Prompting in Pseudonymised Learning Analytics
Implementing Learner Centric Prompts in Legacy Systems with High Privacy Requirements

Daniel Schön[1] and Dirk Ifenthaler[1,2]
[1]*Chair of Economic and Business Education – Learning, Design and Technology, University of Mannheim, L4,1, Mannheim, Germany*
[2]*Curtin University, Bentley, Australia*

Keywords: Learning Analytics, Prompting, Learner Privacy, Learner Centric.

Abstract: Learning analytics have become a well-considered aspect of modern e-learning environments. One opportunity of learning analytics is the use of learning process data enabling lecturers to analyse students' learning progression as well as to identify obstacles and risks. With this analytics knowledge, lecturers may want to scaffold students' learning activities to improve the learning progress and overcome obstacles or risks. Prompts are known to be a possible solution for such scaffolding mechanics. However, implementing prompts into existing legacy systems in learning environments with high data privacy concerns is quite a challenge. This research shows how a prompting application has been implemented into an existing university environment by adding a plugin to the local e-learning platform which injects user centric prompts to specific objects within their e-learning environment. The prompts are dynamically loaded from a separate learning analytics application which also collects the students' learning trails and progress. The system is evaluated in two courses in the fall semester 2017 with more than 400 students altogether. The system collects up to two thousand student events per day. An in-depth empirical investigation on how various prompts influence students' learning behaviours and outcomes is currently conducted.

1 INTRODUCTION

The field of learning analytics (LA) is generating growing interest in data and computer science as well as educational science, hence, becoming an important aspect of modern e-learning environments (Ifenthaler and Widanapathirana, 2014). LA are often discussed and linked with regard to self-regulated learning. One general assumption is that each learning process demands a certain degree of self-regulation (Zimmerman, 2002). How effective a learner can regulate his or her learning depends on cognitive, motivational, volitional, and metacognitive dispositions (Bannert, 2009). Accordingly, self-regulated learning can be seen as a cyclical process including three major phases: (1) Starting with a forethought phase including task analysis, goal setting, planning, and motivational aspects. (2) The actual learning occurs in the performance phase, i.e., focusing, applying task strategies, self-instruction, and self-monitoring. (3) The last phase contains self-reflection, as learners evaluate their outcomes versus their prior set goals. To close the loop, results from the third phase will influence future learning activities (Zimmerman, 2002). Current findings show that self-regulated learning capabilities, especially revision, coherence, concentration, and goal setting are related to students' expected support of LA systems (Gašević et. al., 2015). For example, LA facilitate students through adaptive and personalized recommendations to better plan their learning towards specific goals (McLoughlin and Lee, 2010). Other findings show that many LA systems focus on visualisations and outline descriptive information, such as time spent online, the progress towards the completion of a course, comparisons with other students (Verbert et. al., 2012). Such LA features help in terms of monitoring. However, to plan upcoming learning activities or to adapt current strategies, further recommendations based on dispositions of students, previous learning behaviour, self-assessment results, and learning goals are important (McLoughlin and Lee, 2010). In sum, students may benefit from LA

Schön, D. and Ifenthaler, D.
Prompting in Pseudonymised Learning Analytics.
DOI: 10.5220/0006780903830389
In *Proceedings of the 10th International Conference on Computer Supported Education (CSEDU 2018)*, pages 383-389
ISBN: 978-989-758-291-2

through personalised support of their learning journey.

One of the features with a high impact potential on this personalized support are prompts. Prompts are ad hoc messages, which provide or request individualized information from the students. They can be used to offer hints to the current learning material, to trigger students' self-reflection on their learning process or to request student-specific information. At best, prompts are directly injected into the students' learning environment. Prompts are effective means for supporting self-regulated learning (Bannert, 2009). They are an essential instructional method for aiding certain aspects which are needed for self-regulated learning. Prompts support learners in activating their metacognitive strategies. These strategies make self-regulation, self-monitoring and evaluation possible (Ifenthaler, 2012; Veenman, 1993).

Davis (2003) investigated when (before, during or after the actual learning process) a prompt should be presented to the learner in order to achieve the best learning outcome. Accordingly, prompting depends on what the prompt is aiming at. If the aim is to promote the planning of the learning procedures, a presentation before the learning task is advisable. By contrast, prompting during the learning process is appropriate, when the learner is to be induced to monitor and evaluate learning procedures (Davis, 2003).

However, implementing prompts into existing legacy systems in learning environments with high data privacy concerns is quite a challenge. This research shows how a prompting application has been implemented into such an existing university environment by adding a plugin to the local e-learning platform which injects user centric prompts to specific objects within students' e-learning environment.

In this paper, we describe the concept and implementation of the LeAP (Learning Analytics Profile) application including flexible prompting and present preliminary findings of the data we are able to generate.

2 CONCEPT AND IMPLEMENTATION

2.1 General Concept

The main idea of the LeAP application was to provide a system which can easily be embedded into the existing legacy environment of the university and is easy to maintain and to upgrade in future. Therefore, it had to fit into the world of legacy systems while simultaneously generate few dependencies to the other established applications.

We therefore decided to split the solution into as many different modules as necessary. These modules communicate with each other via a RESTful API and can easily be improved or replaced without affecting the rest of the solution. LeAP can be divided into three types of components. The main part is the core module which holds the largest part of the business logic and deals with the connection to the database. It consists of several sub-modules which are quite independent of each other. Each of these modules provide a separate API to the other components.

The second type of component are the plugins for the existing legacy applications. At the beginning, this is mainly the university's e-learning platform ILIAS (Integriertes Lern-, Informations- und Arbeitskooperations-System). As a further development, the integration into the campus management system and further applications like email or the university library are planned. The first plugin was embedded into the web appearance of the e-learning platform. It gathers the system-specific user data and sends them to the LeAP-core application. In addition, the plugin checks the availability of prompts for the current user, injects the prompt into the web page and deals with the prompt's response.

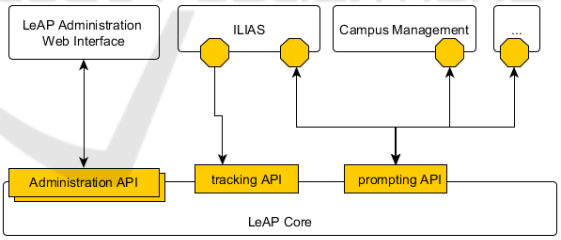

Figure 1: General concept of the *LeAP* solution.

The third type of component are stand-alone web applications. At the current stage of the project, this only includes the administration web interface. It is a stand-alone web application, written with the Angular.js library which communicates with the core application via a separate administration API as shown in Figure 1.

2.2 Data Privacy Issues

One of our main concerns was the handling of data privacy issues. As almost every LA feature collects and processes user data by default, it was inevitable

to consider this topic, particularly in regard of the country's data privacy act. We decided to work within the running, productive environment of our university as soon as possible. Therefore, we were able to collect real data and were not biased by an experimental setting. But convincing the university's IT department to set up our solution within their running environment required additional security and privacy arrangements. Such issues have been documented in recent publications regarding ethical issues and dilemmas in LA (Slade and Prinsloo, 2013; Pardo and Siemens, 2014; Ifenthaler and Schumacher, 2016).

As shown in Figure 2, we decided to use a pseudonymisation in two steps. Wherever we are in direct touch with the students' activities, we use a 32-bit hash value as an identifier. All tracking events and prompting requests use this hash value to communicate with the LeAP core application. The LeAP core API then takes this hash, enriches it with a secret phrase (a so-called pepper) and hashes it again. The doubled hash is then stored within the core's database. As a result, we can match new student generated data to already existing data but are not able to directly trace back a specific student by a given date within the database.

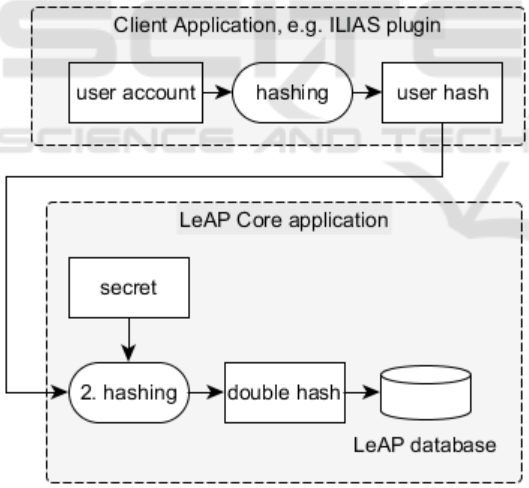

Figure 2: Concept of the encryption of student's identity.

Another benefit of the cooperation with the university's administration is that we do not need to collect demographic student data, as we can catch hold of them from the university's administration afterwards. We are able to receive this data pseudonymised in the same way, so it can be matched with the rest of our collected data.

Upon completion of the current project phase, we will be able to combine the tracking data, the prompting feedbacks, the students' grades, and demographic data for a full-featured analysis without need to have access to this personal data during the data collection phase.

2.3 LeAP Core

The LeAP core component is developed in Java and deployed as a Spring-Boot application. Spring-Boot applications are Java systems which use the spring-web-framework and are deployed with an integrated web application server. Therefore, they can be started as a separate process without the need of an extra web application server like Tomat or Glassfish. In fact, a Tomcat, Undertow or Jetty web server is embedded directly into the executable java file when building the application (Spring Boot Project, 2017).

The structure of the core component is built upon several disjoint modules as shown in Figure 3. These modules offer a separate API to one of the other component types outside of the core. This independence of the modules ensures an easy maintenance and improvement of individual modules without interfering with each other.

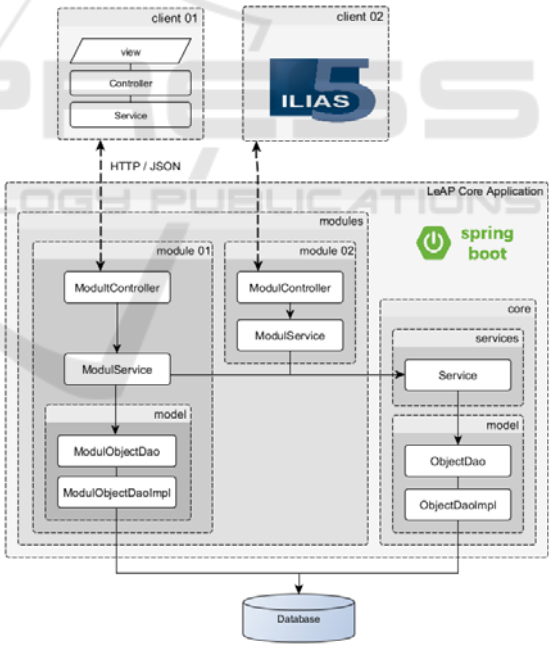

Figure 3: Technical structure of the *LeAP* core architecture.

The application's core part offers a few functionalities which can be used by all modules. The core mainly consists of universally available data objects and database functionality. Beneath the data objects are students, courses, resources, and events. In contrast, prompts are not part of the core and are organised within a separate module.

Data stored in the application is categorised into two types. The first type are resource data like courses and objects. These are stored with an obvious external relation to the object within the source system. For example, reading materials are stored with an external id, which is similar to the id given to the file within the e-learning platform.

The second type of data are individual-related. Beneath these are the students themselves and events which can be assigned to them. These dates have no obvious relation to an external object. They are identified by an individual hash value which is built upon the student's university account and additional secret as described before. This data is not completely anonymous but it ensures a certain amount of privacy through this pseudonymity. Thereby, new user generated data can be connected to a specific hash, however, the user cannot directly be identified by this hash. Data like name, gender, or age are not stored within the LeAP core as they can be gathered from the university's administration later on.

For the projects pilot phase, we only use one instance of the core component which is responsible for the connection to the database and handles all data streams which occur in the current learning analytics environment. But the concept is oriented to duplicate this core component to spread data load and to approach a variety of security requirements. We operate one API at 24/7 which accepts the incoming tracking events and simultaneously operates an API for the lecturer's administration interface which can easily be taken down for steady improvements.

2.4 Plugin for e-Learning Platform

The student's first point of contact with the LeAP system is the learning management system. We developed a plugin for our local learning management system ILIAS which coordinates the tracking and prompting within this system and allows students to choose their current tracking status.

The plugin is written as an *UserInterfaceHook* which adds a new function to the visible layout of ILIAS. The functionality can be enabled for a specific course, which allows the students to see a new tab 'LA-Profile' for setting their personal tracking status. These status are 'active', 'inactive', and 'anonymous'. While in status 'inactive', no data is tracked. In status 'active', the data is allocated to the described, individual, pseudonymous hash. Whereas in status 'anonymous', the data is tracked, but not allocated to a personalized id.

As depicted in Figure 5, additional JavaScript libraries for tracking and prompting, are dynamically embedded during the rendering phase of the page. This new code is augmented with tracking and user information and handles the communication with the LeAP core application.

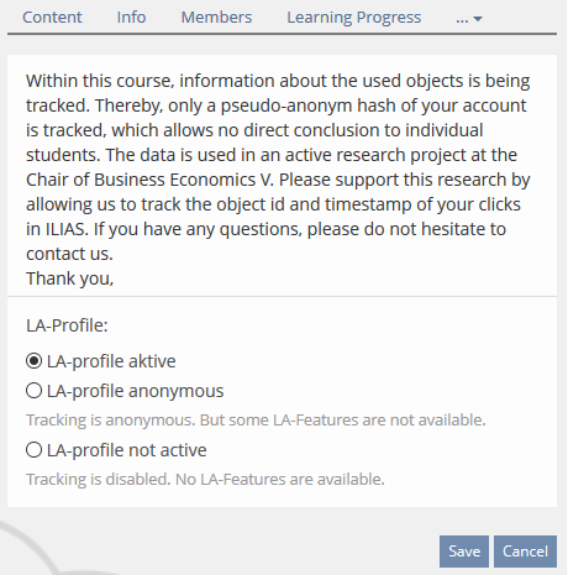

Figure 4: LeAP student privacy features.

Thus, the tracking and prompting features almost completely run within the user's web browser and do not interfere with the ILIAS system.

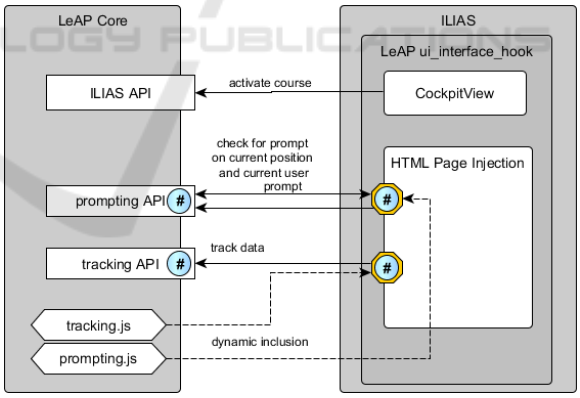

Figure 5: LeAP plugin figure for injection.

As ILIAS is written in PHP, the plugin is also written in PHP. The tracking and prompting libraries are asynchronous JavaScript.

2.5 Prompts

Beside the pseudonymous and anonymous tracking of the students' activities, prompting is currently the second main feature of the LeAP project. Tracking allows us to identify individuals, which should

receive personalised prompts. For example, students who over- or underuse some of the learning features or materials. But prompts can also be given to a complete course. Prompts are always person and location-related. We can put a prompt for every student at the start location of the course, or position a prompt for an individual student to a specific learning material. The prompt is then fired when the student hits that location. But whereas the location is identified by an obvious identifier, persons are only visible in their hash value representation. No personal data like the name of the persons are available within the prompting functionality. Students can only be chosen by their course membership and activities.

When a prompt is fired it is displayed as a small message window in an overlay above the active page as shown in Figure 6. The underlying page is greyed out and cannot be used as long as the prompt is visible. The prompt can consist of a static information message, a question with a text input possibility, a question with a five-point Likert answer possibility, a checkbox, or a combination of these. In addition, we can present a link to a research questionnaire which dynamically adds the student's personal hash value. Thereby, we are able to collect data for accompanying research without collecting the student's personal data or forcing them to reuse a given token. The various questionnaires are all brought together by the hash, which remains constant.

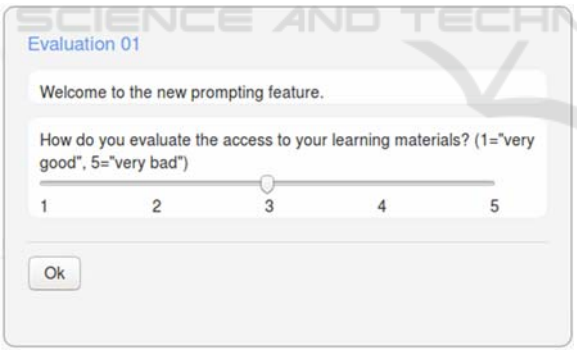

Figure 6: Prompt example of a five-point Likert question.

Beside student and location, prompts can also be executed at a given time and for a given duration. Prompts can therefore be active for a few hours or several weeks. Multiple, different prompts can be active at the same time for several students.

3 PRELIMINARY RESULTS

3.1 Tracking Data

The system is now running reliably for two months, since the start of the fall semester. It is activated in two courses with approximately $N = 400$ students. One course is in the field of economic education, the other in the field of computer sciences. We collected more than 120,000 events and tracked the usage of over 200 learning resources. The underlying technology stack works flawless. The collected data is reliable and satisfies the requirements for later analysis.

3.2 Prompts

During the fall semester, we performed nine prompts in the productive learning environment. Each prompt lasted for one week. We prompted between 150 and 250 students and received response rates between 11% and 64%. The handling of the prompting tool is flawless. The pilot lecturers had no difficulties to create, manage, and submit their prompts. The prompts have been widely accepted and we received no information about noticeable difficulties. Additional survey data is currently analysed which investigates the students' perception toward learning support of the prompts.

3.3 Data Privacy

The default tracking for students' data at the beginning of the semester is set to 'anonymous'. The students are free to change this to 'active' or 'inactive' at every point in time. We informed them several times about the functionality and options. Indeed, we informed them, that it is an active research project and would be happy to have as much participants as possible. But we also guaranteed, that we are not able to identify the individuals until the end of the semester and therefore it could not have an influence on their grading or future studies.

After three weeks, we had 65 active students, 4 inactive students and 348 anonymous students.

4 DISCUSSION

4.1 Data Privacy

As we are seeking to provide a full learner centric system in the future, our approach starts with the

learners' decision to provide their learning progress data. The solution with using a MD5 hash value of the students' university accounts at the front and a doubled hashed value in the core application ensures a satisfying amount of privacy for the projects pilot phase (Slade and Prinsloo, 2013; Pardo and Siemens, 2014). We are able to compute an anonymous, complete, coherent dataset at the end of the semester, without the need to store critical, personal data during the semester.

But as a MD5 hash is not unique, it exists a minuscule possibility to dilute our dataset. In theory, two different university accounts could be hashed to the same value. The current system would not be able to separate them. Nonetheless, this probability is quite low. The hashing and merging of the different data sources is therefore a topic of current research in our project.

The students appreciate the option to include or exclude themselves from the data tracking but mostly ignore this possibility and stay in status 'anonym'. To what extend this is based on an active decision or passive laziness is a topic of further investigation and is depended on their individual privacy calculus for disclosing personal data (Ifenthaler and Schumacher, 2016).

4.2 Impact of Prompts on the Learning Progress

As this part of the project started just at the beginning of the fall semester 2017, we are not yet able to provide convincing insights regarding the impact on the students' learning progresses. We are currently performing a research study, whose results will be available at the end of the semester. Beside the prompts within the productive learning environment, we set up a dedicated copy of the university's learning platform and used this laboratory system to investigate the impact of different prompting types on the students learning progress under laboratory conditions with various sample groups. The first insights might be presented at the conference.

5 CONCLUSIONS

We implemented a tracking and prompting solution into the existing e-learning infrastructure of our university by injecting the respective functionality through separate JavaScript libraries into the legacy systems. By tracking the students via a pseudonymous hash, we are able to collect students' data throughout various systems without the necessity to collect further personal data (Pardo and Siemens, 2014). We are further able to merge this data with other university known data like demographic data and grades at the end of the semester into a complete, anonymous dataset for further investigation.

The solution is used to perform various educational research studies, focussing on effects of prompting for self-regulated learning (Bannert, 2009). We are further planning to extend the various LA features. The next step is the extension of the students' direct feedback. The students will get a more transparent feedback on the amount and type of data which was collected and how this data can be allocated to their current learning processes. Furthermore, we will steadily improve the application and plan to extend the area of research to more courses in the following semester.

REFERENCES

Bannert, M., 2009. *Promoting self-regulated learning through prompts.* Zeitschrift für Pädagogische Psychologie, 23, 139–145.

Davis, E., 2003. *Prompting middle school science students for productive reflection: generic and directed prompts.* Journal of the Learning Sciences, 12(1), 91–142.

Gašević, D., Dawson, S. and Siemens, G., 2015. *Let's not forget: Learning analytics are about learning.* TechTrends, 59(1), 64–71

Ifenthaler, D., 2012. *Determining the effectiveness of prompts for self-regulated learning in problem-solving scenarios*. Journal of Educational Technology & Society, 15(1), 38-52.

Ifenthaler, D. and Schumacher, C., 2016. *Student perceptions of privacy principles for learning analytics.* Educational Technology Research and Development, 64(5), 923–938.

Ifenthaler, D. and Widanapathirana, C., 2014. *Development and validation of a learning analytics framework: Two case studies using support vector machines*. Technology, Knowledge and Learning, 19(1–2), 221–240.

McLoughlin, C. and Lee, M. J. W., 2010. *Personalized and self regulated learning in the Web 2.0 era: International exemplars of innovative pedagogy using social software.* Australasian Journal of Educational Technology, 26(1), 28–43.

Pardo, A. and Siemens, G., 2014. *Ethical and privacy principles for learning analytics.* British Journal of Educational Technology, 45(3), 438–450.

Slade, S. and Prinsloo, P., 2013. *Learning analytics: Ethical issues and dilemmas.* American Behavioral Scientist, 57(10), 1510–1529.

Spring Boot Project: 2017. https://projects.spring.io/spring-boot/. Accessed: 2017-10-02.

Verbert, K., Manouselis, N., Drachsler, H. and Duval, E., 2012. *Dataset-driven research to support learning and knowledge analytics.* Educational Technology & Society, 15(3), 133–148.

Veenman, M. V. J., 1993. *Intellectual ability and metacognitive skill: Determinants of discovery learning in computerized learning environments.* University of Amsterdam, Amsterdam.

Zimmerman, B. J., 2002. *Becoming a self-regulated learner: An overview.* Theory into Practice, 41(2), 64-70.