

UML-QUIZ: AUTOMATIC CONVERSION OF WEB-BASED E-LEARNING CONTENT IN MOBILE APPLICATIONS

Hilda Tellioglu *Vienna University of Technology. Favoritenstrasse 9-11/187, A-1040 Vienna, Austria.*

Alexander von Franqué. *Vienna University of Technology. Favoritenstrasse 9-11/187, A-1040 Vienna, Austria.*

Christian Kletzander. *Vienna University of Technology. Favoritenstrasse 9-11/187, A-1040 Vienna, Austria.*

Alexander Kögler. *Vienna University of Technology. Favoritenstrasse 9-11/187, A-1040 Vienna, Austria.*

ABSTRACT

Many educational institutions use Learning Management Systems to provide e-learning content to their students. This often includes quizzes that can help students to prepare for exams. However, the content is usually web-optimized and not very usable on mobile devices. In this work a native mobile application (*UML-Quiz*) that imports quiz content from Moodle and adapts it for a mobile environment has been developed. The result is highly usable and promising. The existing teaching content can be easily used in the new system by offering advantages of mobile applications. In this paper, the design decisions for *UML-Quiz* and the usability optimizations to the content and to the interaction mechanisms are presented. Within an evolutionary design process we could improve the first version of the quiz and learn about several interaction and user interface design issues. In a second version of the quiz, we managed to integrate users' feedback to the system and improve its acceptance by the users. We present our evaluation process and the results before concluding the paper.

KEYWORDS

E-learning, m-learning, mobile, usability, automatic content adaption, automatic content conversion.

1. INTRODUCTION

Many educational institutions use web-based Learning Management Systems (LMS) to provide e-learning content to their students. Popular examples include Moodle, Blackboard, Desire2Learn, and ILIAS. Besides being a well-known and mostly used system Moodle “is not intrinsically designed to be accessed through mobile devices, but the constructivism pedagogy supported in Moodle VLE¹ could be applied to design the interaction in m-learning” (Hewagamage et al., 2012, p.2). Moodle does not fully support JavaScript. This makes it difficult to access via mobile devices.

Most of the LMS platforms include quiz modules. These quizzes can, for example, help students to prepare for exams. However, these platforms rarely offer easy access for mobile devices, especially not with native applications. Native applications offer numerous (usability) advantages over web-based applications, especially if they are running on mobile devices. However, the creation of native apps is time-consuming and expensive.

In this paper we propose a native mobile quiz application (*UML-Quiz*) in different versions (one for iOS and one for Android, as a result of an evolutionary approach) that can deal with web-optimized e-learning content in a way that preserves usability and encourages easier learning. It is accompanied with an automatic system that converts the web-optimized data into a mobile-friendly format, both in size and layout. A few optimizations and enhancements to the data are being used to make the application compelling and easy to use, even though the data was not originally prepared for this specific device.

At our university we use TUWEL (TU Wien E-Learning, tuwel.tuwien.ac.at) as an e-learning environment based on Moodle. The most courses we offer at our faculty are organized in TUWEL. Time schedules, documents we provide for our students, work assignments, marking, etc. are the main components we use for the administration and management of our courses. We also use the community functionality and communication channels that Moodle facilitates. TUWEL does not replace face-to-face learning modes; it only offers additional support for the communication between the lecturers and students by being mainly a document management system and organizer at the same time.

To improve the learning success of our students, especially in the computer science bachelor studies, we want to explore new ways of supporting learning. Some researchers used quiz as an effective way to engage students in the classroom. Even using a simple true/false quiz to motivate students to react immediately and to share their viewpoints and opinion with others can support the participation of students in a course and the success in learning (Edgar, 2007, p.197). Considering quiz as a possibility to “engage students in fundamental play and enjoy a challenge while teaching them” (Febrina and Adnan, 2013, p.428), we designed and developed a quiz, called *UML-Quiz^{iOS}*, to facilitate learning unified modeling language (UML) in a specific course, called object oriented modeling techniques (von Franqué, 2013). This quiz is based on Apple’s iOS. The evaluation of this application resulted in several factors that needed to be improved. We continued in developing *UML-Quiz^{iOS}* by mainly focusing on the usability issues of its interaction mechanisms and improved it in several ways. We re-designed, re-developed, and re-evaluated it and the result is *UML-Quiz^{Android}* (Kletzander and Kögler, 2014). We studied usability issues of *UML-Quiz^{Android}* interaction mechanisms. This paper is about the creation and evaluation of *UML-Quiz^{iOS}*, and furthermore

¹ Virtual Learning Environment

the improvement of it with the new system *UML-Quiz*^{Android} as well as the analysis and the presentation of our final results.

In the next section we briefly introduce relevant related work. We also will touch upon other work that adapts web-content for mobile devices. In Section 3 we present and evaluate the first version of the *UML-Quiz* (iOS version) in this scope. Build upon the evaluation results we show the second version of the *UML-Quiz* (Android version) and present the evaluation results. The focus of the Android version is the design and usability of the quiz on mobile devices. We conclude after showing the results of the second evaluation and lessons learned.

2. RELATED WORK

In this section, we want to show some other systems developed so far. Second, we want to focus on usability and content conversion in such systems. We will also highlight the differences to our system *UML-Quiz*.

Despite the fact that “accessing learning content through mobile devices is not very comfortable” (Hewagamage et al., 2012, p.32) and e-learning content can be accessed the best through desktop interfaces, the advantages of mobile applications especially for learning cannot be underestimated. There are several systems that are implemented as mobile applications, mainly quizzes for teaching.

The group of Schön et al. (2012a, 2012b) developed a mobile quiz application for the University of Mannheim. Its main purpose is capturing classroom feedback from the students during courses, which is different from the *UML-Quiz* that mainly assists students in the preparation process for exams. It is a web-based application and supports dealing with multiple- and single-choice questions. The application’s data is formatted to fit the mobile screen. Question texts are kept short, possible answers are usually single words, and images are kept at a size which are clearly visible on a mobile screen. Another system was developed to support learning foreign languages (Tabata et al., 2010). It uses the same approach as the one from the University of Mannheim. There are text-based questions and answers whereas texts are kept very short. The system supports only multiple-choice questions.

iClass is a further system for mobile learning (Eyhab and Qusay, 2012). In contrary to the systems above, it hasn’t been developed on top of an existing university infrastructure, but instead has been designed for mobile devices from scratch. However, it is a web-based implementation using the phone’s browser, and this is the main difference to *UML-Quiz*. The decision for this technology was made for the reason to support a variety of devices. Authors did not want to “enforce a particular device” (p.591). At the same time, they are aware of the reduction of usability because of not taking care of the platform and the follow up limitations in the interaction with the system.

The multiple-choice questions realized in (Montanaro, 2012) resemble closely to the structure of *UML-Quiz*. The content presented is dynamic, the text can be displayed in multiple rows, and feedbacks are being shown after having solved the question.

UML-Quiz closely follows the general structure of most of these systems, and in some aspects improves upon them. One of the main differences is that *UML-Quiz* is implemented natively, whereas most of these systems are web-based. We found the following systems as native mobile applications that we unfortunately could not test because of the license

restrictions: Upside2Go with a quiz module, Mooestro without a quiz module by focusing on orientation at campus and while registering for lectures, Pearson Learning Studio and OpenClass as a very popular LMS, MOBL21 including multimedia to its quiz module, which is the closest to our *UML-Quiz* among all. Furthermore we found Adobe Captivate and Articulate as HTML5-based systems, again with no open access. Other systems available like Quiz and Flashcard Maker, Edmodo, Blackboard Mobile, or Socrative are not described here because of the lack of usability or features they offer.

The second important characteristic of *UML-Quiz* is the focus on usability and easiness. Even though the most of the mentioned systems have been developed with the aim to support mainly mobile platforms (Wasserman et al., 2010), they are not exhaustively using the possibilities of a mobile device. This is only possible if the system is developed natively for specific devices, what we do with *UML-Quiz*.

KnowledgePulse (Bruck et al., 2012), Quizzer (Giemza et al., 2012) and MLEA (Castillo et al., 2012) are also native applications for iOS and Android. They also constrain the content of the data to fit certain screen size of mobile devices. This scalability feature is a limitation of native applications (Giemza et al., 2012) and a challenge to expand the system to another platform.

Contrary to scalability, usability is an important advantage of native applications. Native applications usually perform much faster than web-apps – especially scrolling is much more responsive (Charland and Leroux, 2011). Native apps enable access to device-specific APIs, like Apple’s Core Data, which has been used in the development of *UML-Quiz*^{iOS}. They allow for offline use, which is particularly useful in the context of mobile learning. However, web-apps have the advantage of being platform independent and the development costs are usually much lower than the cost of building native apps (Selvarajah et al., 2013).

There are other systems adapting web-content for mobile devices (Casany et al., 2012), some specifically for e-learning material (Gutierrez-Martinez et al., 2012). However, these works concentrate mainly on the format conversion and the interoperability between different learning management systems, but they do not take the usability implications into consideration.

Romero et al. claim that when creating m-learning systems “designers of both software and content have to consider the special restrictions associated to their use” (2010, p. 375). The limiting factors are among others screen size, resolution, and special input mechanisms. Guidelines for designing usable educational material suggest that the display area should not exceed the screen size and the content should be segmented into smaller chunks. Additionally extensive scrolling and the number of clicks should be well designed and one should pay attention on the amount of relevant information on each page. Further points of importance are highlighted like clear navigation structures, consistency of interface elements, and typeface.

The user interface of an application and its interaction design are very much defined by the platform the system is implemented for. There are several libraries for GUI (graphical user interface) elements, usability guidelines to follow, conventions established for interaction mechanisms, and best practice examples for different platforms, what we of course considered in our study. Comparing iOS and Android platforms informs us for our design. We found out that the following two factors play an important role in this matter:

- One major difference between the two platforms is the Back-Button. In Android the Back-Button is used to navigate backwards in an application. If one application interacts with another application it is possible to switch back to the main application with this additional button. This is very useful if you quickly want to change something on the

screen or read something in another application and go back to the controller of this event. Many users miss the Back-Button in iOS applications. A Back-Button has an essential place in the architecture of such applications. That is why developers have to consider including a Back-Button from the beginning of the design process. To integrate a Back-Button into an application from scratch it is necessary to create an activity diagram because it is crucial to configure and administer the access of each activity onto the App-Back Stack. This means that if one presses the Back-Button to switch to the application's main menu, the App-Back Stack has to know which activity was the previous one in order to load it again.

- Another difference is the so-called Action Bar. The Action Bar is a small menu on the top of a screen of a mobile application. It can be used to place an individual help for the displayed activity or a menu to switch back to a specific previous activity.

Furthermore we selected four other mobile e-learning applications for the Android platform from the Google Play Store, mainly by focusing on the interaction design used to implement the quiz (see Figure 1). From layout design and content points of view these four applications are the most relevant ones for the Android version of our quiz. Our goal for the comparison was to find out the best interaction mechanism for an e-learning quiz application: 1) to display mixed content on a small device like mobile phones, and 2) to motivate end users to use their mobile devices to access the quiz because they find it highly usable and flexible. The latter should build the base for the improvement of our users' learning process (Figure 1): Ankidroid (<http://ankisrs.net/>) is an application for learning flashcards. Its design is simple and flat. It uses Action Bar on the top. The layout design provides enough space to show the question and answers of a quiz item. Brainyoo (<http://www.brainyoo.de/karteikarten-app/>) has been found as one of the best applications for displaying a quiz with a question and an image as well as displaying answers with text and images. On the top there is an Action Bar. The question can be shown with an image. It is possible to click on the image and switch it to full screen and back. The possible answers are displayed below the image. CoboCards (<http://www.cobocards.com/de/>) is very similar to Brainyoo. There are only two differences. On the top of the content area there is a progress bar, which shows the current card number and the number of cards left. In the content area there is a line between the question area and the answer area. Repetico (<http://www.repetico.de/>) shows perfectly the combination of grey scales inside the content area to divide the question and the answer area.

UML-QUIZ: AUTOMATIC CONVERSION OF WEB-BASED E-LEARNING CONTENT IN MOBILE APPLICATIONS

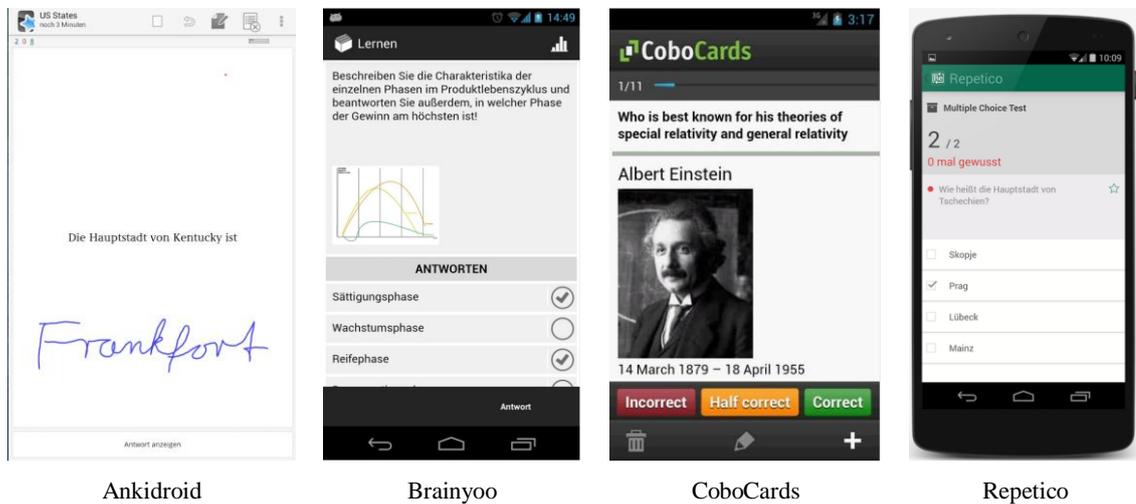


Figure 1. Four e-learning quiz applications selected to study the user interface and interaction mechanisms.

The two major differences between iOS and Android applications listed above and the ways how the compared four applications (Figure 1) implemented a quiz with respect to layout and display options inspired us to improve the interaction design of the existing UML-Quiz^{iOS}. By using the functionalities of Back-Button and Action Bar and by considering the display alternatives shown in the studied systems we built a new application with better interaction mechanisms and with more space for displaying the important things of a quiz, namely the questions and answers.

In the following sections we will present first the UML-Quiz^{iOS} and then UML-Quiz^{Android} with their evaluation.

3. UML-QUIZ^{iOS}: THE APPLICATION

UML-Quiz^{iOS} was designed to automatically import the web-based quizzes from Moodle and display them in a highly usable manner on a mobile screen (von Franqué, 2013). However, as Moodle supports free HTML formatting of the content (text of any length and any number of images), constraints and design goals had to be defined. We did not want to restrict the length of the question or answer text, as the system should be able to deal with existing content. However, we constrained the use of images to one image per question and one image per possible answer. UML-Quiz^{iOS} automatically discards questions that do not follow these constraints. We also wanted to ensure that navigation between questions is fast and easy. Romero et al. (2010) prefer paging over scrolling when navigating, so in UML-Quiz^{iOS} each question consists of a screen-full content. We implemented a simple swipe gesture for navigation between questions. The overall design is focused on providing as much space for the content as possible and minimizing the space that UI-elements take up. We wanted to

make sure that most elements of a question (question text, images, and possible answers) are visible at the same time.

Our system uses Moodle's export feature to obtain quiz data in the Moodle XML format. We then parse the data and import it into our native mobile application, UML-Quiz^{iOS}. Upon import, the content is optimized for use on mobile devices. See Figure 2 for an architectural overview.

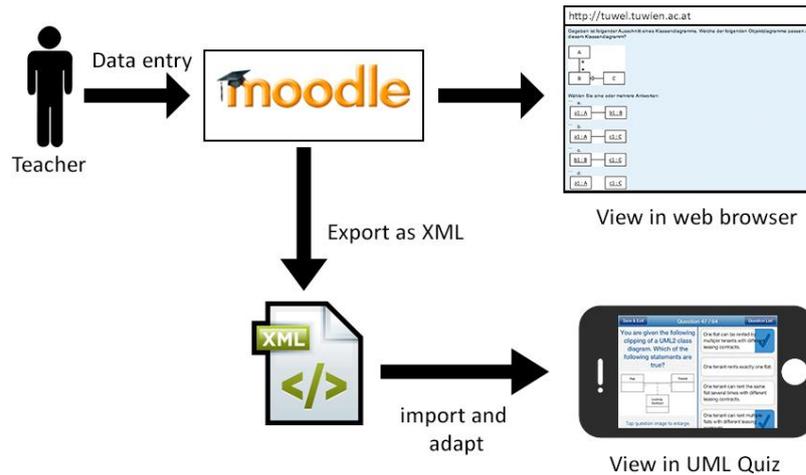


Figure 2. Architectural overview of UML-Quiz^{iOS}.

In TUWEL different question types are presented to the user. The text in all types can be formatted freely with standard HTML tags such as bold and italics. Any number of images can also be added at any point. Each question type has been individually adapted for use in the native app. The question and possible answers are presented as HTML-formatted text to the user. Usually, questions contain at most one image in the question text and one image per answer. However, in TUWEL, there are no constraints to enforce this rule, and theoretically more images are possible. See Figure 3 for an example, which illustrates how this is solved in TUWEL and implemented in UML-Quiz^{iOS}.

The Figure 3 also shows the same question after it has been imported into the mobile application. The interface is forced into landscape mode, splitting the screen content into half. On the left side, the question text and, if available, a single question image are displayed. Tapping the image enlarges it to full screen, also enabling zooming and panning with two-finger pinch gestures. On the right side, the possible answers are displayed, some of which are selected in Figure 3, which is visualized with blue checkmarks. Both sides of the screen are individually scrollable vertically, if the text is longer than screen size allows for.

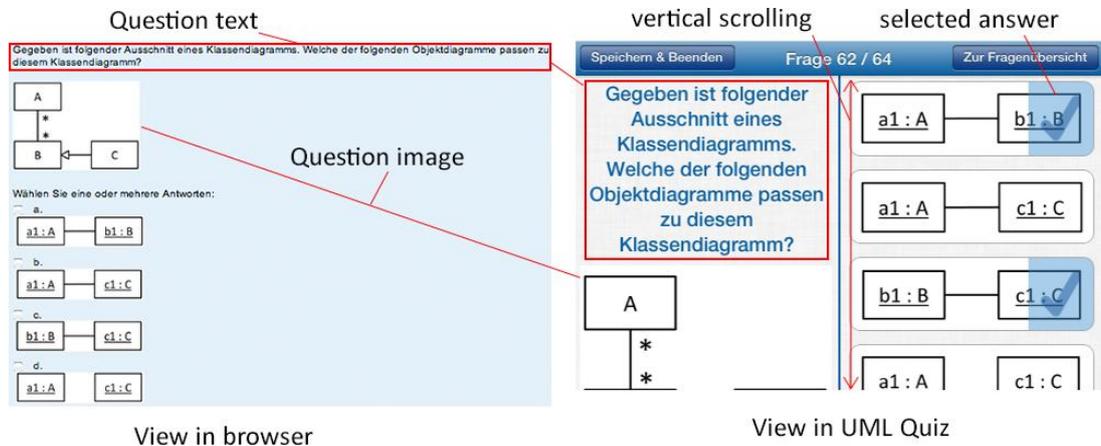


Figure 3. Multiple-choice questions in Moodle and UML-Quiz^{iOS}.

In TUWEL, some questions are implemented with a dropdown list. In these questions, a general text that explains the task is presented to the user (e.g., “Select the right answer from the list”). Then, a number of sub questions is being shown. For each of these sub questions, a drop-down list is displayed. The possible choices of this list are the same for all sub questions. See Figure 4 for an example.

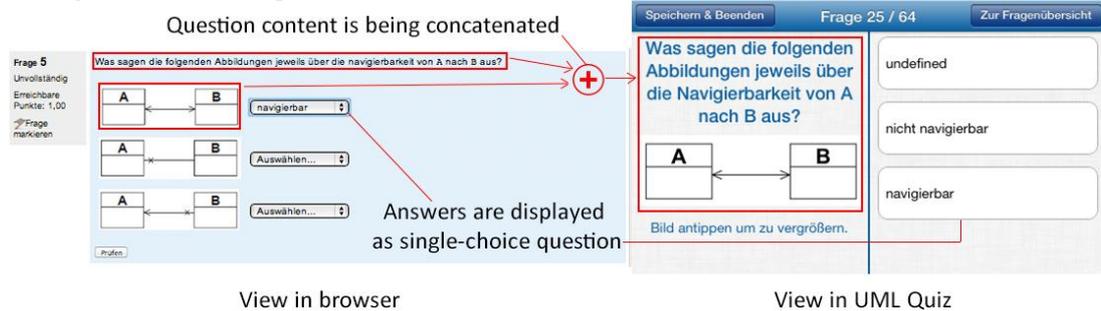


Figure 4. Optimization of drop-down questions in UML-Quiz^{iOS}.

On touch-screen devices drop-down lists are usually implemented with picker control elements. Early iterations of UML-Quiz^{iOS} have implemented this control element to support drop-down questions, however, the picker takes up so much screen space, that all the other content on the screen is occluded. To solve this problem, the following optimization has been used. Upon importing the TUWEL data, questions of the drop-down type are split into numerous single choice questions. Figure 4 shows a drop-down question with images in TUWEL and compares it with the first part of the same question split into a single choice question in UML-Quiz^{iOS}. We interpret the possible selections from the drop-down list as possible answers. The question text is constructed by concatenating the general instruction text of the drop-down question with the text or image from the specific question.

While this approach works much better than the use of the picker control element, it doesn't quite fit some of the existing data. In the TUWEL screenshot above, the general text is written in plural, as it refers to multiple images, however, in UML-Quiz^{iOS}, only one image is visible. In this case the consequences of this issue are harmless, as it merely might confuse

users. However, in other cases, this problem might completely transform the original sense of the question. As this limiting problem is caused by semantics, it is not possible to solve it in an automatic fashion; it requires a manual re-work of the questions.

Another question type is presented with a text box in TUWEL. These questions do not offer users a list of possible answers to choose from. Instead, they are prompted to enter the answer in a text box. The question itself is presented to the user with text, an image, or both, and usually requires them to calculate a number or a certain answer. See Figure 5 for an example and a comparison with the mobile application. As visible in the screenshot, a simple text field has replaced the left side of the screen. Tapping it reveals the on-screen keyboard, as well as a button to dismiss it again.

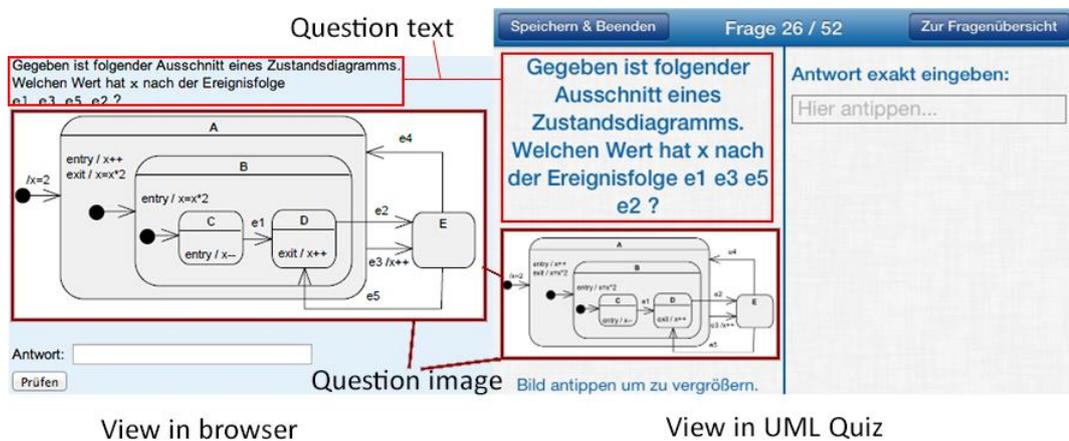


Figure 5. Comparison of text-box questions in UML-Quiz^{iOS}.

After a question has been solved, its correct answers are revealed (see Figure 6). Another feature of the system is the explanation text that is displayed to users after they finish a quiz. Its purpose is to provide additional information about the question or the possible answers. The text may also explain why certain answers are correct or incorrect. It is also possible to include images in the explanation.

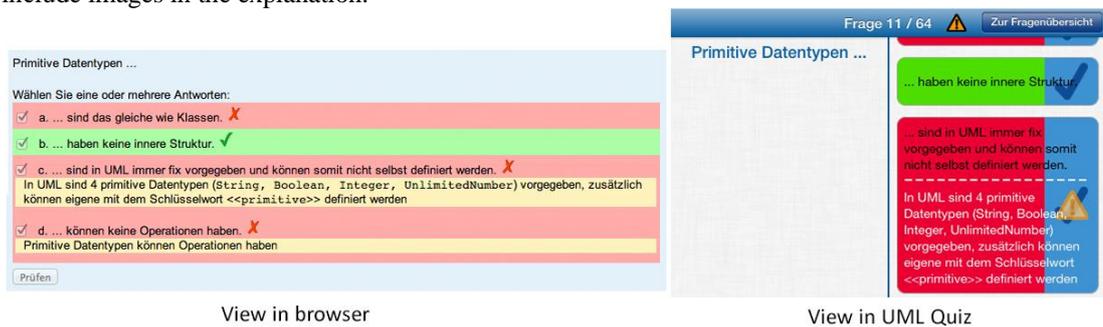


Figure 6. Comparison of solved questions in UML-Quiz^{iOS}.

UML-Quiz^{iOS} offers some additional features that help improve the usability of the system:

- Flashcard system – The application remembers how often each question has been solved and thus allows the user to practice depending on progress.
- Persistent quiz data and progress storage – In the final application, quiz data is loaded on first launch from XML files and then imported in a more convenient format. For access, Apple’s persistence framework, Core Data, was used, which uses an SQLite database as underlying storage format. All data from the XML files is stored, and is associated with some new data to track the user’s progress. The application remembers how often each question has been answered, which enables the flashcard system to divide up the questions into boxes. Also, the progress of saved, unfinished quizzes is stored. In the application settings, users also have the option to completely reset their progress.
- Multiple language support – The application loads different XML files depending on the device language. Quiz progress is stored independently for each language: whenever a new language is loaded, a new SQLite file will be created. When users switch languages, they are prompted if they want to keep the old language database or switch to the new one.
- Online question updates – It is possible to download new XML files from within the application. It will then parse them and update, add or remove any new content. It is also possible to add new images this way. They are first compared by timestamp, so that files that haven’t been changed won’t be re-downloaded this way. This feature enables teachers to change questions after the application has been shipped.

4. EVALUATION OF UML-QUIZ^{iOS}

UML-Quiz^{iOS} was tested by 6 persons from different age groups and with various educational backgrounds. Some were interviewed multiple times during the development process, and some were only shown the final application.

Multiple iterations of UML-Quiz^{iOS} were shown to Ulrike (55 years old, college dropout), including the final application. First prototypes of the application were in portrait orientation and had the entire content of a question (question text, answers, images) scroll vertically. Ulrike did not like this interface at all: solving a question was a very tedious process for her, as she had to scroll constantly between the question text and possible answer choices. To fit more content on the screen, a small text size was chosen, which she also criticized, since it was hard for her to read. In the final application, she said that most of the issues had been improved: the landscape layout made it easier for her to solve the questions, as most of the content was visible at the same time, requiring less scrolling. Font sizes were larger and the navigation between questions was much clearer.

The full application was shown to Micheline (28 years old, non computer science graduate student) after most parts were completed, not having seen the quiz in TUWEL before. She was initially confused with the navigation, not understanding the horizontal swipe gesture to navigate between questions. However, when she had grasped the feature, she stated that she liked it. She was also initially confused by the colors that were used to mark solved questions as correct or incorrect, thinking that the blue check signified correctness, not selection.

Lukas (23 years old, computer science undergrad student) was shown the full application after most parts were completed. He had seen and used the quiz in TUWEL before, as he participated in the UML course himself. He positively noted the navigation between individual questions and he really liked that one could see all the elements of a question in a single screen. The question overview button at the top left corner of the screen initially confused him, because he wasn't sure where it would lead him. However, when he pressed it and then navigated back to the original question, he understood what had happened. He also stated that he would have used the application, had it been available at the time he took the course.

Moritz (27 years old, computer science undergrad student) was also a participant in the UML course and had also seen the questions beforehand in TUWEL. He was also shown multiple iterations of the application, including the final result. He did not like the navigation between questions and was especially confused by the button leading to the question overview. The main reason for his dislike was the problem caused by both horizontal and vertical scrolling. He positively noted the flashcard system and the online updates.

Wolfgang (57 years old, non computer science graduate, working as a high school math teacher) is without a strong technological background. He liked the navigational layout and said that it was easy to understand and use. He especially noted the overview screen, allowing seeing which questions had been answered correctly after finishing the quiz. However, he thought that the font sizes were sometimes too small and hard for him to read. He liked that the question images were zoomable and disliked that the answer images are not zoomable, again making it harder for him to read. From his viewpoint as a teacher, he liked that the content could be revised and adapted through online updates, making it easy to add or change questions.

Marion (instructor of the UML teaching course at Vienna University of Technology) accompanied the entire development process, providing valuable feedback from the perspective of an instructor. She also did not like the initial prototype very much, as the portrait layout made it hard to solve a question without scrolling. She appreciated the switch to landscape mode and suggested further improvements to the layout, which included increasing the font size and removing some UI elements. The initial prototype had an additional UI element to jump from question to question, which was removed in favor of more space for actual content. Marion also liked the idea of splitting drop-down questions into multiple individual single-choice questions, however she was concerned that it might cause confusion for people entering the data in Moodle, who are not familiar with the mobile application.

These tests have shown that UML-Quiz^{iOS} improves the usability of the quiz enormously, when compared with the access on a mobile browser on Moodle. The big advantage is that most parts of a question are visible in a single screen and scrolling can be avoided in most cases. Users also reacted very positively to the additional features described in the previous section, especially to the flashcard system. The system generally does a good job in converting the web-based content and making it usable on the mobile screen.

However, UML-Quiz^{iOS} has some usability issues, which need to be solved. Most of them are contextual problems caused by the content used in the quiz. Such content has usually been designed with and for desktop screens. An example is the way that UML-Quiz^{iOS} deals with images. Because most images in the existing data are rather small, UML-Quiz^{iOS} scales them to the largest size possible with the given screen space, while maintaining aspect ratio. While this works well with most images, it sometimes produces unwanted results, which leads to images like in Figure 7a. In the screenshot, a giant element from a UML diagram is visible, even though the element's actual size is only a few pixels in diameter.

UML-QUIZ: AUTOMATIC CONVERSION OF WEB-BASED E-LEARNING CONTENT IN MOBILE APPLICATIONS



Figure 7. a) Very large images. b) Small unreadable images. c) Long question text. d) Keyboard obscuring content.

On the other hand, some images are much too small to read on a mobile device (see Figure 7b). Tapping the image enlarges it to full-screen and enables zooming. However, in full-screen mode it is not possible to see the question text and possible answers at the same time. Thus, users are again required to switch back and forth. In this design, tap-to-zoom could not be provided to enlarge the images displayed as answers, as the action of tapping already selects and de-selects the answer.

Students also complained about still having to scroll too much, which is the consequence of too long question texts or too many answer choices (see Figure 7c). Another problem was caused by both sides of the screen being scrollable vertically, as well as horizontally: Sometimes users try to scroll horizontally to get to the next or previous question, but instead the gesture is recognized as a vertical scroll by the application, and vice versa. This caused a lot of frustration.

Some users criticized the text-box questions because the on-screen keyboard takes up too much space, obscuring the content (see Figure 7d). Confusion was also caused by the problems arising from splitting drop-down questions into multiple-choice questions, as described in the previous section.

Further improvement of the content could solve most of these problems. However, it will likely require manual adjustment of the data. Possible improvements can target data input, as the input mask in the Moodle quiz module is not intended to produce mobile-friendly data. On the other hand, we continued working on the improvement of usability characteristics of this quiz application, and re-designed and re-developed it. The new version is the UML-Quiz^{Android} that we want to present in the following section.

5. UML-QUIZ^{ANDROID}: THE APPLICATION

The new version of the UML-Quiz^{iOS} is an Android application (Kletzander and Kögler, 2014). It supports the learning process of students in the same course as UML-Quiz^{iOS} (von Franqué, 2013). It helps to learn differences between diverse modeling techniques. It is playful and very informative at the same time. We ported the main content and business logic of the UML-Quiz^{iOS} and improved its usability based on presentation and interaction mechanisms. Figure 8 shows the main interfaces of the UML-Quiz^{Android}.

On the one hand we applied Back-Button. On the other, we used the Action Bar feature to display a tutorial for the current quiz or to switch to a settings menu for changing default settings of the displayed round of the quiz.

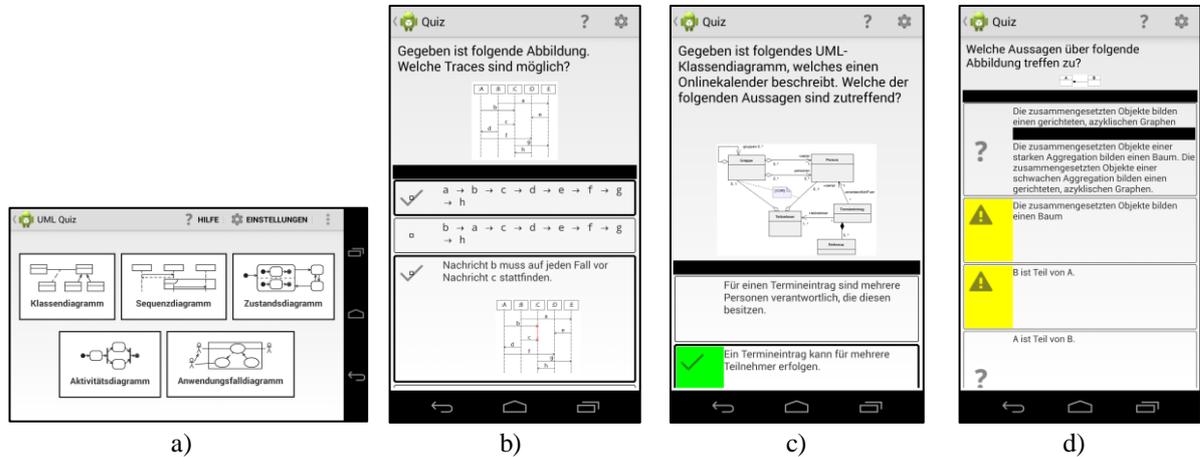


Figure 8. UML-Quiz^{Android}: The main menu to choose between different UML diagrams to start a quiz (a), the question page showing the question and options for answers (b), the feedback page showing the correct answer (with the color green) (c), and the explanations to each wrong answer (with the color yellow) (d).

6. EVALUATION OF UML-QUIZ^{ANDROID}

The development and evaluation process of UML-Quiz^{Android} has also involved a group of students as real users as well as several persons with no university degree as potential users of the application. We had in total 12 users: They were people between 25 and 55 years old, some used similar applications on their own mobile devices, and some with the aim to learn certain content in a very efficient way. These users evaluated the different interaction possibilities and helped the development team to make decisions in critical moments. After having several evaluation sessions with our users, different functions and display options have been updated. Some parts of the final version are presented in Figure 8.

The most important change we made is the orientation of the screen that shows the question and answers (Figure 8b, 8c, 8d). There are several reasons for that: This quiz is available on a mobile device. Our users want mainly use it while they are traveling in the city whenever they have time, e.g., while waiting for public transportation or being in a bus or tram traveling from one place to another. One wants to keep one hand free while moving in a city. The UML-Quiz^{iOS} was only usable horizontally and that is why two hands were always occupied. This means that one had to hold the device with one hand and answer the questions with the other. We solved this restriction in UML-Quiz^{Android}.

The consequence of having the layout vertical than horizontal, we had to think about the layout of the questions and answers (Figure 9). In UML-Quiz^{iOS} the screen was divided into two parts. The left side was the area for questions and the right side for answers. Questions contained normally a question text and an image illustrating the context of the question. Answers displayed the options to choose from. Both areas were vertically scrollable, which was after a while very confusing for the users because they easily lost the overview of current

questions and answers. Additionally the application was not usable for left-hander because questions could only be answered on the right side by putting almost the whole arm all over the device. At the same time the users found that the horizontal view of the quiz was not “natural”. They expected to have the questions on the top and the answers below the question. They also required a scrollbar to scroll the view vertically. The scrollable view enables even using long text for questions and several options for answers.

Usability in terms of accessibility was improved by enabling zooming into the images displayed both in questions and answers. It is possible to display an image bigger on the screen of a smart phone by tapping on it. This is very often necessary to see the details in the images before answering the question. In UML-Quiz^{iOS} magnifying was only provided for images in questions and not in answers.



Figure 9. The question/answer interface of the UML-Quiz^{iOS} (left), the same of the UML-Quiz^{Android} (right).

Feedback to the user has also been improved. First, we modified the feedback for interaction: In UML-Quiz^{Android} checked answers are displayed with a bigger border and a checkmark on the left side (Figure 8c) while in the UML-Quiz^{iOS} only a check mark was displayed on the right side (Figure 3, right image). Second, we considered additional learning support in the feedback mechanisms of the quiz: We developed a new view for showing the result of an answered question (Figure 10). Not only showing what answer is the correct one, we also tried to explain why is an answer a correct one and the others are wrong. This helps the user to understand the reasons for answers and this way it helps learning by answering quiz questions.

The arrangement of a feedback page is of course a very delicate issue. Details about the single answer options fill in normally the whole space available on the screen, and one can easily lose the overview of the overall content of the quiz item. We used colors and symbols to make this easy for the users (Figure 10).



Figure 10. The feedback page of the UML-Quiz^{iOS} (left), the same of the UML-Quiz^{Android} (right).

In the UML-Quiz^{iOS} correct answers were green and false answers were red. If there was a feedback text to the answers this was displayed in the area of the corresponding answer. This caused naturally much more text in the little half sided area and less overview of the whole content. Moreover if an answer was tapped as true there were check marks and warning symbols. These were sometimes irritating, if a red highlighted answer had a check mark.

In addition users with a dyschromatopsia (red-green blindness) had problems to understand this way presented feedback. In our solution we combined the feedback mode with a settings menu. In the settings menu it is possible to select a color scheme, which should be used for displaying right and wrong answers. This means that people with dyschromatopsia can choose their own colors that they can differentiate. Moreover it is possible to select if the whole answer is highlighted with the chosen color or only the left side of the answer.

We changed the symbols used in UML-Quiz^{iOS}. If an answer is true a check mark is displayed on the left side. If an answer is false a cross is displayed on the left side and the answer is displayed in another color than the color used for true answers. Furthermore if there is a false answer, which is not checked by the user, the feedback view puts a call sign in front of it and colors it as a wrong answer to make this explicit for the user. In case there is a feedback text for an answer available, a question mark is displayed on the left side of the answer. If the user taps on the answer the feedback text is displayed. Tapping again closes the feedback text. This means that the user has the possibility to gradually learn why an answer is false.

7. CONCLUSION

In this paper we have shown how a native mobile application can help to improve the preparation for exams at university level. Based on the assumption that “m-learning extensions cannot be developed to an existing e-learning system without changing the pedagogy and design of the learning content” (Hewagamage et al., 2012, p.13), we designed and developed the native mobile app UML-Quiz^{iOS} and evaluated from different points of view. The results

of this evaluation show that it is possible to take existing quiz data from web-based LMS and adapt it for use on mobile devices. Based on evaluation results we developed a second version of the quiz, UML-Quiz^{Android}, on another platform and improved several usability issues we could find out in the first evaluation process. The users found the new version much better, and we learned several factors in interaction design and user interface design with respect to quiz applications on mobile devices. We could document the improvement both on interaction and implementation level.

Further research to improve the content can aim the creation of the content. Another possibility would be a semi-automatic approach to improve the content. Mockups for a new input mask exploring this idea have already been created. The pre-existing content is imported and presented in a live preview in actual size of the mobile screen. It helps the teachers to judge if the content fits the screen and, if necessary, to make adjustments to it (for example, shorten the question text). This approach has received a good response in first evaluations with teachers.

ACKNOWLEDGEMENT

We thank Marion Scholz for her ideas, support, and supervision during the creation and establishment of the UML-Quiz^{iOS}. We also thank our users for the evaluation of the UML-Quiz^{iOS} and UML-Quiz^{Android}. Without their support we would not be able to know how to improve the quiz to make it more usable for them.

REFERENCES

- Bruck, P. A. et al., 2012. Mobile Learning with Micro-content: A Framework and Evaluation. *Proceedings of 25th Bled eConference eDependability: Reliable and Trustworthy eStructures, eProcesses, eOperations and eServices for the Future*. Bled, Slovenia, pp. 527-543.
- Casany, M. J. et al., 2012. Extending Moodle Services to Mobile Devices: The Moodbile Project. *Proceedings of UBICOMM 2012: The Sixth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*. Barcelona, Spain, pp. 24-28.
- Castillo A. et al., 2012. A System for Mobile Learning: A Need in a Moving World. *In Procedia - Social and Behavioral Sciences*, Vol. 83, pp. 819-824.
- Charland A. and Leroux, B. 2011. Mobile application development: web vs. native. *In Communications of the ACM*, Vol. 45, No. 5, pp. 49-53.
- Edgar, T. F., 2007. "A course on energy technology and policy". *In Chemical Engineering Education*, Vol. 41, No 3, Summer, pp.195-201.
- Eyhbab, A-M. and Qusay, H. M., 2012. iClass: Improving the Mobile Learning Experience. *Proceedings of ICCIT 2012*, pp. 587-592.
- Febrina, F., and Adnan, A., 2013. "Combining mastermind question and quick intelligent quiz to improve the content speaking ability of the senior high school students". *In Journal of English Language Teaching* 1.2, pp.426-435.
- Giemza, A. et al., 2012. Challenges in Scaling Mobile Learning Applications: The Example of Quizzer. *Proceedings of 2012 IEEE Seventh International Conference on Wireless, Mobile and Ubiquitous Technology in Education (WMUTE)*. Takamatsu, Japan, pp. 287-291.

- Gutierrez-Martinez J. et al., 2012. E-Learning Contents Automatic Conversion. *In International Journal of Innovative Computing, Information and Control*, Vol. 8, No. 3(B), pp. 2065-2088.
- Hewagamage, K. P. et al., 2012. "M-Learning Not an Extension of E-Learning": Based on Case Study of Moodle VLE. *In International Journal of Mobile and Blended Learning (IJMBL)*, Vol. 4, No. 4, pp. 21-33.
- Kletzander, Ch. and Kögler, A., 2014. *Interaktionsmechanismen für den Lernprozess einer mobile e-Learning Applikation*. Bachelor's Thesis, Vienna University of Technology.
- Montanaro, A., 2012. *Implementing Mini Quizzes to Increase Student Learning for a Class via a Mobile Learning Application*. A Capstone Project Submitted to the University of North Carolina Wilmington in Partial Fulfillment of the Requirements for the Degree of Master of Science.
- Noyes, J., and Baber, C., 1999. "User-centred design of systems". *Applied computing*. Springer, London et al.
- Romero, R. et al., 2010. Designing Usable Educational Material for English Courses Supported by Mobile Devices. *Lecture Notes in Computer Science*, Vol. 6389, pp. 373-383.
- Schön, D. et al., 2012a. A lightweight mobile quiz application with support for multimedia content. *Proceedings of 2012 International Conference on e-Learning and e-Technologies in Education (ICEEE)*. Lodz, Poland, pp. 134-139.
- Schön, D. et al., 2012b. Integrating a Lightweight Mobile Quiz on Mobile Devices into the Existing University Infrastructure. *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2012*. Denver, USA, pp. 1901-1907.
- Selvarajah, K. et al., 2013. Native Apps versus Web Apps: Which Is Best for Healthcare Applications? *Lecture Notes in Computer Science*, Vol. 8005, pp. 189-196.
- Tabata, Y. et al., 2010. An iPhone quiz system for learning foreign languages. *Proceedings of 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR)*. Wuhan, China, pp. 299-302.
- von Franqué, A., 2013. *Optimizing quiz-centered e-learning content for usability on mobile devices*. Bachelor's Thesis, Vienna University of Technology.
- Wasserman, A. I. et al., 2010. Software engineering issues for mobile application development. *Proceedings of the FSE/SDP workshop on Future of software engineering research*. Santa Fe, USA, pp. 379-400.