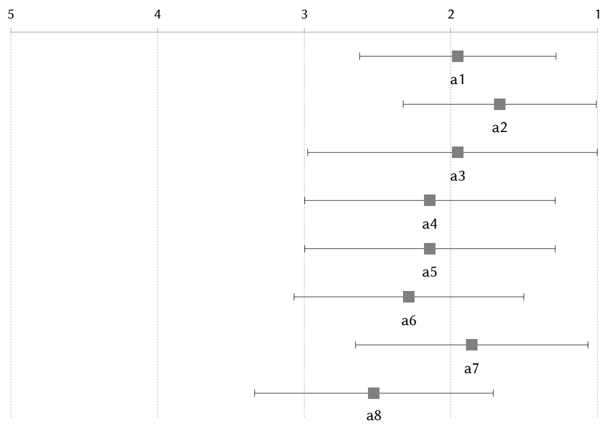


Figure H.2: Mean and std. deviation value per item (*askMe!* questionnaire)

The next set of items (p1-p9) aimed at identifying how the test persons had prepared themselves for the exam either by student working groups, textbooks, lecture script/notes, etc. and/or by using the *askMe!* system. As a result, the most important means for students' exam preparation were the lecture script (n = 21; mean = 1,90; std.dev. = 0,70) and own notes (n = 21; mean = 1,90; std.dev. = 1,14), followed by the attendance to the lectures (n = 21; mean = 2,33; std.dev. = 1,56) and the use of the *askMe!* system (n = 21; mean = 2,95; std.dev. = 0,86). The distribution of students' answers per rating scale is shown in Table H.3. It also shows that textbooks (p1) and further literature (p6) did not play a role for their individual exam preparation in this course.

The next two items (h1 and h2) asked the test persons for their perceived frequency and extent of use of the *askMe!* system. In order to compare the real and the perceived frequency and extent of use, the number of logins (frequency) and the number of question completed (extent) were mapped to the five-point scale as used by the questionnaire. The mapping scheme is presented in Table H.4. After mapping the values to the rating scale, a correlation analysis was performed. It is not surprisingly that the real and the perceived values show a great correlation. In terms of the frequency of use, the correlation could be considered as significant (n = 21; r = 0,524; p = 0,015) and with respect to the extent of use as highly significant (n = 21; r = 0,892; p = 0,000).

item no.	n	answers per rating scale item				
		−−	−	−/+	+	++
p1	21	3	3	2	3	10
p2	21	6	5	4	4	2
p3	21	0	0	4	11	6
p4	21	1	1	3	6	10
p5	21	3	6	3	6	3
p6	21	8	5	6	2	0
p7	21	1	4	9	7	0
p8	21	0	2	0	15	4
p9	21	3	1	8	3	6

Table H.3: Answers per rating scale (exam preparation)

rating scale item	number of logins	number of question completed
1	$4 < x$	$35 < x$
2	$3 < x \leqslant 4$	$25 < x \leqslant 35$
3	$2 < x \leqslant 3$	$15 < x \leqslant 25$
4	$1 < x \leqslant 2$	$5 < x \leqslant 15$
5	$0 < x \leqslant 1$	$0 < x \leqslant 5$

Table H.4: Mapping scheme

The next eight items (l1-l8) asked the test persons for their opinion about the general use of the *askMe!* system in this or in other courses. Table H.5 shows the answer distribution for each item and Figure H.3 reflects these results in the form of mean and standard deviation values. The first item (l1) asked for whether the system was introduced in this course with clear objectives. This seemed to be the case because almost all test persons agreed on this statement (13x +, 1x ++) and only one test person remained undecided. The next item (l2) aimed at identifying whether the system was appropriate for the course content. The results show that the majority of test person answered in the affirmative (7x +, 4x ++). Furthermore, four test persons provided a negative response to this statement. The third item (l3) measured whether the *askMe!* system is desirable for use in other courses. From the results of this item, it can be shown that the test persons would like to use the system also in other courses because 20 out of 21 test persons answered positive (13x +, 7x ++) and only one person remained undecided. The fourth item (l4) was created in order to check whether the system is easy to understand and requires little time to become familiar with it. This seemed to be the case because also 20 out of 21 test persons agreed on this statement. Even though the number of positive statement is the same as in the previous item, the responses are significantly more positive (6x +, 14x ++). The fifth item (l5) concerned the learning with the *askMe!* system. Although the number of positive answers (6x +, 1x ++) is higher than those of negative answers (3x –), however, the majority of test persons chose an unbiased answer. The persons obviously had problems assessing their learning. The next item (l6) asked for whether the test persons more intensively dealt with the course and its content after using the *askMe!* system. The system provides students an overview about strengths and weaknesses (self-assessment) so that they could subsequently be addressed very specifically (e.g., by further literature). Nine test persons confirmed this kind of learning support and only 5 persons disagreed with this statement. Besides, seven test persons had no clear opinions. The seventh item (l7) concerned students' self-regulated learning. As a result, the majority of test persons stated that by using the *askMe!* system, they gradually became more autonomous in terms of learning (+). Only five person claimed the opposite (–) and seven persons remained undecided. The last item (l8) aimed at figuring out whether the system supports the students in monitoring their own learning progress. This is one of the main objectives of the *askMe!* system and it is very gratifying to receive positive responses from the test persons. More than 52% of the test persons agreed (+) or strongly agreed (++) on this statement and only 14% claimed the opposite (–). Furthermore, the remaining seven test persons did not provide a clear answer.

User Experience

Table H.6 shows the answer distribution per user experience scale for the second user study. In comparison with the results of the first user study, it can be seen that the mean values are mostly slightly below the values of the first

Item No.	N	Answers per rating scale item				
		−−	−	−/+	+	++
I1	21	0	3	4	13	1
I2	21	0	4	6	7	4
I3	21	0	0	1	13	7
I4	21	0	0	1	6	14
I5	21	1	3	10	6	1
I6	21	0	5	7	9	0
I7	21	0	5	8	7	1
I8	21	0	3	7	8	3

Table H.5: Answers per rating scale (*askMe!* course integration)

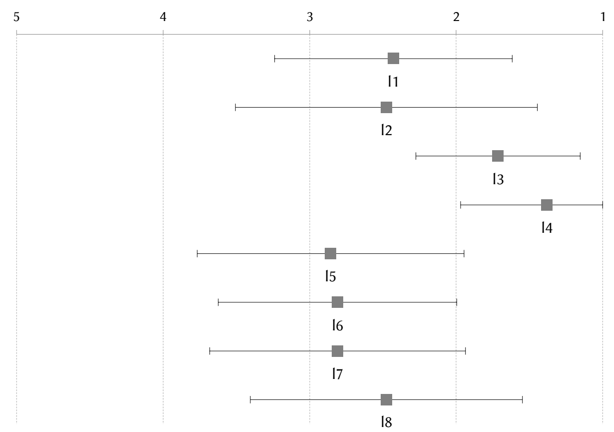


Figure H.3: Mean and std. deviation value per item (*askMe!* course integration) evaluation phase. The standard deviation values are also slightly higher than in the first user study. This can be justified by the fact that the number of test persons was more than twice as much. Due to the fact that these persons have different opinions and answer tendencies, the mean values (always between 0,5 and 1,5) look not as positive as they really are. But, not all mean values are below the ones obtained from the first user study. The results of item no. 6 (+0,17), no. 9 (+1,36), no. 10 (+0,36) and no. 26 (+0,37) show higher values than in the first user study. As a consequence, the students rated the system as more exciting, faster, inventive and innovative than the authors.

Particular notable is the result of the first efficiency item (no. 9), which asked the test persons whether the system is *fast* or *slow*. In the first user study, the mean value of the results of this item was negative (mean = -0,5; std.dev. = 1,6). This could be explained by the fact that the *askMe!* system was running on a laptop with limited hardware resources. In the second user study, however, the *askMe!* system was running on a dedicated server with much more hardware resources. Surprisingly, the responses to this item were now much better (mean = 0,86; std.dev. = 1,43). Consequently, the concerns regarding the processing times of the *askMe!* system no longer applies. Finally, Figure H.4 graphically shows the mean and standard deviation value per each user experience item.

ITEM NO.	N	ANSWERS PER RATING SCALE ITEM							SCALE
		−−−	−−	−	−/+	+	++	+++	
1	21	0	1	2	2	7	8	1	ATT
2	21	0	0	1	0	11	5	4	PER
3	21	0	2	2	2	10	5	0	NOV
4	21	1	1	0	4	3	10	2	PER
5	21	1	0	0	6	8	5	1	STI
6	21	0	0	2	6	10	3	0	STI
7	21	0	0	1	2	12	5	1	STI
8	21	0	0	1	5	7	5	3	DEP
9	21	1	0	0	3	7	10	0	EFF
10	21	0	1	0	8	6	6	0	NOV
11	21	0	0	0	2	8	9	2	DEP
12	21	0	0	1	2	8	10	0	ATT
13	21	0	0	0	2	10	8	1	PER
14	21	0	0	0	10	5	6	0	ATT
15	21	0	1	3	3	12	2	0	NOV
16	21	0	0	0	3	8	9	1	ATT
17	21	0	1	3	3	4	8	2	DEP
18	21	0	0	3	6	5	6	1	STI
19	21	0	0	2	5	6	8	0	DEP
20	21	0	0	3	2	8	7	1	EFF
21	21	0	2	1	1	8	8	1	PER
22	21	0	0	0	12	7	2	0	EFF
23	21	0	1	0	1	12	7	0	EFF
24	21	0	0	1	3	11	5	1	ATT
25	21	0	0	1	3	13	4	0	ATT
26	21	0	0	3	3	10	5	0	NOV

Table H.6: Answer distribution per user experience scale (second user study)

ATT = Attractiveness
EFF = Efficiency
STI = Stimulation
PER = Perspicuity
DEP = Dependability
NOV = Novelty

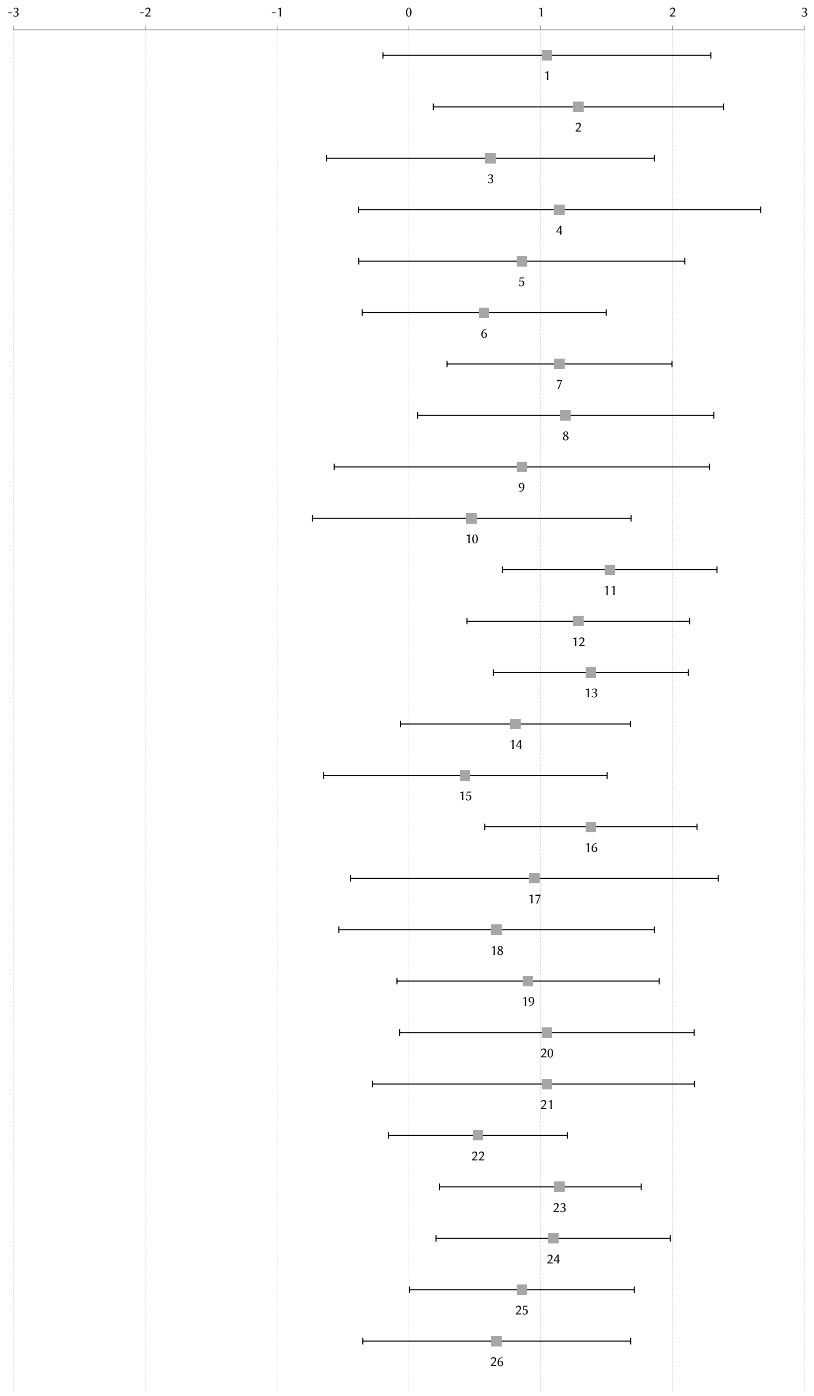


Figure H.4: Mean and std. deviation value per user experience item (second user study)