Teaching Advanced Web Technologies with a Mobile Learning Companion Application

Christopher Krauss
Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany,
ckr@fokus.fraunhofer.de

Agathe Merceron
Beuth Hochschule, Luxemburger Str. 10, 13353 Berlin, Germany,
merceron@beuth-hochschule.de

Truong-Sinh An
Beuth Hochschule, Luxemburger Str. 10, 13353 Berlin, Germany,
sinhan@beuth-hochschule.de

Miggi Zwicklbauer
Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany
mzw@fokus.fraunhofer.de

Stephan Steglich
Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany
sts@fokus.fraunhofer.de

Stefan Arbanowski
Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany
sta@fokus.fraunhofer.de

ABSTRACT

The Learning Companion Application was actually designed to fit the needs of master craftsmen in a blended learning Energy Consultant Training at the chamber of crafts. It supports mobile learning particularly through its responsive design and recommendation engine. However, its design follows the recommended practice taught in a university course for master students about Advanced Web Technologies. That is why we introduced the same application for this computer science course to provide students with a contextual and situated learning experience: students learn with the help of a system that implements many concepts they have to learn. Topics, such as HTML5, the development of responsive web applications and recommender systems, are introduced in the lecture and can be experienced as real world examples by the students in the learning app as well. Similar to common learning management systems, our Learning Companion Application offers the lecture materials as digital media assets, such as texts, source code, animations or videos. In addition, the application tracks the interactions of the students in order to give overviews of the learners' knowledge levels on the different learning objects at every time, in order to identify learning weaknesses to improve teaching with the help of a learning analytics module. It can recommend appropriate learning objects which fit the predicted knowledge and the current situation of the learner, e.g. available time for learning. This paper presents taught concepts in the lecture and their implementation in the Learning Companion Application as well as a study of the interaction and learning behavior of the computer science students.

CCS CONCEPTS

• **Information systems → Recommender systems**; *Users and interactive retrieval*; Environment-specific retrieval

KEYWORDS

Adaptive Learning; Recommender Systems; Web Technologies

1 INTRODUCTION

The Learning Companion Application (LCA) has been designed and is being developed to fit the needs of full-time employees who take part in an Energy Consultant training in a Chamber of Crafts [1]. It supports mobile learning particularly through its responsive design and recommendation engine. LCA targets technical affine users without requiring expert knowledge. It is independent of any topic and any institution and, therefore, can be used in other contexts and for other courses.

The Technische Universität Berlin offers a course on Advanced Web Technologies (AWT) and targets master computer science students. Five technical experts teach in 12 presence lectures nine topics that are of interest for future web developers – from web technology basics, such as HTML, over media delivery and protection, to personalization through recommender systems. Until the winter term 2015/2016, the presented slides were exported as PDF and uploaded to a Learning Management System (LMS) called Moodle after the lecture. Students downloaded and often printed the PDF files in order to prepare for the final test with 50 multiple-choice questions. Since October 2016 LCA replaces Moodle. Thereby, LCA as a system implements many topics taught in the AWT course and can be used to demonstrate the realization of concepts students have to grasp and reflect on – this approach is known from situated learning. Students need to acquire knowledge about computer science concepts, analytical understanding of problems and requirements, as well as the ability to transfer and apply these concepts.

The remainder of this paper is structured as follows: Section II discusses related work. Section III presents the Learning Companion Application giving implementation details and describes features of LCA that are used in the AWT course. Following that, some analyses on the use in the AWT course are shown. The paper concludes with a summary and an outlook.

2 RELATED WORK

A set of Learning Management Systems (LMS) based on well-established specifications are frequently used by different organizations. Blackboard, Moodle, and Desire2Learn are three examples for LMSs [2] especially used for university courses, vocational training, and further professional education. Two main requirements have led to developing LCA instead of adopting an open source LMS. The first requirement is that learning resources have to be stored in one central repository to be accessed without replication when a course is taught in different institutions. The second requirement is to offer a recommendation service to learners which selects appropriate contents that fit into the available time slot of a student and an analytics service to different stakeholders, in particular to teachers. This second requirement has consequences for learning resources metadata and for tracking users' interactions.

Predicting appropriate learning objects that fit the user's learning needs with the help of a recommender system requires different data compared to traditional e-commerce recommendations: Students are not searching for items they like, but for items they need to learn. "Instead of buying a product and then owning it, learners achieve different levels of competencies that have various levels in different domains" [3]. Students shall get reasonable content hints in order to be motivated, recall existing knowledge and illustrate, visualize and represent new concepts and information [3]. A unique feature of our recommender service is to take forgetting into account (that was first presented in [4]). Moreover, in informal learning settings, students often need to select and answer questions that fit their prior knowledge and skills. Adaptive systems can, therefore, offer adaptive navigation support which, for instance, works well for the selection of tasks of optimal difficulty in computer science courses, such as for Java Programming [5].

Han and Shin have investigated the use of mobile LMSs in an online university and found out that "the use of a mobile LMS positively influenced online students' academic achievement" [6]. Using LCA gives hands-on-practice to students, which could raise their interests in the topic, and positively influence their performance as observed in a related context by Wu, Hsu, Lee and Wang [7].

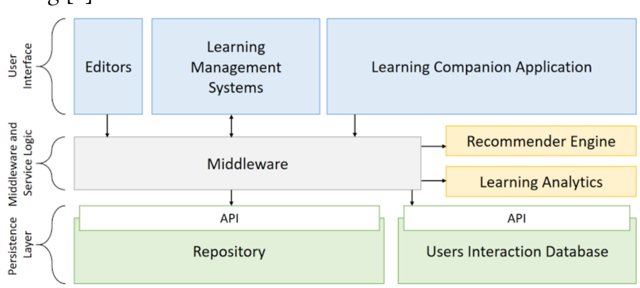

Figure 1: LCA Infrastructure

3 LCA FEATURES TAUGHT IN AWT

The Learning Companion Application (LCA) is based on a reusable generic infrastructure [8]. Figure 1 illustrates the service-oriented architecture and interworking of its core components. Each component is encapsulated and only connected to the *middleware,* which, in turn, exchanges information in universal specified formats via interfaces.

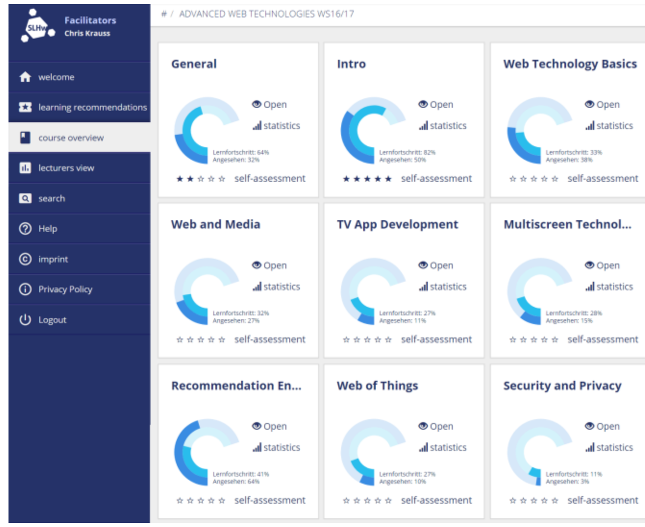

Figure 2: The LCA user interface showing the chapters of the course AWT.

The World Wide Web Consortium (W3C) compliant *Learning Companion Application* plays a key role in that infrastructure as it is the entry point for students to access courses, learning objects and triggers the tracking of all relevant user interactions. Figure 2 visualizes LCA with the AWT course outline. The Twitter's Open Source Framework Bootstrap [1] has been used to make it a responsive web application. We teach such responsive concepts and this framework in the chapter *Multiscreen Technologies and Standards*. That means LCA can be displayed on every screen size (e.g., desktop, laptop, tablet or smartphone) by default and automatically adopts the screen layout to the available size, thus enabling mobile learning. Moreover, teachers use LCA to get access to the *Learning Analytics* module.

The *repository* acts as a digital asset store, which essentially holds course structures, learning objects and their metadata. At the lowest level, a learning object (LO) is a digital item like a piece of text, or a video, a screencast, a test, a piece of program, and so on, all with at least one learning objective. Low-level LOs in LCA are stored as LTI-Tools [9] to be usable with other systems. For the same interoperability reason, questions and tests are specified according to the IMS QTI specification [10]. For AWT, we present a set of multiple choice questions at the end of each learning unit. Metadata are based on the Learning Object Metadata (LOM) specification and contain among others the learning objective and the typical learning time of that LO [11] that is processed to find the best fitting item in the demanded situation. Low-level learning objects can be bundled into bigger learning objects like learning units or chapters, and this can be repeated hierarchically. That way, low-level LOs can be reused in several courses. An LTI player, that is presently part of the repository, renders the learning units and QTI specified tests. It is programmed in HTML5 and uses CSS based on profiles loaded at runtime to skin the appearance of LOs according to the client institution. One of the

[1] Bootstrap. See: http://getbootstrap.com/

first lectures is *Web Technology Basics* and refers to the required background knowledge on HTTP as well as basics of the HTML5 development technologies such as HTML, CSS, and JavaScript. It uses an accordion view that displays only one LO at a time to help users stay focused on that LO. To see the next LO, a user has to click its title. The player includes further interactions possibilities to enhance the learning experience of users. First, it generates at the beginning and at the end of each learning unit the list of the learning objectives of that unit. Users can assess how much they know each learning objective. These self-assessments encourage learners to reflect on their previous knowledge, and on how much they know after learning the unit. Second, following a best practice expressed in Hsu, Y.-C., & Ching, Y.-H. [12], users can rate the estimated learning time of each learning unit. Finally, users can give feedback, report mistakes and discuss in a forum each learning unit.

Different *editors* have been provided as easy-to-use web applications for instructional designers to create and manage learning media, questions, and its metadata. In turn, users' interactions with any LO in the Learning Companion Application including LO selection, answers given to exercises and so on, are stored according to the accepted terms of conditions by the user, see component *users interaction database* in Figure 1. Interactions are persisted using the xAPI specification [13] in a learning record store, presently learning locker[2]. The *learning analytics* service is designed for teachers. They can customize their own dashboards to get at a glance how many students are mastering, are in the process of mastering or do not master at all a particular unit or LO [1].

In the AWT chapter *Recommender Systems*, we teach different Content-based and Collaborative Filtering approaches as well as Machine Learning techniques for the prediction of appropriate items. At the same time, we present learning recommendations in LCA by applying the same concepts: The Smart Learning Recommender [14] aims at identifying the next, most suitable LO for the requesting student based on the calculated knowledge level and learning need for that item. The learning recommender system matches each student with each available learning object in the taken course and aggregates all xAPI statements with time information. Our model includes various factors (e.g. processing time and duration, performance in exercises, self-assessments, and even the forgetting effect) supported by different statements, and represents the personal knowledge level on an LO in a range from [0,1], where 0 is the lowest possible knowledge and 1 the highest. It determines the learning need, which is inversely proportional to the knowledge level – the lower the knowledge level, the higher the learning need of the current item for a student at the given time and vice versa. We do predict the user's knowledge level on all content items and, in turn, recommend relevant learning objects adapted to the learning need of the user. At the end, the most relevant contents are presented to the user in different categories, such as "You should prepare that content for the next lecture", "We identified some weaknesses in that exercise", or "You might have forgotten this", see [14] for a detailed explanation. According to the situated learning paradigm, presented learning items can additionally be filtered by the intended time of consumption given by the instructors.

4 USAGE PATTERNS

The *Learning Companion Application* was used for the first time during the winter term 2016/2017 to teach the AWT course. A group of 145 students enrolled in the course and 99 of them used LCA. Five teachers and five more administrators and content developers also used the system. Their interactions with the LCA are included in the analysis. We presented almost 1,000 learning objects grouped in 105 sub-topics of the 9 main chapters. Besides a few videos and some interactive exercises, learning objects are mainly sets of slides that can be downloaded.

We collected over 92,000 xAPI statements as well as exam results of students who participated the exam. Thereby, a total amount of 84% of all interactions on LCA were done on Desktop computers, the rest on mobile devices[3]. In contrast to another experience concerning a blended learning course performed with LCA at a Chamber of Crafts where almost 40% of all interactions were performed using a smartphone or tablet, the AWT usage pattern surprisingly did not correspond to the paradigm of mobile learning at smartphones or tablets. This can be explained by the fact that the computer science students prefer learning on the same device that they use to program: Observations show, most of them used their laptops during lectures, at home, and on the ways. LCA is most frequently used during the working hours from 8:00 am to 6:00 pm – with two major exceptions: Computer science students learn a lot during noon (especially Thursdays during the lecture times 12 pm – 2 pm) and they start interacting with the app again in the evening between 7 pm and 1 am.

At the end of the course, students had to answer 50 multiple choice question plus 5 additional bonus questions in an offline final test. In the LCA course, we asked around 50% new questions compared to last year's course without LCA. 39 students attended the final test in the iteration with Moodle and 75 in the iteration with LCA. Which means that we almost doubled the exam participation in the second course. However, we do not have data on the initial course enrollments of the first course and, thus, cannot judge about an improvement of the drop-out rate. The average percentage of correctly answered questions are almost equal: 84% (Moodle) and 83% (LCA), so the average marks: 2.0 (with Moodle) and 1.9 (with LCA). Keep in mind that 1.0 is the best and 5.0 is the worst. Only one student failed each course and then passed the re-sit.

The collected xAPI statements (on elapsed time per LO and assessment scores in exercises) have been processed in order to discover typical learning patterns [15]. We identified 3 clusters, labeled as *Completers*, *Strong Starters* and *Auditors* first introduced by Kizilcec [16]. Subsequently, we calculated the average mark in the final test per cluster. The *Completers'* cluster contains the 26

[2] Learning Locker. See: http://www.learninglocker.net

[3] Operating Systems: 43% Windows, 31% Mac and 8% Linux, 11% iOS, 6% Android;

students who accessed all LOs and solved correctly all exercises. Their average mark of 1.5 is better than the total average of 1.9. The 9 *Strong Starters* had an average mark of 2.0 (almost the total average). At a certain point in the course, *Strong Starters* engaged less with LCA than *Completers*. The *Auditors'* cluster, in contrast, shows students who infrequently used LCA and is the biggest cluster with 64 students – including all students who did not participate in the final test. They reached an average mark of 2.2 which is less good than the total average.

5 CONCLUSION AND FUTURE WORK

The Learning Companion Application is a novel Learning Content Management System that integrates services such as a recommendation service for learners and aims at making the learning process of students more convenient. We have used LCA in a university course about Advanced Web Technologies. The responsive web application gives access to the lecture material and, in addition, presents technical features, such as web development using HTML5, the use of Frameworks or even more complex techniques, such as learning recommendations through recommender systems. All user interactions are stored as xAPI statements.

A valuable feature of LCA is the possibility to get a deeper understanding of learning patterns due to the collected usage data. Therefore, we clustered the learners according to their interaction with LCA. As a result, we determined typical learning behaviors that correspond to different average marks in the final exam. Moreover, in one cluster we even identified all students who did not appear at this exam. Thus, LCA allows to better understand the learners which would not have been possible with the before used PDF files of lecture slides. It is planned to further explore typical behaviors while using LCA, as a basis to adapt the learning recommender system and learning analytics module to different behavior types.

ACKNOWLEDGMENTS

This work is partially supported by the German Federal Ministry of Education and Research grant number 01PD14002B. The authors would like to especially thank the whole team involved in the realization of this project.

REFERENCES

[1] An, T-S., Dubois, F., Manthey, E., Merceron, A. “Digitale Infrastruktur und Learning Analytics in Co-Design”. In the Workshop Learning Analytics, co-located with the 13th DeLFI Conference, Potsdam, Germany, September 11, 2016.

[2] Taneja, Shilpi, and Anita Goel. “MOOC providers and their strategies.” International Journal of Computer Science and Mobile Computing 3.5 (2014): 222-228.

[3] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, and R. Koper, “Recommender systems in technology enhanced learning”. In Recommender systems handbook. Springer, 2011, pp. 387–415.

[4] Krauss, Christopher; Chandru, Rakesh; Merceron, Agathe; An, Truong-Sinh; Zwicklbauer, Miggi; Arbanowski, Stefan. “You Might Have Forgotten This Learning Content! How the Smart Learning Recommender Predicts Appropriate Learning Objects”. In International Journal On Advances in Intelligent Systems, 2016 no 3&4. Xpert Publishing Services, December 31, 2016

[5] Hsiao, I-H., Sergey Sosnovsky, and Peter Brusilovsky. "Guiding students to the right questions: adaptive navigation support in an E-Learning system for Java programming." Journal of Computer Assisted Learning 26.4 (2010): 270-283.

[6] Han, I. and Shin, W.S. 2016. “The use of a mobile learning management system and academic achievement of online students”. Computers & Education 102 (Nov. 2016), 79-89.

[7] Wu, H-T., Hsu, P-C., Lee, C-Y., Wang, H-J., Sun, C-K. „The impact of supplementary hands-on practice on learning in introductory computer science course for freshmen”. Computers & Education 66 (Aug. 2013), 1-8.

[8] Krauss, Christopher; Merceron, Agathe; An, Truong-Sinh; Zwicklbauer, Miggi; Arbanowski, Stefan. “The Smart Learning Approach - A mobile Learning Companion Application”. In The Eighth International Conference on Mobile, Hybrid, and On-line Learning 2016. April 24 - 28, 2016 - Venice, Italy.

[9] IMS LTI, “Learning Tools Interoperability Specification”, IMS Global Learning Consortium, 2015.

[10] IMS QTI, “IMS Question & Test Interoperability Specification”, IMS Global Learning Consortium, 2005.

[11] IMS LOM, “Learning Resource Meta-data Specification”, IMS Global Learning Consortium, 2006.

[12] Hsu, Y.-C., & Ching, Y.-H.: “Mobile app design for teaching and learning Educators' experiences in an online graduate course.” In The International Review of Research in Open and Distance Learning, 14(4), 2013.

[13] Kevan, J. & Ryan, P. “Experience API: Flexible, Decentralized and Activity-Centric Data Collection in Technology, Knowledge and Learning”. Springer Science+Business Media Dordrecht, 2015, 2211-1662.

[14] Krauss, Christopher. “Smart Learning: Time-Dependent Context-Aware Learning Object Recommendations.” In The 29th International Florida AI Research Society Conference (FLAIRS-29), AAAI, Key Largo. 2016.

[15] An, Truong-Sinh; Krauss, Christopher; Merceron, Agathe;. “Can Typical Behaviors Identified in MOOCs be Discovered in Other Courses?”. In: Proceedings of The 10th International Conference on Educational Data Mining (EDM 2017), 25 – 28 June 2017, Wuhan, China.

[16] Kizilcec, R.F., Piech, C., and Schneider, E. 2013. “Deconstructing disengagement: analyzing learner subpopulations in massive open online courses”. In LAK'13, April 8-12, Leuven, Belgium. ACM, p. 170-179.