You Might Have Forgotten This Learning Content! How the Smart Learning Recommender Predicts Appropriate Learning Objects

Christopher Krauss *, Rakesh Chandru ‡, Agathe Merceron †,
Truong-Sinh An †, Miggi Zwicklbauer * and Stefan Arbanowski *

*Fraunhofer Insitute for Open Communication Systems
Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany
Emails: christopher.krauss@fokus.fraunhofer.de,
miggi.zwicklbauer@fokus.fraunhofer.de,
stefan.arbanowski@fokus.fraunhofer.de

†Beuth Hochschule fuer Technik Berlin
Luxemburger Str. 10, 13353 Berlin, Germany
Emails: merceron@beuth-hochschule.de,
sinhan@beuth-hochschule.de

‡DILAX Intelcom GmbH
Alt-Moabit 96b, 10559 Berlin, Germany
Email: rakesh.chandru@dilax.com
(Rakesh Chandru contributed to this paper as employee of Fraunhofer FOKUS)

Abstract*—In digital learning environments, analysis of students' interactions with the learning objects provides important information about students' behavior. This can lead to a better understanding of the learning process and thus, optimizes teaching and learning. The aim of the ongoing research project "Smart Learning" is to introduce a novel mobile Learning Companion App in order to support a blended-learning approach in the training of Energy Consultants at the Chamber of Crafts Berlin, university lecturers as well as company internal summer schools. 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 expected learning progress. Moreover, teachers make use of learning analytics in order to get an overview of students' progress and so, be aware of possible weaknesses. The relevance of learning items change significantly over the period of a course – students may start with a low knowledge level, learn specific topics and afterwards slowly forget the lessons learned. Thus, learning environments require a new prediction paradigm for recommender systems: The relevance score of an item depends on different contextual factors. Especially forgetting plays a crucial role as people tend to forget lessons learned. Information stored in individual's memory is either erased or cannot be retrieved due to several reasons. This process is influenced by different parameters: external ones, such as the item's media type, difficulty level and so on, as well as the individual's memory strength. This paper introduces the main ideas of the overall system, its architecture, app design and mathematical concepts as well as a novel approach to include the effect of forgetting in a time-dependent recommender system that is specialized in the area of Technology Enhanced Learning.

***Keywords*–Smart Learning; Forgetting; 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 (Handwerkskammer) provides numerous vocational trainings that lead to the obtainment of a certificate. Trainees are full-time professionals. Until 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 training of Energy Consultants. Learning material is currently structured and developed using different digital media: texts, animations, screencasts, videos. During lecture 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 [2]. 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 through the material by means of recommendations distinguishes LCA from common LMS. A learning analytics module is being developed for instructors/instructional designers. As LMSs do, LCA differentiates between users according to their role.

Instructors can use LCA to access dashboards that give them 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 the individual course elements, named 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. Afterwards the learning analytics and recommender modules are introduced and followed by a detailed explanation of the Smart Learning Recommender. Especially the forgetting effect in Section VIII shows a novel approach in Technology Enhanced Learning. We conducted two studies resulting in a mathematical formula that represents an approximation of human forgetfulness. However, it is still a challenge to compare the overall system with existing ones in academia. These issues are discussed in section IX. The paper concludes with a short summary and an outlook on planned further 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 content items. Users quite often lose the overview and get buried in details. A recommendation engine aims at identifying the most relevant items for a specific user that fit the individual needs and thus, makes the interaction on that web service more efficient.

A. Recommender Systems in Technology Enhanced Learning

Olga C. Santos [3] discusses different barriers in Technology Enhanced Learning (TEL) and describes six factors that influence this domain. The factors are motivation, platform usage, collaboration with class mates, accessibility considerations when contributing, learning style adaptations and previous knowledge. The results show a decrease of the need to consume all available learning objects, when getting learning recommendations.

Learning and recommending learning objects in a digital environment, such as in mobile, hybrid and online learning, is "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" [4]. 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 [4]. Moreover, recommender systems in On-line Learning can also be used for actual teaching as well as for knowledge evaluation and assessment.

Manouselis et al. [4] argued that more than the half of all published recommender systems in the area of Intelligent Learning Technologies were still at a prototyping or concept level and only 10 have been evaluated in trials with real participants. Most of these systems are designed to predict items in a closed system using the two-dimensional Collaborative Filtering user-item-matrix, such as "CourseRank" [5] of the Standford University, "Altered Vista" [6] that uses Association Rules of frequently used learning objects in courses or "RACOFI" [7], a rule-applying collaborative filtering system "that assists on-line users in the rating and recommendation of audio (Learning) Objects". However, these recommenders only work on a flat item hierarchy and without time or extended context data. Nevertheless, it seems to be very important to include the intrinsic and extrinsic motivation of students, in terms of "pedagogical aspects like prior knowledge, learning goals or study time" [4].

B. Context-Sensitive Learning Recommendations

In order to improve the online learning environment, Hayriye Tugba Ozturk [8] proposes a method of sequential analysis of discussions among students and teachers in a Learning Management System (LMS). A similar kind of research is carried out by Angel F. Agudo-Peregrina [9], where the interactions in the learning management system is analyzed, based on an agent (student-student, student-teacher, student-content), frequency of use (access to learning resources, creation of class interactions and so on) and participation mode (active or passive) for predicting students' academic performance.

The "APOSDLE" recommender service [10] uses an extended user profile as input for appropriate content recommendations and a web tool for ontology evaluation for identifying semantic similarities. The "Multi-Attribute Recommendation Service" [11], in turn, uses ratings on different attributes and criteria for the same learning object in order to calculate proper recommendations. Moreover, Huang et al. [12] uses a Markov chain model to calculate sequences of learning objects and recommend learning paths, and the "Learning Object Sequencing" [13] uses a novel sequencing rule algorithm by processing topical ontologies. The "Moodle Recommender System" [14] shows the significant role of learning paths and completion rates of learning objects that are of interest for recommender systems to assist other learners. Our Smart Learning Recommender engine follows a similar approach by taking the context, in terms of various factors, as well as item sequences and hierarchies into account.

C. Time-dependent Recommenders

Drachsler et al. [15] underline the significance of the attribute that represents time taken to complete learning objects. Pelanek et al. [16] evaluated the closed correlation between multidimensional student skills and the timing of problem solving that "may be useful for automatic problem selection and recommendation in intelligent tutoring systems and for providing feedback to students, teachers, or authors of educational materials". However, most research on time-dependent recommendation engines have been done in the area of movie predictions: Liang Xiang [17] demonstrates four different time factors that affects recommending videos. They are the change in interest of a whole society over the time, changes in users' rating habits, changes in items' popularities and changes in users' attitudes towards some type of items. In a study conducted by Liang He [18], neighborhood group and user preferences are computed for different time intervals. In

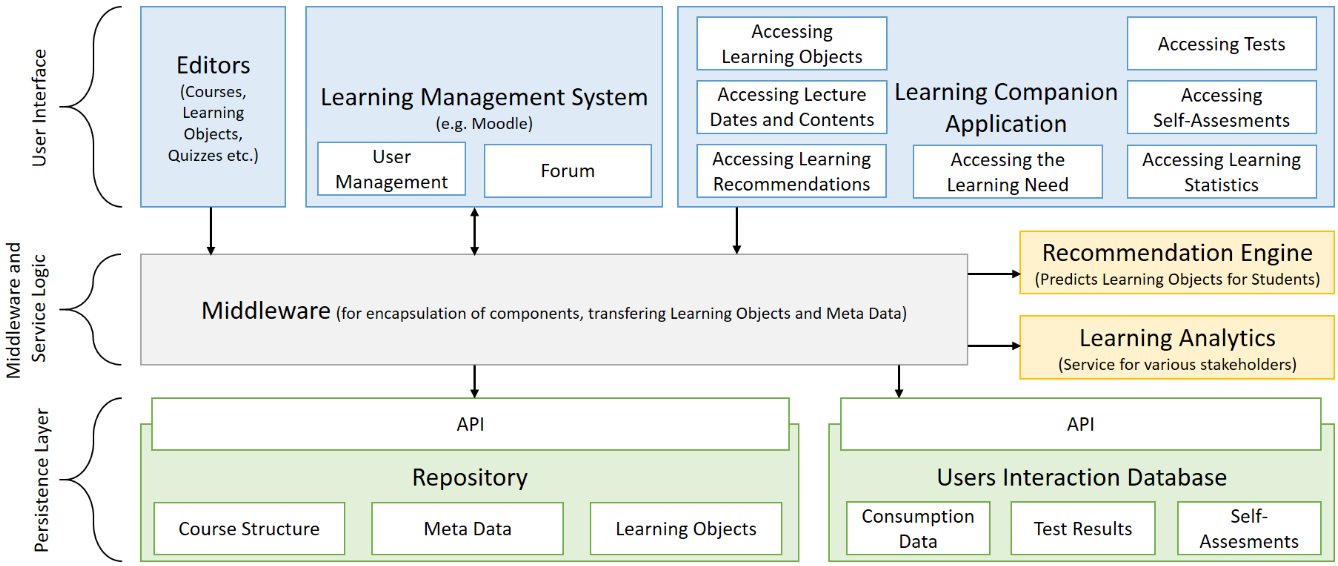

Figure 1. Architecture of the Smart Learning Infrastructure

each interval, rating data is analyzed to find users with similar interests. The predictions are computed using ratings and weights provided for each interval. This approach improves the prediction accuracy for time based collaborative filtering. The recommender system designed by Pooyan Adibi [19] tries to find each user's interest in different groups of items and computes the predictions of rating the user will give in near future. It was observed that the designed system has lower prediction errors when considering the rating time – not only in normal scenario, but also for cold start users. These studies show that the inclusion of forgetting in recommender systems improves the prediction accuracy. Zhang et al. [20] describe an approach to consider changes in users' interests when recommending the items for the users. A novel K-nearest neighbor algorithm [21] finds time-based neighborhoods of a user. Even though these movie recommenders inspire the development of our Smart Learning Recommender, they mostly analyze and predict the users' interests in items instead of considering the users' knowledge [22]. Learning environments, in turn, represent a significantly different prediction paradigm: students have to learn all relevant objects to pass the final exam, no matter whether they are interested in it or not.

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 continuously tracks the learning behavior of individual users over the whole course period, forecasts their learning need on specific LOs at every point in time and recommends appropriate learning objects.

D. Dashboards and Graphical User Representations

Santos [23] proposes "a graphical representation that will help to compare the recommenders' performance in eLearning scenarios". While this was originally designed for life-long learning, we reused this idea to give the student insights in his learning progress within a course and to visualize different context factors in order to show the current status of his learning need on a specific content.

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 [24] [25] [26]. The visual summary proposed by Elkina, Fortenbacher and Merceron [25] 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 of learning progress in the class; the grey portion of the bar shows students with too little activity to enable a classification of their performance. As argued by Martinez-Maldonado et al. [27], 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 [25], dashboards are being developed that adapt elements of the overview proposed 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 in a repository and can be shared by and accessed from various learning management systems.

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

Figure 2. Learning Companion Application: Presentation of the course structure on a smartphone (left picture) and the learning need overview on a tablet (right picture)

The Learning Companion App plays a key role for the infrastructure. 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 tracks all relevant user interactions. It is a responsive web application capable of being 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 offered in the taken courses. Presently, it recommends the top-five most relevant learning objects for the current situation, right after the login process. However, students can access the whole course in a chronological or didactic order, or according to their own personalized setting. Students can optimize their schedule by filtering the important items that fit in the available time period – for instance when waiting for the bus or going to class. Each learning object item is represented by a tile and shows, besides its title, its predicted relevance for the student on a 0 - 100% scale, so that a student can identify and compare the importance of different items at a glance. Different buttons allow access to the content itself, sub-items, exercises, discussion forums or a more detailed version of the learning need representation and the predicted relevance of that item. Figure 2 shows two screens: the course structure on a smartphone and the detailed learning need representation on a tablet. Teachers, in turn, use LCA to get access to the Learning Analytics module that gives an overview of students' progress in the course.

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 middleware.

The repository acts as a digital asset store, which holds course structures, learning objects and their metadata. At the lowest level, a learning object is a simple HTML document, a video, a screencast, a progress evaluation quiz and so on, all with at least one learning objective. Low-level LOs are stored as IMS Learning Tools Interoperability (LTI) [28] to integrate them with different LMSs. Moreover, questions and tests are specified according to the IMS Question and Test Interoperability (QTI) specification [29]. 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 set of sections make up the course. That way, low level LOs can be reused in several courses. A 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-assessment questions using the learning objectives of a learning unit. The metadata associated with a low level LO contains, among others, its learning objectives, average study time defined by instructors and prerequisite LOs. These data are stored using the IMS Learning Resource Meta-data (LOM) specification [30]. When LOs are combined, the metadata of the whole is generated automatically from the parts. A course structure is stored following the IMS Common Cartridge standard [31].

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; presently seven types of questions are available: single selection, multiple selection, extended text, text entry, numeric, matrix and order. Finally, a LO-Editor allows bundling LOs into bigger ones and generating the metadata file automatically. Users interactions with any LO are stored according to the opt-in procedure chosen by the user. Interactions are persisted using

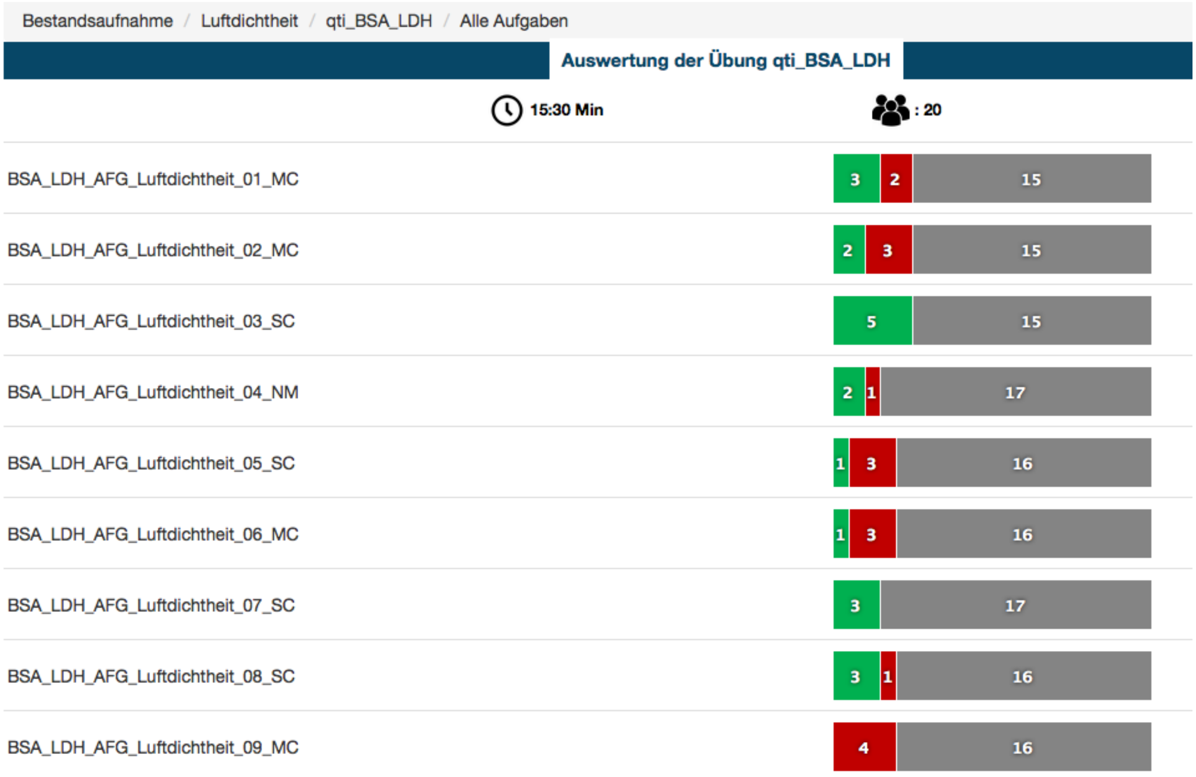

Figure 3. Learning Analytics Module

the xAPI specification [32] in the free learning record store called learning locker.

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 (SLR) aims at identifying the 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. Teachers can observe the overall progress and performance of students and figure out weaknesses in learning and understanding.

IV. Personal Tracking via 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 [32] with its xAPI statements supporting long-term data mining continuously moves in the academic focus. 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 is success with 2 attempts based on the raw score of 80 with a max score of 100 and a scale of 0.8"
- "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 LCA triggers xAPI statements.

A. Learning Analytics as Feedback for the Teachers

As mentioned in the introduction, the main dashboard should allow instructors to get an overview of learners' progress and receive details on demand [33]. It offers details on the selected learning object. We follow the approach of the LeMo-Tool [25] in associating a bar to each object with the colors green, yellow and red, see the work of An et al. [34] for more detailed explanations. 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. Following a similar scheme, details on each question show the number of correct answers in green, partially correct in yellow, wrong answers in red and missing answers in grey; see Figure 3. Further details according to the question's type can be obtained. Filters allow to choose particular time periods for the dashboard.

B. Recommendations for Students

Recommendation engines in a closed domain, like in intelligent learning, are following a special paradigm: At the end of a course, where only a closed user group interacts with a finite amount of items, in this case learning objects, most students provide feedback on almost all items. However, common prediction techniques do not cover that the user's need for learning particular items changes significantly over time – inversely proportional to the user's knowledge level on these learning objects. In this work a time-dependent context-sensitive representation of a user-model is introduced. This helps users to be aware of their learning level and get appropriate learning recommendations as well as teachers to get a direct feedback on the learning behavior. In our Smart Learning Recommender (SLR) – first introduced in [35] – we relate each student with each available learning object in the taken course and aggregate all xAPI statements with time information. We do that to predict the user's knowledge level on all content items and, in turn, recommend relevant learning objects in order to compensate predicted weaknesses. The remainder of the paper focuses on this novel recommender approach.

V. Learning Need and Relevance Function

Learning recommendation is all about identifying the learning need of a user u for an item i at a specific time t. The user-item-pair is presented by a relevance score $rscore_{u,i}$ having the value from 0 to 1, where 0 indicates the lowest relevance and 1 indicates the highest possible relevance. The relevance score defines a time and context dependent value and is expressed as a time dependent function:

$$rscore_{u,i,t} = rf_{u,i}(t) \quad (1)$$

The relevance function $rf_{u,i}(t)$ of user u for item i is derived from several sub-functions $rf_{u,i,x}(t)$ of individual factors $x_1, ..., x_n$, as a function of time t, each representing another context.

The factor itself is based on real user-item value pairs $rscore_{u,i,t,x}$. Since the real learning need changes continuously over time, the factor can be abstracted as continuous function, as well. We identified the following different factor types and considered formulas per user, item and time. Figures 4 to 9 show the spectrum of relevance scores as functions of the main factor parameters.

1) Interaction with a learning object: This factor indicates how much of the available material $availableContent_i$ for a learning object i was accessed by a student u at a specific moment in time t:

$$rscore_{u,i,t,x1} = 1 - \frac{accessedContent_{u,i,t}}{availableContent_i} \quad (2)$$

This can be the percentage of a watched video or audio item as well as how much the student scrolled through a text. We cannot guarantee that student really studied that content, but is our first indication for predicting the knowledge level.

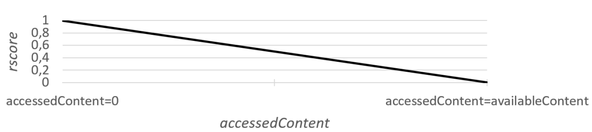

Figure 4. Relevance values for interactions

2) Processing time of a learning object: This factor indicates how long the student learned a learning object. It is 0 when the student needed exactly the intended time and between 0 and 1 if he needs more or less time than defined in the metadata.

$$rscore_{u,i,t,x2} = \left|1 - \sqrt{\frac{timeNeededForLearning_{u,i,t}}{timeIntendedForLearning_i}}\right| \quad (3)$$

In an initial phase, we work with an upper bound for the time that is needed for learning: The square root lessens the effect when a student did not exactly learn the intended time. If the user needed more than 4 times of the time, the learning need is 1. In combination with the percentage of interaction, the processing time allows a good approximation whether a student really worked through that content.

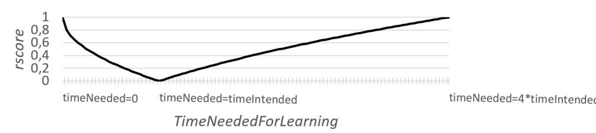

Figure 5. Relevance values for processing time

3) Self-assessments for this learning object: A student can explicitly define his knowledge level in particular points in time on a 1 to 5 stars scale:

$$rscore_{u,i,t,x3} = 1 - \frac{currentKnowledgeLevel_{u,i,t}}{highestKnowledgeLevel} \quad (4)$$

We ask for this feedback in various situations, even when students have just read the title of a learning object. They can adjust the self-assessment at any time.

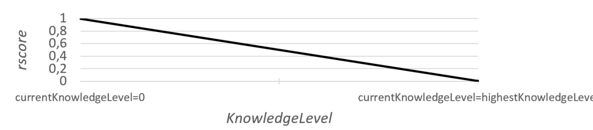

Figure 6. Relevance values for self-assessments

4) Performance in exercises: The percentage of wrong answered questions represents the relevance of the exercise factor:

$$rscore_{u,i,t,x4} = 1 - \frac{rightAnswers_{u,i,t}}{allAnswers_i} \quad (5)$$

Equation (5) is the same as $\frac{wrongAnswers_{u,i,t}}{allAnswers_i}$. Left out answers will be treated as wrong answers, so that the relevance score is only 0, when a student correctly answered all questions.

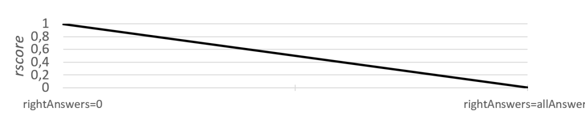

Figure 7. Relevance values for exercises

5) Fulfilled pre-requisites: The more a student learned the underlying learning objects, the higher the relevance score of the subsequent items:

$$rscore_{u,i,t,x5} = \frac{fulfilledPreRequisites_{u,i,t}}{allPreRequisites_i} \quad (6)$$

This factor is not directly affected by the users' interaction. It depends on the learning of prerequisite items. Thus, a learning object gets more relevant when a users understood the required basics. In case there is no pre-requisite for a learning object, the relevance score is 1.

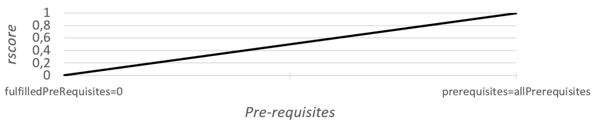

Figure 8. Relevance values for fulfilled prerequisites

6) The lecture times factor indicates the timely relevance of a learning object for face-to-face lectures. The closer the lecture at time $timeOfLecture$, the higher the relevance score:

$$rscore_{u,i,t,x6} = 1 - \left(prepPhaseFactor * \frac{timeOfLecture_i - t}{courseDuration} \right)^2 \quad (7)$$

The $courseDuration$ (time of course ending minus time of course start) must be higher then 0 and the $timeOfLecture$ must be within the $courseDuration$. The $prepPhaseFactor$ needs to be defined by the teacher and defines the duration of both: the preparation as well as the wrap up phase of a lecture, where the contents concerned are more relevant. For instance, for $prepPhaseFactor = 1$ the preparation phase is exactly the course duration, for $prepPhaseFactor = 2$ it is half the course duration, for $prepPhaseFactor = 4$ the preparation phase is one quarter of the course duration and so on. The higher the number, the later the begin of the preparation phase and, thus, the later the recommendation of this learning object. In case the current time t does not fall in the preparation or wrap-up phase, the relevance score is set to 0.

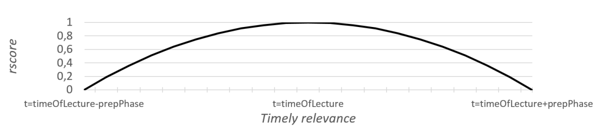

Figure 9. Relevance values for lecture times

7) Exam relevance: Learning objects that are more relevant for exams show a higher relevance score than optional contents – expressed by a constant value defined in the learning object metadata.
8) Collaborative learning needs: The relevance functions of similar users on this learning object are taken into account in order to offset underestimations and bad learning plannings for the current user. In an initial phase, the mean average learning need of all other students (but without their collaborative learning need to avoid recursion loops) for this learning object represent this factor. The plan is to replace this mean learning need by a weighted average factor of nearest-neighbors, who show the most similar learning curves.
9) Forgetting effect: After learning an object, the gained knowledge will decrease over time. This factor has been analyzed with real students in order to model an appropriate forgetting factor. Details on this factor are described in the following section.

Each factor's relevance score represents an aspect of the learning need. It is restricted to the range of $[0, 1]$ and will be further evaluated and adjusted in the future. At the end, all single-factor functions are weighted. The weighted average of all factors describes the total learning need of the learning object for that user and is calculated as

$$rf_{u,i}(t) = \frac{\sum_{x=1}^{n}(w_x * rf_{u,i,x}(t))}{\sum_{x=1}^{n} w_x} \quad (8)$$

Here w_x is the weight of a single factor x in $\{x_1, ..., x_n\}$ and n is the number of factors – currently, $n = 9$. At the beginning, the weights are predefined by experts, such as teachers.

VI. Forgetting Factor

The study on human forgetting began in early 19th century. Hermann Ebbinghaus [36] gave the first and still representative equation for the forgetting curve. He noticed that forgetting is high during the initial period after learning and gradually decreases over time. An experiment was conducted by Harry P. Bahrick [37] to test the recall and recognition of 50 English-Spanish word pairs over a period of 8 years. The results showed that the recall and recognition percentage of words is greater in larger intersession intervals, indicating the influence of spacing on forgetting. In recent years, some recommender systems re-used the equation from Ebbinghaus to improve predictions for the e-commerce and entertainment domain [38] [39].

Some research highlights positive aspects of forgetting, especially forgetting in the area of big data. For instance, "forgetting" or "trashing" might be a necessary instrument when storing or processing information of huge data sets in order to handle less data in total and improve the overall performance [40]. However, this section focuses on human forgetting of learning contents that must be avoided to pass the final exam.

A. Parameters of Forgetting

The decay theory [41] states that a person's memory of a learned content fades away over time, when it is not used. That is why forgetting represents a special relevance factor function $rf_{u,i,forgetting}(t)$ in the Smart Learning Recommender. Apart from time, we identified the following parameters that influence forgetting:

- Media type: The type of learning objects plays a vital role in remembering (cf. [3] [42]). In our approach, the media types are text, exercise, audio, graphics/image, video and multimedia.
- Difficulty level: The learning content represents different difficulty levels [42]. A common way of categorizing difficulty level is easy, medium and hard. It can be set based on the amount of content, detail level of information and so on. Forgetting will increase from difficulty level easy to hard.
- Prior knowledge: Learner's prior knowledge about the course helps to easily understand the course as

compared to a learner who is new to the course. Hence, with prior knowledge, the learner remembers more [3] [42].

- Learner's interest towards the content: If the learner is not interested in a subject, forgetting tends to be at a higher rate compared to the subject of interest [42].
- Learner's memory strength: Every person in the world is different from each other, so their memory. A learner with higher memory strength can remember more, compared to a learner with lower memory strength.
- Repetition: During repetitions, students learn the offered items again. Repetition is very helpful in strengthening the memory of learned content [36] [43].
- Repetition spacing: Constant repetitions at regular intervals will help to retain learned content. However, when the time spacing between initial learning and repetition is large, the percentage of increase in memory of the content is also high compared to the percentage increase with short repetition interval [43].
- Re-remembrance due to retention tests: Most research does not take the effect of retention tests for the re-remembrance into account: When people are asked about a topic, the questions in the retention test can act as a retrieval cue and allow the learner to correlate the words in the question to the previously learned content – thereby, remembering the forgotten concept as a side-effect.
- Retention test spacing: Similar to the repetition spacing, the gap in time between two successive retention tests can also have an influence on the re-remembrance. The more spacing, the smaller the effect of re-remembrance.

We conducted an experiment with a group of eight people to confirm and study the effect of the parameters that could influence forgetting. The duration was eight to ten weeks and involved people from different fields like information technology, medicine and business administration. The eight people learn a learning object at the beginning of the experiment. In regular intervals, we performed retention tests to observe the learner's knowledge. 5-8 questions from a group of 10-12 questions that represent the key information of the given topic are randomly picked and posed at the person to test the percentage of forgetting. The assumption: the progress of wrongly answered questions (in percent) over time represent the progress of forgetting. In the experiment, the parameter values are varied, to get a clue about its effect on forgetting. We presented different media types, in terms of texts and videos with different complexity levels. In some cases, the learning objects needed to be learned again at specific points in time to evaluate the repetition effect. Figure 10 shows an excerpt from the survey results with the forgetting progress of videos that have been watched only once and show different lengths and complexity levels.

B. Mathematical Description

Based on the experiment, a mathematical model for forgetting is derived, given by Equation (9). We analyzed the progress of forgetting during the study (e.g., in Figure 10) and searched for a mathematical abstraction that represents the forgetting best:

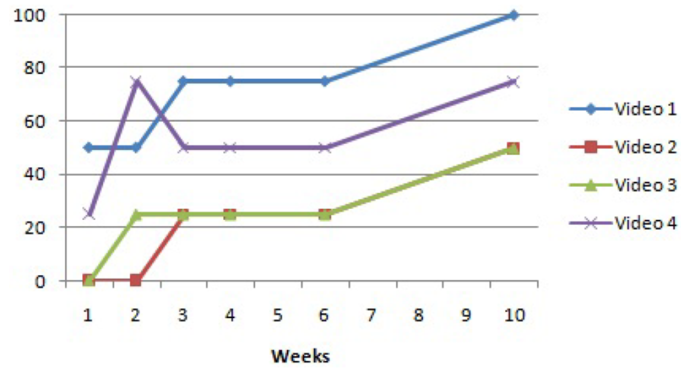

Figure 10. Survey results for forgetting of videos with different complexity levels that have only been watched at the beginning of the experiment. The y-axis represent the percentage of wrong or not answered questions per retention test. Thereby each 12,5% stand for one of eight answered questions.

$$rf_{u,i,forgetting}(t) = factorEffect(t) + \alpha \tag{9}$$

The model includes two main parameters: the $factorEffect$ with respect to learning times and media metadata as well as a personalized parameter α that customizes the forgetting equation for the learner and thus, represents the individual forgetting process. The value of forgetting is restricted between 0 and 1 as well as all the other factors. The $factorEffect(t)$ corresponds to the following exponential function:

$$factorEffect(t) = 1 - e^{-(timeFactor(t) - E_{rep} - E_{ret} + E_m + E_d)} \tag{10}$$

Thereby, the e-function shows the most similar progress compared to the participants forgetting in the survey and a set of parameters represent the exponent. Where E_{rep} is the repetition effect, E_{ret} is the retention test effect, E_m is the media type effect and E_d represents the difficulty level. If the learner does not repeat to learn a learning object or does not answer retention tests, these parameters are set to 0.

The values for E_m are in the range $[0, 0.1]$. For example, media types which increases the retention by the highest possible value would be assigned 0. In our case, videos and animations are assigned 0.04 and text with 0.1. The difficulty level E_d can vary between 0 and 0.1, as well, where easy is assigned 0 and hard with 0.1. The $timeFactor$ represents the forgetting – only with respect to time:

$$timeFactor(t) = \frac{t - T_f}{T_c} * \sqrt{\frac{T_c}{T_s}} \tag{11}$$

Where $t - T_f$ corresponds to the difference between current time t and the first access time of the learning content T_f in days. T_c is the total course duration. T_s represents the speed of forgetting (if no other parameter affects forgetting), that is set in an initial version to 30 days for a course duration T_c of 180 days, and can be adjusted by the teacher if needed. As forgetting starts as soon as one starts learning an object, the relevance score is set to 0 before the first access of the content.

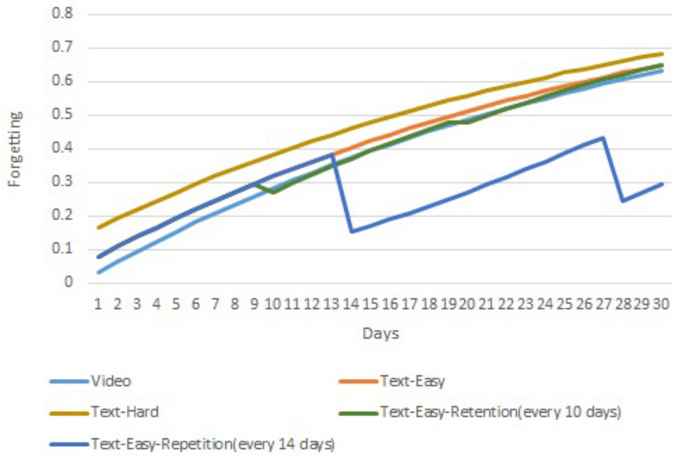

Figure 11. Mathematical model for predicted forgetting of different media types and with retention tests and repitions on a 0 to 1 scale.

It can be observed from the Formula (11) that there will be an increase in the value of $timeFactor(t)$ with longer course durations.

The equations for the effect of repetition and re-remembrance due to retention tests are given by Equations (12) and (13).

$$E_{ret} = n_{ret} * \frac{T_c - T_{d,ret}}{T_c} * K_{ret} \tag{12}$$

Where T_c is the total course duration expressed in days and $T_{d,ret}$ is the number of days after the last retention test. n_{ret} is the retention test count. The constant K_{ret} at the end of the equation indicates the weight by which the forgetting is reduced. The ideal value for the constant is 0.01, based on the observations from the experiment.

$$E_{rep} = timeFactor * \frac{T_c - T_{d,rep}}{T_c} * K_{rep} \tag{13}$$

E_{rep} includes the *timeFactor* that was initially given by Formula (11), $T_{d,rep}$ is the number of days elapsed after last repetition. The constant K_{rep} at the end of the equation indicates the weight by which the forgetting is reduced. In our experiment the ideal value for this constant is 0.75.

Factors like memory strength and the learner's interest towards the content are specific to each learner. These factors make the forgetting curve unique for each learner, but need further evaluations. The adapting constant α requires the conduction of retention tests. If there is no retention test planned, the adapting constant is set to 0. The value α personalizes the forgetting curve by taking the learner's performance in the retention into account. It is given by the following formula:

$$\alpha = \frac{predictedScore - actualScore}{2} \tag{14}$$

It represents the deviation between the regular forgetting curve – given by the factor effect in Formula (10) – and the real forgetting progress of a single person – determined by retention tests. Figure 11 shows the curves for forgetting under different settings of the factors, but without the inclusion of personalization parameter.

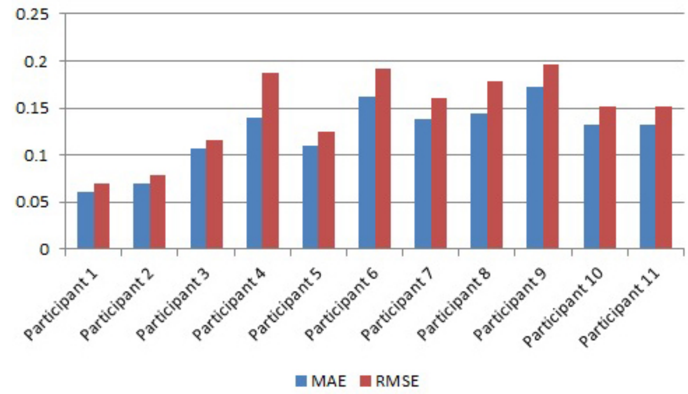

Figure 12. Error values for different participants

C. Preliminary Evaluation of the Forgetting Effect

Since this work shows a novel approach on forgetting and only a few data sets were published (e.g., [44]), which do not match all requirements of this approach, we generated our own data. We conducted a similar experiment as at the beginning in order to evaluate the correctness of our thesis. In our evaluation, 11 participants learned a previously unknown learning object just once. Over a period of eight to ten weeks, they had to answer retention tests in regular intervals. Each questionnaire consisted of four to eight questions and every single question was just asked once. The evaluation resulted in the analysis of the prediction accuracy using Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). See Figure 12 for error values of different participants.

The results are compared with those from the equation of Hermann Ebbinghaus [36]. Table I shows the average values of MAE and RMSE for the results from this study and from the Ebbinghaus equation for forgetting. It can be noticed that the average MAE and average RMSE are nearly 3 times lower in the model developed in the current study compared to the existing model from Ebbinghaus. This indicates a higher prediction accuracy of the model developed for forgetting in the current study, but still requires further experiments. We incorporate the forgetting factor as well as the other factors for a learning object and thus, predict its overall learning need for the given user.

VII. Multi-level Learning Recommendations

After aggregating the factor values for each single learning object, the overall recommendation engine compares the learning need values of the requesting student for all contents in the course. The model of the SLR can be created offline: The relevance functions are computed in regular intervals by processing all existing user-item-time-triplets. When a user requests recommendations, the relevance scores for all items, in turn, are calculated online by considering the current time value for t. Afterwards, the items are sorted by their learning

TABLE I. Error values for forgetting models

Equation	MAE	RMSE
by Hermann Ebbinghaus	0.3518	0.3896
SLR Forgetting	0.1245	0.1461

need value. The result is a list of learning objects beginning with the highest relevance scores that represents the most important topics for the student that need to be learned at that time.

Figure 13 shows how the learning need of an item is presented graphically to a learner. It shows an example of a learning object in a 21 weeks long course and with a lecture on this topic in week 6; the student learned the content in weeks 4, 6, 7, 15, 18 and 19. The personal knowledge level defines how successful a student is in learning an item and is inversely proportional to the learning need of the student towards the learning object. The knowledge level $kl(t)$ can be computed from the relevance function $rf(t)$ as

$$kl(t) = 1 - rf(t) \tag{15}$$

Using this visualization, students have a chance to understand how the system calculates the prediction and might change factor weights to adjust new recommendations.

Moreover, learning objects are stored in a multi-level hierarchy to topically structure a course – top-level items represent a container with a set of sub-level items. A leaf item (a learning object without children) contains the minimum information set that is at least required to understand the given (sub-)topic. The user may provide item feedback, in terms of self-assessments, exercises, interactions and processing time, on all hierarchy levels. This differentiation allows a representation of diverging knowledge levels for top- and sub-level items – e.g., a student might have a good high-level understanding of a topic, but misses some details on specific sub-level contents or vice versa. In case the user has not provided the same type of feedback on the item's parent before, it is implicitly transferred from the child to the parent object. So, the parent may implicitly represent the average of all child learning objects.

The engine needs to avoid recommending the same topic with different detail levels within the predicted list. An algorithm iterates over the generated Top-N list, beginning with the most relevant learning object. An item will be eliminated from the list, in case a related child or parent learning object that describes the same topic shows a higher score. As a result, students will get recommendations for all topics of a course in a predicted order, but only on an appropriate detail level.

VIII. Challenges and Experimental Design

The introduced three-dimensional user-item-matrix (containing user-item-time-triplets) also leverages common collaborative filtering approaches. The calculation and weighting of nearest neighbors will be done by also considering the time aspect. Therefore, the deviation of two user-item-pairs will not only be based on the subtraction of two constants any more – as for common collaborative filtering approaches. It will be based on the correlation of the corresponding relevance functions of two learners. The assumption: the higher the correlation coefficient of two learning need functions of different students on the same item, the more similar their knowledge and their learning behavior. If one learning need function decreases from 1 to 0 over time, just because the student has learned this content perfectly, and the learning need function of another user shows the same progress, the correlation coefficient is high. If, in contrast, one function goes down and the other one goes up, the coefficient will indicate an anti-proportional progress and is very low. The Pearson Correlation Coefficient, for instance, requires a linear trend function. Thus, the main progress, in terms of the starting point and the end point values can be used for a reduced linear learning need function. Taking the correlation information into account, the system can identify similar learners, because of similar learning trends and similar knowledge levels. Moreover, it can be used to classify and cluster general learning types (e.g., slow or fast ones), all with similar learning trends. The ideal composition of different learning types in one learning group will then need intensive calibration. Algorithms covering time-dependent item-based as well as neighborhood-based classification and rating prediction are going to be evaluated. Another big challenge of this approach comes from predictions of the future learning need by extrapolating specific factor functions, for instance the forgetting effect. In the planned experiments, different algorithms, weights and settings are going to be further analyzed.

Since this work shows a novel approach for time-dependent learning object recommendations, the need for an evaluation based on an academic data set is very high. Unfortunately, only a few data sets are published (e.g., [45] or [46]) and no data set matches all requirements of this approach. We need the information for at least one factor function (as introduced in Section V), such as learning object interactions or performance in exercises, in order to conduct basic experiments. It is essential that there is a set of learning objects not only learned by different students, but also by the same student at multiple points in time. The change of the relevance of a learning object for a specific user over time represents the key aspect of the approach. Moreover, we need detailed feedback for each user interaction as well as on the learning object itself. At least, the challenge data set from KDD Cup 2010 on Educational Data Mining [44] matches some requirements. It is divided into 5 different packages (e.g., "Algebra I" and "Bridge to Algebra" from 2005 and 2008) with between 575 and 6,043 students per package. It contains a detailed description of the students' performances when solving mathematical problems and thus, represents typical learning behavior. One evaluation approach would be to subdivide the KDD item data into different context factors – each influencing the total learning need. However, the KDD data set contains a lot information on the interaction with learning objects as well as the processing time and results in exercises, but data on other essential factors as well as structured metadata on the hierarchy and topical sequences of learning objects are missing.

That is why new studies with learners are going to be conducted at the Chamber of Crafts Berlin in two consecutive 5 month courses – each with the same newly generated learning objects and metadata, but different students in each course. At the beginning of each course, the participants are asked to answer surveys with demographic information and their motivation. We require this information to set their interaction with our system in a context and afterwards draw conclusions (such as "digital natives enjoyed the system", "technical beginners did not understand it" or the like) and adapt the system. During the course, participants will get access to the learning objects exclusively via a provided Learning Companion Application to keep track of their learning behavior. Moreover, they can give feedback at any time, how helpful a specific recommendation was and whether the learning need shows a proper presentation of their real knowledge. This data is important for accuracy

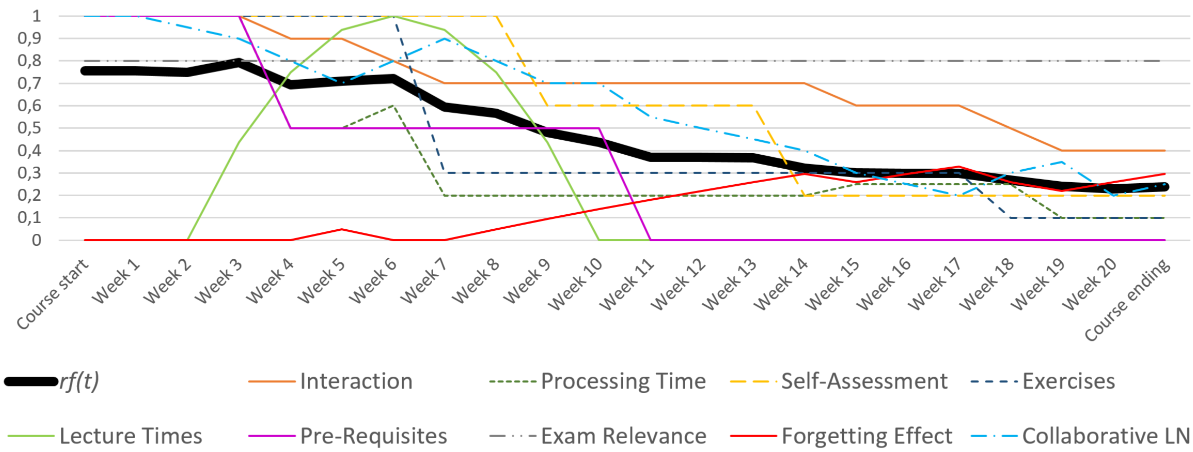

Figure 13. Example of a learning need function with individual factors

measurements as well as the analysis and adjustment of factor weights. At the end, the participants are asked to answer a second survey summarizing their overall perception of the system. We will relate the final exam grade with the tracked learning behavior data, but due to local data protection regulations, it is optional only.

Moreover, we are adapting this system to additional studies: at the Technical University of Berlin (a programming-oriented course called "Advanced Web Technologies") as well as an institute-internal summer school with the same contents as for the Technical University. The first one will have approximately 40 student participants with a mandatory exam at the end. As the target groups are different for all three experiments, we hope to get more generic feedback, that allows us to optimize the system for a broad range of future participants. The studies help to increase the performance of the system with each experimental iteration. We plan to publish the mined information as an anonymized open source data set.

IX. Conclusion

We introduced a novel Learning Companion App, which allows learners to access standardized learning objects everywhere and at any time. 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.

The Smart Learning Recommender aims at assisting students during blended-learning courses; in lectures; during the preparation of these lectures, the wrap-up and exam learning phase. Thereby, the engine shows an extended user model: the item feedback of each user is subdivided into different context factors. In contrast to rule-based recommendation engines and classification machine learning algorithms, it also respects the changing knowledge level on specific learning objects in a continuous time interval. Moreover, the system respects the overall course structure, in terms of the best topical sequence and thematic hierarchies consisting of topics and sub-topics. Due to the lack of an appropriate academic data set, studies with real learners will evaluate typical learning behaviors, how the SLR performs with different settings and how users accept learning recommendations. An analysis of the students' knowledge level at several points in time will result in an accurate representation of the different factors and their weights.

The inclusion of human forgetting in recommending learning items is a whole new approach in the field of TEL. The equation developed for forgetting in this study, serves as a preliminary model for computing the forgetting curve for a learner. The results from the assessment of the model have shown an improvement of the prediction accuracy. But there is still a lot of work for improving this model and the results are based on an initial experiment, with a small group of people. There is a need to evaluate this model on a large scale. In addition, other parameters like size and structure of the content, effect of successive repetitions, meaningfulness of the content, other media types and inter-dependencies of different parameters will be further studied.

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 September 2016 and the second in March 2017. Moreover, a University course as well as a summer school is planned. These studies evaluate real world learning behavior: how LCA performs and how users accept digital learning media and individual learning recommendations. 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. Special thanks go to Christian Fuhrhop, who proofread this paper and made a lot of practical suggestions.

The Smart Learning project is sponsored by the German Federal Ministry of Education and Research (Bundesministerium fuer Bildung und Forschung – BMBF) under the project funding number 01PD14002A.

REFERENCES

[1] C. Krauss, A. Merceron, T.-S. An, M. Zwicklbauer, and S. Arbanowski, "The smart learning approach - a mobile learning companion application," In Proceedings of The Eighth International IARIA Conference on Mobile, Hybrid, and On-line Learning (eLmL 2016), Venice, Italy, Volume: 8, April 2016, pp. 13–16.

[2] M. Zwicklbauer, C. Krauss, M. Merceron, Scharp, and J. Kania, "Smart learning: the digital learning companion for vocational training [org. title: Smart learning: Der digitale lernbegleiter fr die berufliche bildung]," Proceedings of DeLFI 2015, Gesellschaft fr Informatik Publisher, Munich, 2015, pp. 227–232.

[3] O. C. Santos and J. G. Boticario, "Recommendation strategies for promoting elearning performance factors for all," Proceedings of the 6th Workshop on Intelligent Techniques for Web Personalization & Recommender Systems, 2008, pp. 89–98.

[4] 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.

[5] G. Koutrika, B. Bercovitz, F. Kaliszan, H. Liou, and H. Garcia-Molina, "Courserank: A closed-community social system through the magnifying glass." in ICWSM, 2009.

[6] M. M. Recker and D. A. Wiley, "A non-authoritative educational metadata ontology for filtering and recommending learning objects," Interactive learning environments, vol. 9, no. 3, 2001, pp. 255–271.

[7] H. Boley, "Racofi: A rule-applying collaborative filtering system. in 2003 ieee," in WIC International Conference on Web Intelligence/Intelligent Agent Technology, 2003, pp. 430–434.

[8] H. T. Ozturk, D. Deryakulu, H. Ozcinar, and D. Atal, "Advancing learning analytics in online learning environments through the method of sequential analysis," in Multimedia Computing and Systems (ICMCS), 2014 International Conference on. IEEE, 2014, pp. 512–516.

[9] Á. F. Agudo-Peregrina, Á. Hernández-García, and S. Iglesias-Pradas, "Predicting academic performance with learning analytics in virtual learning environments: A comparative study of three interaction classifications," in Computers in Education (SIIE), 2012 International Symposium on. IEEE, 2012, pp. 1–6.

[10] H. Stern, R. Kaiser, P. Hofmair, P. Kraker, S. N. Lindstaedt, and P. Scheir, "Content recommendation in aposdle using the associative network." J. UCS, vol. 16, no. 16, 2010, pp. 2214–2231.

[11] N. Manouselis, R. Vuorikari, and F. Van Assche, "Simulated analysis of maut collaborative filtering for learning object recommendation," in Proceedings of the 1st Workshop on Social Information Retrieval for Technology Enhanced Learning, 2007, pp. 27–35.

[12] Y.-M. Huang, T.-C. Huang, K.-T. Wang, and W.-Y. Hwang, "A markov-based recommendation model for exploring the transfer of learning on the web," Journal of Educational Technology & Society, vol. 12, no. 2, 2009, pp. 144–162.

[13] L.-p. Shen and R.-m. Shen, "Learning content recommendation service based-on simple sequencing specification," in Advances in Web-Based Learning–ICWL 2004. Springer, 2004, pp. 363–370.

[14] J. Janssen, C. Tattersall, W. Waterink, B. Van den Berg, R. Van Es, C. Bolman, and R. Koper, "Self-organising navigational support in lifelong learning: how predecessors can lead the way," Computers & Education, vol. 49, no. 3, 2007, pp. 781–793.

[15] H. Drachsler, H. Hummel, B. Van den Berg, J. Eshuis, W. Waterink, R. Nadolski, A. Berlanga, N. Boers, and R. Koper, "Effects of the isis recommender system for navigation support in self-organised learning networks," Journal of Educational Technology & Society, vol. 12, no. 3, 2009, pp. 115–126.

[16] R. Pelanek and P. Jarusek, "Student modeling based on problem solving times," International Journal of Artificial Intelligence in Education, vol. 25, no. 4, 2015, pp. 493–519. [Online]. Available: http://dx.doi.org/10.1007/s40593-015-0048-x

[17] L. Xiang and Q. Yang, "Time-dependent models in collaborative filtering based recommender system," WI-IAT '09., vol. 1, 2009, pp. 450–457.

[18] L. He and F. Wu, "A time-context-based collaborative filtering algorithm," in Granular Computing, 2009, GRC'09. IEEE International Conference on. IEEE, 2009, pp. 209–213.

[19] P. Adibi and B. T. Ladani, "A collaborative filtering recommender system based on user's time pattern activity," 2013 5th IKT Conference, 2013, pp. 252–257.

[20] Y. Zhang and Y. Liu, "A collaborative filtering algorithm based on time period partition," in Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium on. IEEE, 2010, pp. 777–780.

[21] Y. Liu, Z. Xu, B. Shi, and B. Zhang, "Time-based k-nearest neighbor collaborative filtering," in Computer and Information Technology (CIT), 2012 IEEE 12th International Conference on. IEEE, 2012, pp. 1061–1065.

[22] C. Krauss, L. George, and S. Arbanowski, "TV predictor: personalized program recommendations to be displayed on smarttvs," in BigMine'13. ACM, 2013, pp. 63–70.

[23] O. C. Santos, "A recommender system to provide adaptive and inclusive standard-based support along the elearning life cycle," in Proceedings of the 2008 ACM conference on Recommender systems. ACM, 2008, pp. 319–322.

[24] K. Verbert, S. Govaerts, E. Duval, J. L. Santos, F. Van Assche, G. Parra, and J. Klerkx, "Learning dashboards: an overview and future research opportunities," Personal and Ubiquitous Computing, vol. 18, no. 6, 2014, pp. 1499–1514.

[25] M. Elkina, A. Fortenbacher, and A. Merceron, "The learning analytics application lemo–rationals and first results," International Journal of Computing, vol. 12, no. 3, 2013, pp. 226–234.

[26] A. Dollár and P. S. Steif, "Web-based statics course with learning dashboard for instructors," Proceedings of computers and advanced technology in education (CATE 2012), 2012.

[27] 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," in Proceedings of the Fifth International Conference on Learning Analytics And Knowledge. ACM, 2015, pp. 1–10.

[28] IMS-LTI - Learning Tools Interoperability Specification, IMS Global Learning Consortium, 2015 Std.

[29] IMS QTI - IMS Question & Test Interoperability Specification, IMS Global Learning Consortium, 2005 Std.

[30] IMS LOM - Learning Resource Meta-data Specification, IMS Global Learning Consortium, 2006 Std.

[31] IMS CC - IMS Common Cartridge Specification, IMS Global Learning Consortium, 2015 Std.

[32] J. M. Kevan and P. R. Ryan, "Experience api: Flexible, decentralized and activity-centric data collection," Technology, Knowledge and Learning, vol. 21, no. 1, 2016, pp. 143–149.

[33] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in Visual Languages, 1996. Proceedings., IEEE Symposium on. IEEE, 1996, pp. 336–343.

[34] T.-S. An, F. Dubois, E. Manthey, and A. Merceron, "Digitale infrastruktur und learning analytics im co-design," In Proceedings of the Workshop learning Analytics, co-located with the 14th e-Learning Conference of the German Society for Computer Science, Potsdam, Germany, 11th September 2016. To appear., pp. 8–17.

[35] C. Krauss, "Smart learning: Time-dependent context-aware learning object recommendations," in Proceedings of the 29th International Florida AI Research Society Conference (FLAIRS-29), AAAI, Key Largo, 2016, pp. 501 – 503.

[36] H. Ebbinghaus, "Memory: A contribution to experimental psychology, translated by henry a ruger and clara e bussenius," 1913.

[37] H. P. Bahrick and E. Phelps, "Retention of spanish vocabulary over 8 years," Journal of Experimental Psychology: Learning, Memory, and Cognition, vol. 13, 1987, pp. 344–349.

[38] Y. Hong and L. Zhuanyun, "A collaborative filtering recommendation algorithm based on forgetting curve," Journal of Nanjing University, vol. 46, no. 5, 2010, pp. 520–527.

[39] H. Yu and Z. Li, "A collaborative filtering method based on the forgetting curve," in 2010 International WISM Conference, vol. 1. IEEE, 2010, pp. 183–187.

[40] V. Ermolayev, R. Akerkar, V. Terziyan, and M. Cochez, "Towards

evolving knowledge ecosystems for big data understanding," Big data computing, 2013, pp. 3–55.

[41] J. Brown, "Some tests of the decay theory of immediate memory," Quarterly Journal of Experimental Psychology, vol. 10, no. 1, 1958, pp. 12–21.

[42] W. Thalheimer, "How much do people forget?" Work-Learning Research, Inc. Somerville, MA, USA, 2010.

[43] P. I. Pavlik Jr. and J. R. Anderson, "Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect," Cognitive Science, vol. 29, no. 4, 2005, pp. 559–586.

[44] J. Stamper, A. Niculescu-Mizil, S. Ritter, G. Gordon, and K. Koedinger, "Algebra i 2008-2009," Challenge data set from KDD Cup 2010 Educational Data Mining Challenge., 2010.

[45] P. Cortez and A. M. G. Silva, "Using data mining to predict secondary school student performance," Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008), 2008.

[46] R. Wang, F. Chen, Z. Chen, T. Li, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, and A. T. Campbell, "Studentlife: assessing mental health, academic performance and behavioral trends of college students using smartphones," in Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, 2014, pp. 3–14.