The Smart Learning Approach

A mobile Learning Companion Application

Christopher Krauss[1], Agathe Merceron[2], Truong-Sinh An[2], Miggi Zwicklbauer[1], Stefan Arbanowski[1]

[1] Fraunhofer FOKUS
Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany
email: {forename}.{lastname}@fokus.fraunhofer.de

[2] Beuth Hochschule für Technik Berlin
Luxemburger Str. 10, 13353 Berlin, Germany
email: {lastname}@beuth-hochschule.de

***Abstract*— The aim of the ongoing research project "Smart Learning in Vocational Training" is to introduce a novel mobile Learning Companion App in order to support a blended-learning approach in the energy consultant training. Thereby, students can keep track of their individual predicted knowledge level on different learning objects at every point in time and get personalized learning recommendations based on the determined learning need value. Moreover, teachers make use of Learning Analytics in order to get an overview of students' progress and so, be aware of possible weaknesses. This work in progress paper introduces the main ideas of the overall architecture and technical concepts in that project.**

Keywords-Smart Learning; Learning Companion; Recommendation Engine; Learning Analytics.

I. INTRODUCTION

Career advancement requires employees to continuously update their skills and, in many cases, to document their up-to-date knowledge with a certificate. In Germany, the Chamber of Crafts provides numerous vocational trainings that lead to the obtainment of a certificate. Trainees are full-time working professionals. Till now most of the trainings are fully face-to-face. The aim of the project "Smart Learning in Vocational Training" [1] is to introduce a blended-learning approach in the energy consultant training. Learning material is being currently structured and developed using different digital media: texts, animations, screencasts, videos, etc. During face-to-face phases trainees learn hands-on with a professional. To prepare and to review face-to-face learning, they can access online what they need when they need it with the help of a novel mobile web application called the Learning Companion App (LCA).

LCA provides trainees with access to learning materials and stores user interactions according to an opt-in procedure. In that respect, it is similar to a learning management system (LMS) that can also run on desktops or mobile devices. However, LCA makes use of a set of server-side software components. The full system integrates a recommendation engine and a learning analytics module. Based on the stored interactions and making use of the recommendation engine, LCA shows trainees their progress and recommends them learning material after calculating their current learning need. This feature of actively guiding learners in their learning through the learning material by means of recommendations distinguishes LCA from common LMS. A learning analytics module is being developed for instructors/ instructional designers. As LMS do, LCA differentiates between users according to their role. Instructors can use LCA to access dashboards that provide them with an overview of learners' progress. Thus, before face-to-face meetings for example, instructors can review the advancement of trainees and adapt their teaching. Other dashboards show the progress of learners when completing self-estimation of learning objectives before and after completing a learning unit. These dashboards, also accessible through LCA, are useful for instructional designers to judge and, possibly, review the quality of learning objects (LO).

This paper is organized as follows: Section II introduces related work on recommendation techniques and analytics for education. Section III explains the overall architecture of the currently implemented components and Section IV focuses especially on the Learning Companion App and the underlying techniques, recommendation and learning analytics modules. The paper concludes with a short summary and an outlook on the planned evaluations.

II. RELATED WORK

Many modern web services, such as movie portals and e-commerce services, but also online learning courses, offer a vast amount of items. So, users quite often lose the overview and get buried in information. A recommendation engine aims at identifying the most relevant items for a specific user that fit his/ her individual needs and thus, makes his/ her interaction on that web service more efficient.

Manouselis et al. describe learning and recommending learning objects in a digital environment, such as in Mobile, Hybrid and On-line Learning, as "an effort that takes more time and interactions compared to a commercial transaction. Learners rarely achieve a final end state after a fixed time. Instead of buying a product and then owning it, learners achieve different levels of competences that have various levels in different domains" [2]. The learner shall find appropriate content for the preparation of a lesson in order to: 1) Be motivated, 2) Recall existing knowledge and 3) Illustrate, visualize and represent new concepts and information [2]. Moreover, recommender systems in On-line Learning can also be used for actual teaching as well as knowledge evaluation and assessment. In a closed corpus, such as a course, students have to learn all relevant objects in

 ISBN: 978-1-61208-471-8

order to pass the final exam, no matter if they are interested in it or not.

Therefore, recommendation engines shall also respect the intrinsic and extrinsic motivation of students. The "Moodle Recommender System" [3] showed the significant role of learning paths and completion rates of learning objects that are of interest for recommender systems to assist other learners. In [4], Pelanek et al. evaluated the closed correlation between multidimensional student skills and the timing of problem solving in intelligent tutoring systems that is "useful for automatic problem selection and recommendation in intelligent tutoring systems and for providing feedback to students, teachers, or authors of educational materials" [4]. Manouselis et al. listed an existing set of learning recommenders, but more than half of these systems were still at a concept or prototyping level [2]. Only 10 of the listed systems have been evaluated with real participants and most of the systems focus on past user behavior or on preferences and interests in order to predict new learning content. Unfortunately, it seems that in the area of mobile and online learning, no recommender system covers the time aspect of changing knowledge levels - even though it seems to have great impact. LCA will continuously track the learning behavior of individual users over the whole course period, forecast there learning need on specific LOs at every point in time and recommend appropriate learning objects.

Learning dashboards to support teachers in different teaching contexts have been proposed by different authors. All dashboards provide some form of visual summary of the use of learning materials by learners [5][6]. The visual summary proposed in [6] is particularly interesting: All interactive exercises and questions in the course have been tagged with learning objectives. Based on students' performance the Learning Dashboard provides instructors with an overview of all learning objectives represented by a bar each. The proportion of green, yellow and red in the bar reflects the proportion in the class; the grey portion of the bar shows students with too little activity to enable a classification of their performance. As argued in [8] it is essential to establish a dialogue with future users of dashboards while designing them. Adopting such a user-centered approach and re-using analyses developed in the LeMo tool [6] dashboards are being developed that adapt elements of the overview proposed in [6] to the bigger variety of learning objects present in LCA.

III. Architecture

One focus of the Smart Learning project is to provide a reusable generic infrastructure for various users with different client devices, for different courses covering several topics - not restricted to institutions like chamber of crafts, but also usable by universities and adult education centers. While users are still managed in the LMS, learning objects are stored only once and can be shared by and accessed from various learning management systems.

Figure 1 illustrates the architecture and interworking of the core components. Each component is encapsulated and only connected to the so-called middleware, which, in turn, exchanges contents and metadata in standardized formats via standardized interfaces. Components are described in turn.

The Learning Companion App plays a key role. It is the entry point for students to access courses, learning objects and lecture dates as well as to get recommendations for the next best contents to be learnt and triggers the tracking of all relevant user interactions. It is a responsive web application to be displayed on regular modern desktop web environments, but especially on smartphones and tablets to enable mobile learning. The application gives everywhere-and-everytime-access to all learning objects. Students can optimize their free time by filtering the most important items that fit in the available time period - for instance when waiting for the bus or going to class. Moreover, teachers use the LCA to get access to the Learning Analytics module.

The LMS is used to register and manage all users and offers discussion forums. In order to allow a consistent interaction with all components, the students (and teachers) credentials of the existing Learning Management Systems are required to authenticate at the Learning Companion App. This kind of single-sign-on approach is implemented in the

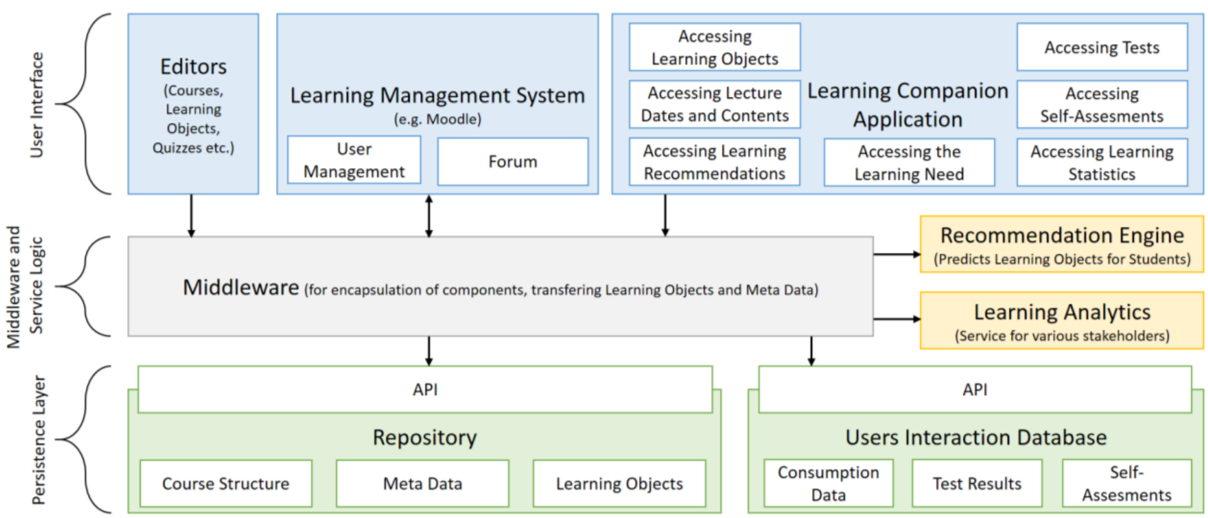

Figure 1. Architecture of the Smart Learning Project

 ISBN: 978-1-61208-471-8

middleware and transparent to users.

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 is a simple document (technically in HTML), a video, a screencast, a test and so on, all with at least one learning objective. Low-level LOs are stored as LTI-Tools [9] as to integrate them with different LMS. Moreover, questions and tests are specified according to the IMS QTI specification [10]. Low level learning objects can be bundled into bigger learning objects, and this iteration can be repeated. In the current energy consultant course, low-level LOs are combined in learning units, learning units in sections and a few sections make up the course. That way low level LOs can be reused in several courses. A so-called manifest file is created to bundle the LOs together. A player that is presently stored in the repository renders the learning units and QTI specified tests. Further, the player generates automatically self-assessments questions using the learning objectives contained of a learning unit. The metadata associated to a low level LO contains among others its learning objectives, at least one as mentioned above, average study time defined by instructors and prerequisite LOs. These data are stored using the LOM specification [11]. When LOs are combined, the metadata of the whole are generated automatically from the parts. A course structure is stored following the IMS Common Cartridge standard [12].

Different editors have been implemented as easy to use web applications for instructional designers. A LOM-Editor allows to specify the metadata of any existing LO and to store the corresponding file in the repository. A QTI-editor allows creating questions, to bundle them into tests following the QTI specification and to store them in the repository too; presently seven types of questions are available: choice, choice multiple, extended text, text entry, numeric, matrix and order. Finally, a LO-Editor allows bundling LOs into bigger ones and generating automatically the metadata file as written above.

Users' interactions with any LO are stored according to the opt-in procedure chosen by the user. Interactions are persisted using the xAPI specification [13] in the free

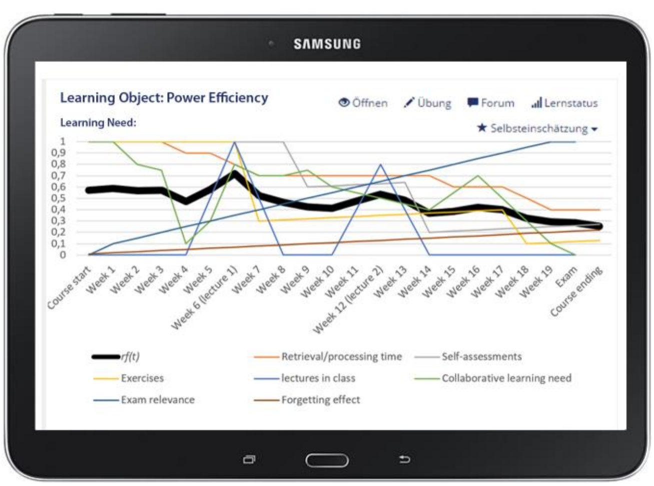

Figure 2. The mobile Learning Companion App

learning record store called learning locker[1].

The recommendation engine and the learning analytics service load the needed interaction data in regular intervals in order to determine students' performance. The Smart Learning Recommender aims at identifying the next best, most suitable learning object for the requesting student based on the calculated knowledge level and learning need for that item. The learning analytics service, in contrast, is designed for other stakeholders. So, teachers can observe the overall progress and performance of students and figure out weaknesses in learning and understanding.

IV. RECOMMENDATIONS AND LEARNING ANALYTICS IN THE LEARNING COMPANION APP

A key role in connecting the users' interaction in LCA with the learning analytics service or recommendations engine is attributed to the formal and informal activity statements reflecting the collected user data. In recent years, the Experience API [13] with its xAPI statements continuously moves in the academic focus supporting long-term data mining. A typical statement consists of the three properties: "Actor", "Verb" and "Object". An xAPI statement can also carry the optional properties "Context" and "Results" containing more information for new insights like in the following statements:

- "StudentA (Actor) completed (Verb) Question1 (Object) in the *context* of Quiz1 in Course1 and the *result* of success with 2 attempts based on the raw score of 80 with a max score of 100 and a scale of 0.8" or
- "StudentB stopped VideoY started at position 00:01:30 in the *context* of LearningUnit2 of Course1 *resulting* in duration of 00:01:42".

The player mentioned in the preceding section as well as the app trigger the xAPI statements.

A. Learning Need and Recommendations for Students

In the Smart Learning Recommender [14], we relate each student with each available learning object in the taken course and aggregate all xAPI statements with time information into a relationship, called user-item-time-triplet. This triplet consists of various factors supported by different statements, and represents the personal knowledge level on a LO on a range from [0,1], where 0 is the lowest possible knowledge and 1 the highest. This triplet determines the learning need, which is inverse proportional to the knowledge level - the lower the knowledge level, the higher the learning need and so relevance of the current item for a student at the given time and vice versa. Presently, we consider the following factors that influence the learning need and so implicitly the recommendations:

- **Interaction** with a learning object: How much of the LO has been consumed?
- **Processing time** of a learning object compared with the pre-defined duration of the metadata and average consumption time of others.
- **Self-assessments** for this learning object.

[1] http://www.learninglocker.net/

Copyright (c) IARIA, 2016. ISBN: 978-1-61208-471-8

- Performance in related **exercises**.
- The **forgetting effect;** the knowledge level decreases over time until learning has to be repeated.
- **Pre-requisites** of the underlying learning objects.
- Timely relevance for **face-to-face lectures**.
- **Exam relevance** in contrast to optional contents.
- **Collaborative learning needs** to offset under-estimations of the current student.

At the end, all single-factor values (each in range of [0,1], as well) are weighted. The weighted average of all factors describes the total learning need of the learning object for that user. A learner can see at any time the factors calculated by the system as well as her learning need, see Figure 2. The recommendation engine predicts the most relevant items by comparing the single learning need values and retrieves an ordered list of Top-N learning objects for a student.

B. Learning Analytics as Feedback for Teachers

As mentioned in the introduction, the main dashboard should allow instructors to obtain an overview of learners' progress and recieve details on demand [15]. In our context, several instructors teach in the same training program, each being responsible for specific training units. Thus, inside a course, the overview has to be at the learning unit level. It offers details on the selected learning object. We follow the approach of [6] in associating a bar to each object with the colors green, yellow and red. Taking the example of a test made of several questions green represents how often the test has been completely solved, yellow how often it has been partially solved and red how many students have not solved it at all. Details on each question of the test show the number of correct answers, partially correct and wrong answers following a similar scheme and adapting it to the question's type. Filters allow to chose particular time periods for the dashboard.

V. Conclusion & Future Work

We introduced a novel Learning Companion App, which allows learners to access standardized learning objects everywhere and any time. Thereby, each user interaction will be persisted and processed in order to predict the individual knowledge level of students, recommend the most important learning contents for a user and allows to analyze the overall performance in the course with the learning analytics service.

As a next step, studies with trainees enrolled in this 5 months training at the chamber of crafts will be conducted with two consecutive courses in order to improve the system iteratively; the first one begins in April 2016 and the second in September 2016. These studies evaluate real world learning behavior: how LCA performs and how users accept digital learning media and individual learning recommendations. Moreover, teachers will adapt their traditional courses to a LCA supported blended-learning approach with the help of the information provided by the learning analytics module. This component will be developed further to include analyses appropriate for instructional designers.

Acknowledgment

The authors would like to thank the whole Smart Learning team for their great work and many constructive ideas. This project is sponsored by the German Federal Ministry of Education and Research.

References

[1] M. Zwicklbauer, C. Krauß, A. Merceron, M. Scharp, and J. Kania, "Smart learning: the digital learning companion for vocational training," [Smart Learning: Der digitale Lernbegleiter für die berufliche Bildung], Proceedings of DeLFI 2015, Gesellschaft für Informatik Publisher, Munich, 2015, pp. 227-232.

[2] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, and R. Koper, "Recommender systems in technology enhanced learning," Recommender systems handbook, Springer, 2011, pp. 387-415.

[3] J. Janssen et al., "Self-organising navigational support in lifelong learning: how predecessors can lead the way," Computers & Education, Elsevier, vol. 49, pp. 781-793, 2007.

[4] R. Pelanek and P. Jarusek, "Student modeling based on problem solving times," International Journal of Artificial Intelligence in Education, Springer, New York, vol. 25, 2015, pp. 493-519.

[5] K. Verbert et al., "Learning dashboards: an overview and future research opportunities," Personal and Ubiquitous Computing, vol. 18(6), 2013, pp. 1-16.

[6] M. Elkina, A. Fortenbacher, and A. Merceron, "The learning analytics application LEMO - rationals and first results," International Journal of Computing, vol. 12(3), 2013, pp. 226-234.

[7] A. Dollár and P. Steif, "Web-based statics course with learning dashboard for instructors," Proceedings of Computers and Advanced Technology in Education (CATE 2012), Napoli, 2012.

[8] R. Martinez-Maldonado, A. Pardo, N. Mirriahi, K. Yacef, J. Kay, and A. Clayphan, "The LATUX workflow: designing and deploying awareness tools in technology-enabled learning settings," Proceedings of the 5th International Conference on Learning Analytics and Knowledge, ACM, New York, 2015, pp. 1-10.

[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] IMS CC, "IMS Common Cartridge Specification," IMS Global Learning Consortium, 2015.

[13] J. Kevan and P. Ryan, "Experience API: flexible, decentralized and activity-centric data collection in technology," Knowledge and Learning, Springer Science Business Media Dordrecht, 2015, pp. 2211-1662.

[14] C. Krauss, "Smart learning: time-dependent context-aware learning object recommendations," Proceedings of the 29th International Florida AI Research Society Conference (FLAIRS-29), AAAI, Key Largo, 2016, in press.

[15] B. Scheiderman, "The eyes have it: a task by data type taxonomy for information visualizations," Proceedings of the 1996 Symposium on Visual Languages (VL'96), IEEE, 1996, pp. 336 – 34.