

Karlsruhe Reports in Informatics 2012,7

Edited by Karlsruhe Institute of Technology,
Faculty of Informatics
ISSN 2190-4782

Design and Implementation of a Tool for Peer Assessment

Tobias Kimmerle, Christian Hütter, and Klemens Böhm

2012



Fakultät für **Informatik**

Please note:

This Report has been published on the Internet under the following
Creative Commons License:

<http://creativecommons.org/licenses/by-nc-nd/3.0/de>.

Design and Implementation of a Tool for Peer Assessment

Technical Report 2012-7

Tobias Kimmerle, Christian Hütter, and Klemens Böhm

Institute for Program Structures and Data Organization
Karlsruhe Institute of Technology, Germany
Email: `klemens.boehm@kit.edu`

Abstract. This report describes the design and implementation of several modules for the course management system *Moodle*. These modules facilitate a learning process where students create multiple choice questions, which are combined into quizzes. There, students answer and rate questions according to different criteria. An incentive mechanism motivates students to rate questions.

Keywords: Moodle, multiple choice questions, peer assessment

1 Introduction

The modules described in this report facilitate a learning process where students create and answer multiple choice questions (MCQ). For peer assessment, students grade questions of other students. To evaluate the quality of peer assessment, questions are also graded by teachers. An incentive mechanism computes bonus points for grading questions.

The *Question Creation Module* combines the creation of questions by students with the grading of questions by teachers. The *Quiz Module* allows teachers to publish questions as a quiz where students are able to answer and rate questions. As a motivation for rating questions, a high score table was implemented which shows the number of bonus points for each participant.

1.1 Moodle

Moodle [1] is an open source¹ course management system and an e-learning Platform. *Moodle* is an acronym for *Modular Object-Oriented Dynamic Learning Environment*. With over 70.000 registered² systems with over 50 million users [2] it is used for online courses as well as a supplement to traditional face-to-face courses.

The most important feature of Moodle is to create and manage virtual classrooms where learning resources and practical tasks, so-called *Activities*, can be published. The following examples show some typical applications of Moodle:

¹ GNU General Public License

² It is possible to register a Moodle system to submit anonymous usage statistics

Publication of learning resources: Teachers upload lecture slides, static text pages as well as other files. Similarly, standards for interchangeable e-learning content, such as *SCORM*³ packages are supported.

Communication: Message boards and live chats allow learners to communicate with each other. A wiki engine can be provided to allow students to create learning material collaboratively.

Assignments: Tasks can be assigned to students, who submit solutions by file upload or web forms. Teachers grade submissions and provide feedback to the learner. Certain assignments, such as answering multiple choice questions, can be graded automatically.

Grade management: Students' test scores (both online and offline) can be recorded and aggregated into an overall assessment. A gradebook can be made available as an overview to the learner.

It should be noted that Moodle is implemented as a classic web application. Today's popular architectural patterns such as Model View Controller⁴ are not applied in Moodle. Object-oriented design patterns can be found but are not consistently used. Smaller changes are planned by the developer community. For example, Moodle 2.1 introduced an object-oriented interface to access its question database.

1.2 Plugin Support

New features can be added to an installation through a variety of plugins, in Moodle called *modules*. It is possible to define the necessary database tables for a module through Moodle's own XML description format for database schemata. The most important types of modules are:

Activity modules: Some of Moodle's features, such as message boards, quizzes, and the wiki engine are implemented as activity modules. They are logically separated from the core system but can be executed as a web application integrated in the Moodle user interface.

Question Types: Questions in Moodle's database are instances of a particular data type that extend the parent data type *Moodle question*. Supported question types for example are multiple choice questions and essays. The user interfaces for question input and output as well as their internal data structure are defined in these modules.

Display Blocks: In Moodle simply called *Blocks*, they serve as a display element at the left or right side of a page. Blocks that are commonly used, are for example the navigation menu, a calendar and the element to display the latest news on a course page.

This report also describes some minor enhancements, such as different import and export formats as well as support for integration with other software systems.

³ *Shareable Content Object Reference Model: Collection of standards and specifications for web-based e-learning*

⁴ Structural separation of data model, presentation layer and program control

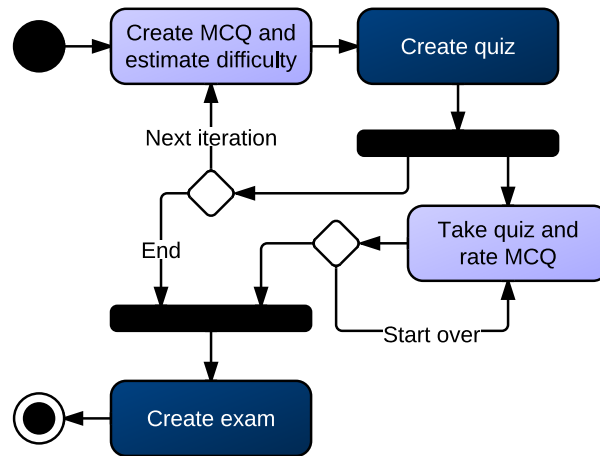


Fig. 1. Peer-supervised learning process

2 Learning Process

In this section, we specify a process that guides students through peer-supervised learning. The goal of the process is a complete peer assessment of students. That is, no expert ratings are required. In a nutshell, students create MCQ and estimate their difficulty. Questions are combined into quizzes during the term where students rate the MCQ created by their peers. The ratings and difficulty estimations help the teacher to select questions for a written examination at the end of term.

The process consists of one or more iterations; each iteration covers a different set of topics. A *topic* refers to a section in the lecture notes or textbook used for the course. The learning process is repeated until all topics of the course have been covered. Figure 1 shows the peer-supervised learning process as activity diagram. Activities in light colors are performed by students, activities in dark colors by the teacher. We now describe the individual activities:

2.1 Create MCQ

Each MCQ consists of a title and a textual description. Students are required to provide four possible answers, with exactly one being correct. For each answer we request an explanation why it is correct or not. We also demand that students specify which topic of the course their MCQ addresses. This allows us to verify that each student creates at least one question for each topic. Students are also asked to estimate the difficulty of their MCQ on a six-point scale. The estimation should correspond to the time it takes other students to answer the question. Finally, students can give their consent that the MCQ is used in the quiz or examination (opt-in). The motivation for students to create MCQ is that they are graded as part of our process. The chance that one's own question appears in

the examination is an additional incentive. Moreover, we expect that the creative task of creating MCQ increases the motivation further.

2.2 Create Quiz

After all MCQ have been turned in, the teacher combines questions into a quiz. The quiz remains open until the end of term, allowing students to prepare for the examination. The teacher can either use all MCQ created or only a sample. However, he may use only questions students have given their consent for.

2.3 Take Quiz

In the quiz, students may answer the MCQ of their peers. Questions are presented in random order. Students get immediate feedback on whether their answers are correct or not. The quiz is a very good preparation for the examination because it is based on the same collection of MCQ. In addition, we ask students to assess the MCQ of their peers. They can rate the quality and difficulty of each question on a six-point scale. Students can see the MCQ, but not the author name or the ratings by their peers.

2.4 Create Exam

At the end of term, the teacher creates a written examination out of the MCQ created during the process. Students are graded according to the ratings of their MCQ as well as their performance in the examination. To do so, the teacher aggregates the quality ratings for each MCQ into a score. These scores are then combined into a grade for each student. Because of legal obligations, the grades in our case study must rely on expert ratings.

3 Technology

The Moodle system runs on a LAMP⁵ environment. Table 3 serves as a reference for future operations.

Server	Debian GNU / Linux 6.0.1 with 12GB of RAM
Webserver	Apache 2.2.16
Database	MySQL 5.1.49 (InnoDB)
Scripting language	PHP 5.3.3
Moodle	Moodle 2.0.1 stable
IDE	NetBeans IDE 6.9.1 with wampServer 2.1

Table 1. Used hard- and software

⁵ Acronym for the combined use of Linux, Apache, MySQL and PHP

Shortly before the start of the implementation phase of this work Moodle version 2.0 was published [3]. With the most significant update in its version history, many changes have been made in most Moodle interfaces. Often they were so fundamental that no Deprecation⁶ process could be used.

Almost all modules from version 1.9 need to be adapted in order for them to be compatible to the new version. Probably because of this and the lack of a complete documentation, almost 75 percent of all registered Moodle installations are currently (November 2011) still in version 1.9.x or older [2].

Because of many improvements, particularly in the area of security, database layer, question database and user interface as well as in regard to the future viability of the modules created in this work it was decided to develop for Moodle version 2.0.1.

4 Question Creation Module

During the research phase of this work a module for Moodle 1.9 with the name *QCreate Module* [4] was discovered, which implements a Moodle Activity, where questions can be created by students and graded by teachers in a question overview. It was released in 2008, but no further developed.

Due to the similarity of the features that were needed for this work, the module was rewritten for Moodle version 2.0 and extended to meet our requirements. This includes improvements to the rights management and the creation of a convenient interface to evaluate questions, as explained below and can be seen in figures 12 and 13.

4.1 Database Structure

Figure 2 shows the database structure of the Question Creation Module. Less important fields, such as timestamps and internal flags are not displayed in the diagram. Items marked with (*) are existing tables in the Moodle question database and are displayed to show the relationship with the newly created tables. Moodle requires an own primary key with a unique ID for every table in the database, which makes it necessary to implement relationships with more than one key with a *UNIQUE*-constraint.

The following list explains the tasks of every table:

mdl_qcreate is the main class of every Question Creation Module and is used to store all its configuration settings. Besides name and description of the activity, properties such as the related course, the amount of questions that are required from the students and dates when the activity should be available are stored here.

⁶ By deprecation old interfaces remain functional after a change in code, but the programmer is warned of a possible deactivation in the future

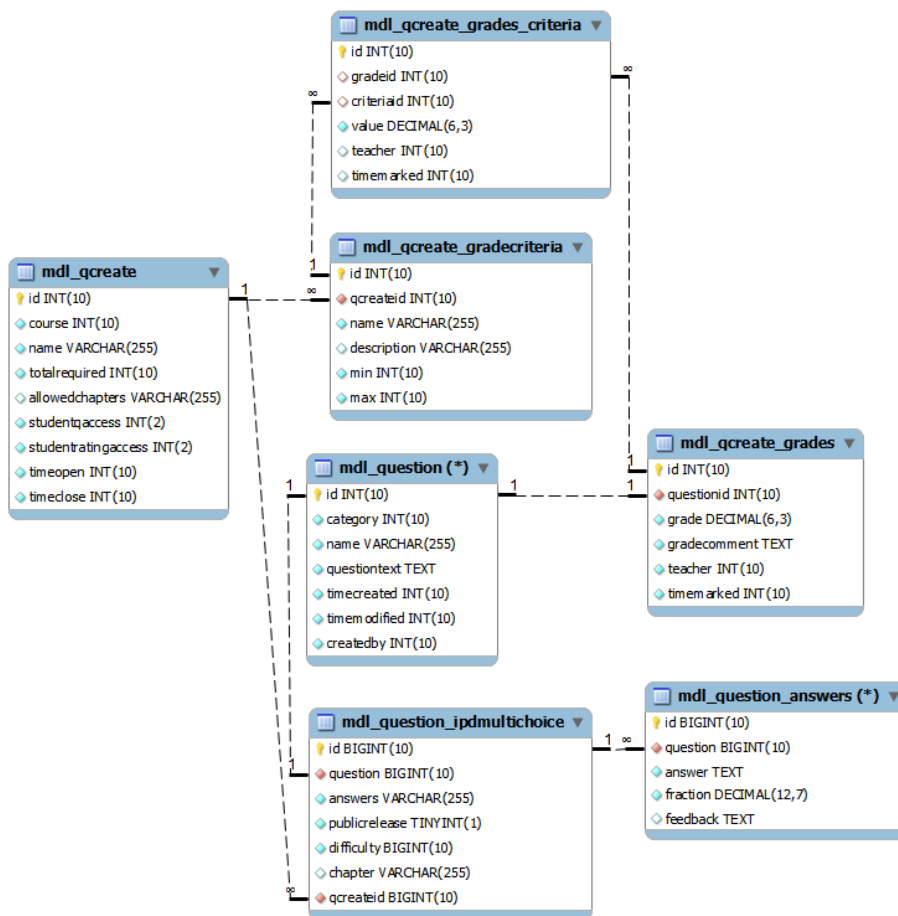


Fig. 2. EER Diagram of the Question Creation Module

mdl_question is a proprietary data type in Moodle and realizes the abstract superclass of all questions in the Moodle Questionbank. All common information about a question, such as title, question text, author and category is stored here.

mdl_question_ipdmultichoice is an extension of the superclass *mdl_question* and implements the specific question type *IPD Multiple Choice*, which was used as the question type for this work. The one-to-one relationship by the field *question* links both classes together.

mdl_question_answers, also a proprietary Moodle data type, links given answers to a question. An answer consists of the answer text and feedback, which can be used for a further explanation to the learner why the marked answer was right or wrong. Multiple Choice questions use this table to store

all possible answers. The field *fraction* is used to define the percentage of points a participant earns if he selects this answer in a quiz. This supports the use of MCQ with several possible right answers. For the MCQ type used in this work with only one possible correct answer, *fraction* = 0 means the answer is wrong and *fraction* = 1 means the answer is correct.

mdl_qcreate_grades has a one-to-one relationship with *mdl_question* and stores the overall grade of a question and the ID of the teacher who is responsible for the grade. The field *comment* allows a text comment from the teacher.

mdl_qcreate_grades_criteria links a question's overall grade with a set of criteria ratings. The type of a criterion is denoted by *criteriaid*, its value by *value*.

mdl_qcreate_gradecriteria is used to define all criteria referenced by *mdl_qcreate_grades_criteria*. The value range is defined by the fields *min* and *max*. The field *qcreateid* links the criteria to its related Question Creation Activity to support different sets of criteria in different activities.

4.2 User Interface

The user interface for the Question Creation Module has the task to provide a convenient way to create questions and give an overview over previously created questions to the students. Teachers need to be able to create and configure an Activity by themselves and to efficiently grade all submitted questions. In the following, the steps to use the activity will be described:

Creation and Configuration. Like any other activity in Moodle, the module can be created with the *Create Activity*-Menu (Figure 3) and placed in a desired position on the course overview page (Figure 4).

On the configuration page, all relevant settings of the activity can be set up. The name of the activity will be displayed on the course page. The entered description will be shown to the students as a short introduction to the assignment (Figure 5)

The period during which the activity is open to the students can be set up by using the time control settings shown in Figure 6.

To facilitate the grading of questions and to give the possibility to divide workload among the teachers, authors are asked to enter the chapter their respective question is related to. Figure 7 shows the definition of the chapters and the amount of questions the students are asked to create.

Question Overview for Students. Figure 8 shows the overview page for students. Questions that have already been created can be previewed, edited or deleted. Once the questions are graded by the teachers, the ratings and comments can be displayed to the students.

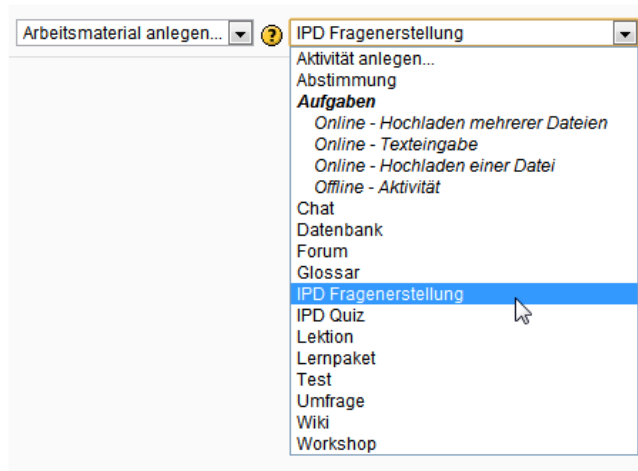


Fig. 3. Selection of the module *IPD Question Creation* in the *Create Activity* menu



Fig. 4. The created activity in the course overview

Question Creation. The implementation of the input and output mask for a question of a certain type is done by the question type itself, in our case in the module *ipdmultichoice*.

Creating a new question starts with entering a question title and the actual text of the question (Figure 9) with the help of a Rich Text Editor from the Moodle API. This makes the use of formatted text, images and HTML tables for the question text possible.

Figure 10 shows how one of the four possible answers can be entered. With a check mark at *Correct answer*, the current answer is marked as the correct one to the respective question. The text in the field *feedback* appears as a response to the participant, once this answer was submitted. This allows the display of an explanation next to the result *correct / incorrect* to why an answer is correct or not.

As shown in Figure 11, authors were asked to estimate the difficulty of a question and to specify which chapter of the lecture the question belongs to. If

Name* Fragenerstellung zu Übungsblatt 3

Einführung*

Schriftart | Schriftgröße | Vorlage

B | *I* | U | ABC | x_s | x_l

Englisch | **de** | **HTML** | **ABC**

Angenommen, Sie müssen für eine schriftliche Prüfung im Fach "Datenbanksysteme" Multiple-Choice Aufgaben erstellen. Wie sehen Ihre Aufgaben aus?

Die Aufgaben, die Sie einreichen, sollen sich auf mindestens drei unterschiedliche Kapitel der Vorlesung beziehen. Insgesamt sind fünf Fragen gefordert zu den Kapiteln 1-7 unter Berücksichtigung der im Übungsblatt 3 genannten Bedingungen.

Pfad:

HTML-Format

Fig. 5. Name and introduction of the activity to be created

Zeitsteuerung

Aktivität eröffnen 6 | September | 2011 | 15 | 45 | Aktivieren

Aktivität schließen 16 | September | 2011 | 20 | 00 | Aktivieren

Fig. 6. Configuration of timer settings

Bewertung

Erlaubte Kapitel ⓘ 1,2, relationale Algebra, SQL

Anzahl der zu erstellenden Fragen 5

Studentischer Zugriff

Zu eigenen Fragen Vorschau, Anlegen und Editieren

Bewertungen den Studenten in der Fragenübersicht anzeigen

Fig. 7. Settings for grading and user access

Startseite ► Meine Kurse ► Datenbanksysteme ► IPD Fragenerstellungen ► Fragenerstellung zu Übungsblatt 3

Angenommen, Sie müssen für eine schriftliche Prüfung im Fach "Datenbanksysteme" Multiple-Choice Aufgaben erstellen. Wie sehen Ihre Aufgaben aus?

Die Aufgaben, die Sie einreichen, sollen sich auf mindestens drei unterschiedliche Kapitel der Vorlesung beziehen. Insgesamt sind fünf Fragen gefordert zu den Kapiteln 1-7 unter Berücksichtigung der im Übungsblatt 3 genannten Bedingungen.

Aktivität ist geöffnet. Aktivität eröffnet seit Sonntag, 6. November 2011, 13:45. Aktivität schließt Mittwoch, 16. November 2011, 18:00.

3 von 5 Fragen erstellt.

Neue Frage erstellen

- Funktionale Abhängigkeiten
- ER-Modellierung
- Relationale Algebra

Sie sind angemeldet als [Alexandra Schmitz](#) (Logout)

Fig. 8. Overview of the activity for students

Titel der Frage*

Fragetext

Schriftart Schriftgröße Vorlage

B I U ABC x_n x^2

Gegeben sei folgende vollständige Historie:

$$\begin{array}{ccccccccc}
 r1[x] & \rightarrow & w1[x] & \rightarrow & r1[y] & \rightarrow & r1[x] & \rightarrow & w1[y] & \rightarrow & c1 \\
 & & \downarrow & & \nearrow & & & & & & \\
 r2[y] & \rightarrow & r2[x] & \rightarrow & w2[x] & \rightarrow & w2[z] & \rightarrow & c2 & &
 \end{array}$$

Wie viele der folgenden drei Histories (H1, H2, H3) sind zu obiger History (H0) Konflikt-äquivalent?

Pfad:

HTML-Format

Fig. 9. Field for question title and editor for the task

Antwort 2

Antwort

Richtige Antwort?

Begründung

Schriftart - Schriftgröße - Vorlage -

B *I* U ABC X₂ x₂ -

Diese Antwort ist richtig.

H1 ist zu H0 konfliktäquivalent, da H1 die gleichen Transaktionen und Operationen wie H0 enthält und auf den konfigurierenden Operationen dieselbe Ordnung definiert ist. Die hinzugefügte Ordnung $r2[y] \rightarrow w1[x]$ betrifft ein Operationenpaar, das nicht in Konflikt steht, also ergibt sich keine Änderung auf Pfad:

HTML-Format

Fig. 10. Field for an answer choice with feedback

the author checks the box under *Publication*, he agrees to the publication of his question in future exercises and the examination.

Eigenschaften

Schwierigkeit

Kapitel:

Veröffentlichung

Ich stimme einer Verwendung dieser Frage in zukünftigen Übungen (inklusive Präsenzübung am Semesterende) zu.

Fig. 11. Further information about the question

Grading Overview. The Grading Overview (Figure 12) serves as a starting point for all actions the teachers can carry out in the context of the Question Creation Activity.

All questions that were created in this activity are listed in a sortable table. To divide the workload between teachers, it is possible to filter questions by author name or by chapter.

Questions that are already graded are displayed green. The column *Criteria* shows all criteria ratings, *Grade* shows the overall grade for each question. Exclusion shows if a question has been marked as unusable or as a duplicate.

Vorname / Nachname	Frage	Kapitel	Status	Kommentar	Zuletzt geändert	bewertet am	Kriterien	Bewertung	Exklusion
Alexandra Schmitz	Funktionale Abhängigkeiten	5	bewertet		17. June 2011, 08:55	17. June 2011, 15:22	Originalität: 4 Korrektheit: 5 Anspruch: 3 Formales: 5	3.0	Duplikat? <input checked="" type="checkbox"/> Unbrauchbar? <input checked="" type="checkbox"/>
ER-Modellierung		4	bewertet	Begründungen zu Antwortmöglichkeiten fehlen. Die Frage selbst ist nur zum Auswendiglernen, kein neuer Blickwinkel	16. June 2011, 19:51	9. November 2011, 04:04	Originalität: 5 Korrektheit: 3 Anspruch: 4 Formales: 3	3.0	Duplikat? <input checked="" type="checkbox"/> Unbrauchbar? <input checked="" type="checkbox"/>

Fig. 12. Question and grade overview for teachers

Question Grading View. A view to grade each question (Figure 13) can be accessed by clicking on the question title in the Grading Overview. The questions' text and all possible answers, including feedback and information on which answer is marked as correct, are displayed here.

The section to grade the current question is at the bottom of the page. After grading each of the criteria on the left side, the field for the overall grade automatically gets filled out by a Javascript function which aggregates all ratings into a single value. For maximum transparency and to avoid mistakes it was decided to let the client-side handle the calculation. Otherwise, after a change of the aggregation function both client as well as the server would have to be updated. Also, if desired, a grade can be changed manually regardless of the value the aggregation suggests.

A comment to the assessment can be entered and then saved. With the *Save & next Question*-Button it is possible to jump to the next question directly. Using this makes it possible to define a custom list of questions through filtering in the Grading Overview and quickly iterate through them in the described detailed view.

5 Quiz Module

Moodle already has a Module named *Quiz* to publish questions from the database as a quiz package. Here, it is possible to individually present the questions in a random order for each of the students' quiz attempts⁷. For the evaluation of a quiz, every attempt for each question is recorded in the database. Therefore, it made sense to develop the quiz module further and to change certain properties

⁷ A quiz attempt is, regardless of browser session, always open as long as a teacher or a student did not close it

1
Punkte: --/1,00

Title: Standardkardinalität

erstellt von: Alexandra Schmitz

Kapitel: 4

Veröffentlichung: Ja

Es sei das folgende ER-Diagramm gegeben:

```

graph TD
    CK[Chefkoch] ---|1,1| K{ }
    K ---|1,4| R[Restaurant]
    K ---|7,14| G[Gericht]
    style K fill:none,stroke:none
    
```

Welche der folgenden Aussagen stimmt nicht mit dem gezeigten Modell überein?

Antwort wählen:

<input type="radio"/> a. In einem Restaurant wird das selbe Gericht nicht von verschiedenen Köchen gekocht. x	Die Aussage ist richtig, da eine Koch-Gerichtkombination nur genau einmal vorkommen kann (das sagt das [1,1] bei "Chefkoch")
<input type="radio"/> b. Ein Chefkoch kocht in einem Restaurant höchstens 14 verschiedene Gerichte x	Die Aussage ist richtig, da ein Koch in einem bestimmten Restaurant zwischen sieben und vierzehn verschiedene Gerichte kocht (das sagt das [7,14] bei "Gericht")
<input type="radio"/> c. Ein Koch kocht das selbe Gericht in ein bis vier Restaurants x	Die Aussage ist richtig, da eine Koch-Gericht Kombination nach dem Diagramm genau ein bis viermal vorkommen kann.
<input checked="" type="radio"/> d. In jedem Restaurant kocht höchstens ein Koch ✓	Die Aussage ist falsch, da in einem Restaurant verschiedene Gerichte auch von verschiedenen Köchen gekocht werden können.

Originalität	5				
Sachgerechtigkeit	5	Bewertung 4.0	<input type="checkbox"/> Duplikat	<input type="checkbox"/> Unbrauchbar	Kommentar In der Vorlesung wurde Standardkardinalität als eine Zahl notiert und Teilnehmerkardinalität als Intervall. Diese Aufgabe ist bei den Kardinalitäten inkonsistent.
Korrektheit	5				
Angemessenheit	4	Speichern			
Anspruch	5	Frage 1 von 50			
Verständlichkeit	4	Speichern & nächste Frage			
					nächste Frage

Fig. 13. View for teachers to grade a question

to meet our requirements. Support for extensions within the module is not possible. Because of this, a copy of the original module was created and developed further under a new name. This keeps the original module untouched for other use. Only as few changes as possible were made in the given source code. Instead, the new sources for the user interface are stored in separate files. It is therefore possible to easily translate changes, such as officially released patches, from the original quiz module into the new module.

5.1 Database Structure

The EER diagram in Figure 14 shows the created tables and their relationships for the quiz module.

The following explains the tasks of each displayed table:

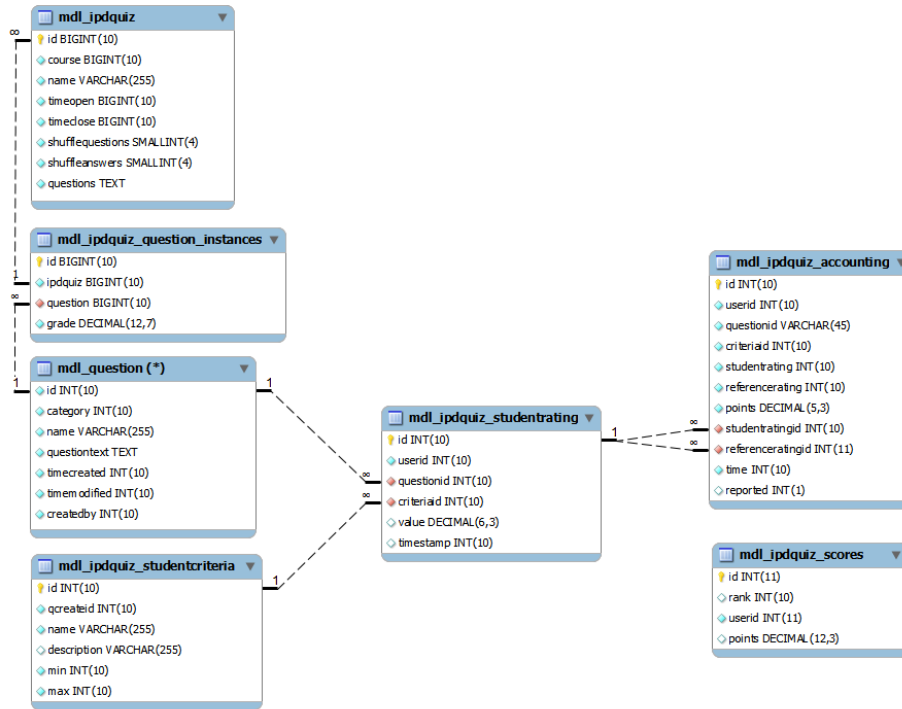


Fig. 14. EER diagram of the quiz module

mdl_ipdquiz contains, similar to *mdl_qcreate* (Section 4.1), information about all quiz activities that have been created.

mdl_question, as described in Section 4.1, the superclass of all Moodle questions.

mdl_ipdquiz_question_instances is a linking table to realize the many-to-many relationship between a number of quizzes and a number of questions. Additionally, the field *grade* specifies how many points are given when a question was answered correctly.

mdl_ipdquiz_studentrating has the constraint *UNIQUE* (*userid*, *questionid*, *criteriaid*). Each entry contains one student rating of a certain type of a specific question.

mdl_ipdquiz_studentcriteria defines the student's rating criteria similar to *mdl_qcreate_gradecriteria* in Section 4.1.

mdl_ipdquiz_accounting is used for bookkeeping of bonuses that were given to a student for rating a question. For evaluation purposes, every entry contains the amount of bonus points that were awarded and additional information about the rating, such as the ID of the related question. The field *reported* tells the system whether the entry has already been reported back to the user or not.

`mdl_ipdquiz_score` serves as a caching table for the high score list. Every five minutes a users total points based on `mdl_ipdquiz_accounting` are summed up and written into this table.

5.2 User Interface

89
Punkte:
0,00/1,00

Richtiges Insert Statement

erstellt von: Michael Peterson

Eine Relation wurde mit folgendem Statement erstellt:

```
Create table Bücher (ISBN char(10),
Titel varchar(250),
Erscheinungsjahr int,
Primary key(ISBN),
check (Erscheinungsjahr between 1800 and 2050))
```

Welches Insert Statement muss gewählt werden, damit der Datensatz nach dem vorliegenden Create Statement korrekt eingefügt werden kann, es also zu keiner Fehlermeldung kommt.

Antwort wählen:

<input type="radio"/>	a. INSERT INTO Bücher VALUES ('1258390376', '', 2001)	
<input checked="" type="radio"/>	b. INSERT INTO Bücher VALUES (1234098765, 'Moderne Computer', 1990) ❌	Bei der ISBN handelt es sich um den Datentyp char, diese Angabe würde aber als Zahl interpretiert, da die Hochkomma fehlen.
<input type="radio"/>	c. INSERT INTO Bücher VALUES (NULL, '3D Grafik', 2009)	
<input type="radio"/>	d. INSERT INTO Bücher VALUES ('9087547582', 'DirectX und C++', 2057)	

Falsch

Schwierigkeit ★ - ▾

Duplikat ★

Nein

Ja

Qualität ★ - ▾

Diese Frage ist brauchbar ★

Ja

Nein

Diese Frage soll in der Prüfung gestellt werden.

Ja

Nein

Fig. 15. A quiz question with rating interface

Figure 15 shows a question attempt after a user submitted a solution. The answer choices are presented in random order. After selecting an answer and

clicking on *Submit*, the feedback entered by the question author is displayed to justify why the given answer was right or wrong.

The interface for the voluntary rating of a question is at the bottom of the page. Apart from *difficulty* and *quality* with a points scale, both binary criteria *duplicate* and *unusable* are selectable. The star symbols indicate that it is possible to earn bonus points by rating one or more of these criteria. As long as a criterion of a specific question has not yet been rated, its control element is in an undefined state. If a previous rating was found, the control element is disabled and cannot be changed any longer.

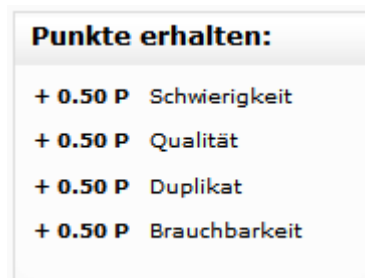


Fig. 16. Display block to report earned bonus points



Fig. 17. Quiz Navigation

A display block (Figure 16) is displayed on the left side of the page to report the bonus points earned from the last question back to the user. Below, a block called *Quiz Navigation* shows every question in a quiz represented by a number. The order of questions for each user is randomly determined and remains the same throughout the quiz attempt. Therefore it is possible for students to note down questions they had problems with during the quiz.

6 Pseudonym Generation

The pseudonyms are implemented to facilitate recognition of users over time. To protect the anonymity of students, pseudonyms should not resemble the real

names of the participating students. For this reason, two word lists consisting of common first and last names in the US were combined into a full name, as described in the following.

Because the alias of a person has to be different for different observers, a function p which returns different pseudonyms for different arguments $p(observer, observee)$ is needed.

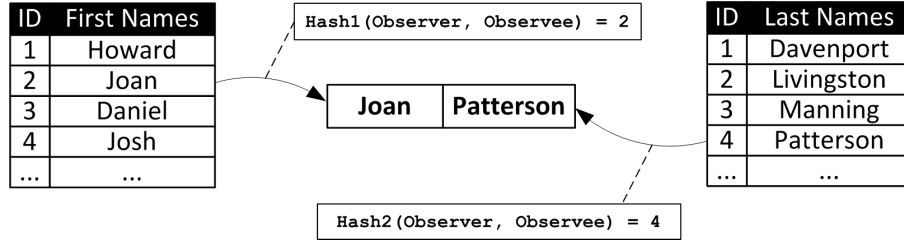


Fig. 18. Illustration of the pseudonym generation

The selection of individual names is done by a hash function. Good hash functions try to avoid collisions of the output value for different input values.

The lists of names are stored in the database and directly combined in all queries where aliases are needed. Figure 18 illustrates the selection of a pseudonym.

The function for a word list L with two unique Moodle User-IDs is as follows:

$$\text{hash}_L(ID_{\text{observer}}, ID_{\text{observee}}) = \text{DEC}(\text{SHA}(ID_{\text{observer}} \cdot \text{SALT}_L \cdot ID_{\text{observee}})) \bmod \text{SIZE}(L) \quad (1)$$

Operator \cdot stands for string concatenation. SHA stands for a function that returns a SHA^8 -hash for a given string which gets converted into the decimal system by function DEZ . This value is limited to the size of a word list by the modulo function. SALT_L is a string used as a hash salt to raise the entropy of the input value and is different for both lists.

7 Conclusion

This report described the design and implementation of several modules for Moodle. These modules facilitate a learning process where students create and answer multiple choice questions. For peer assessment, students rate the questions of other students. To evaluate the quality of peer assessment, teachers grade

⁸ Secure Hash Algorithm

the questions as well. The *Question Creation Module* allows students to create questions and teachers to grade them. The *Quiz Module* is an advancement of Moodle's Quiz module with an additional user interface to rate questions based on different criteria. Both modules can be used in combination or separate from each other.

References

1. Moodle.org (2011), <http://moodle.org>
2. Moodle.org: Moodle-statistik (2011), <http://moodle.org/stats>
3. Moodle.org: Release notes (2011), http://docs.moodle.org/dev/Moodle_2.0_release_notes
4. Pratt, J.: Question creation module (2008), <http://moodle.org/mod/data/view.php?d=13&rid=1120>