Smart Learning Analytics

David Boulanger[1], Jeremie Seanosky[1], Vive Kumar[1], Kinshuk[1], Karthikeyan Panneerselvam[2], Thamarai Selvi Somasundaram[2]

[1] Athabasca University, Athabasca, Canada
david.boulanger@dbu.onmicrosoft.com
[2] Anna University, Chennai, India

Abstract. A smart learning environment (SLE) is characterized by the key provision of personalized learning experiences. To approach different degrees of personalization in online learning, this paper introduces a framework called SCALE that tracks finer level learning experiences and translates them into opportunities for custom feedback. A prototype version of the SCALE system has been used in a study to track the habits of novice programmers. Growth of coding competencies of first year engineering students has been captured in a continuous manner. Students have been provided with customized feedback to optimize their learning path in programming. This paper describes key aspects of our research with the SCALE system and highlights results of the study.

Keywords: SCALE framework · smart learning environment · programming · e-learning technologies · novice programming · big data learning analytics

1. Introduction

Smart learning could mean customized learning that optimizes learning pathways, engages learners in positive interactions and guides instruction in a goal-oriented fashion. While the why (optimal learning through customization), where (ubiquitous learning interactions), and how (technologies for goal-oriented learning) of smart learning environments are rather obvious at a coarser level, the degree of customization, the scalability of ubiquity, and the integration of learning-related data are still key challenges facing educational technologists. Smart learning environments encompass traditional classrooms as well as online and distance education. Taking learning anywhere and everywhere in a consistent fashion requires technologies that move such as the smart phones supported by 3G and 4G networks [4] as well as learning environments that move such as the flipped classrooms at homes. To provide context-aware learning, hardware and software sensors are necessary to recognize the context and the learning needs of the user to tailor learning content and activities. Smart learning environments are expected to be highly distributed and cloud-based to accommodate federated and goal-oriented study activities. To make a smart learning environment context-aware, technologies need to collaborate seamlessly and purposefully in order to recognize the context, translate the knowledge of the context in a proper learning recommendation and provide learning materials based on that recommendation. [9] provides an

G. Chen et al. (eds.), *Emerging Issues in Smart Learning*,
Lecture Notes in Educational Technology, DOI 10.1007/978-3-662-44188-6_39

interesting description of a context in e-learning: a user's prior knowledge, learning style, speed of learning, current activities, goals, available learning time, location and interests. In this paper, we will introduce the SCALE system and outline how it relates to smart learning environments following a short literature review on SLEs. Results of a preliminary study are described afterwards.

2. Literature Review

Smart learning environments involve context-awareness. However, context may involve almost anything. Different research projects on smart learning environments may analyze and focus on different aspects of a context. The precision with which a context is defined and recognized by an SLE will influence its overall performance significantly.

At Sookmyung Women's University, Seoul, South Korea, a smart cloud computing framework has been developed according to a model called E4S (elastic four smarts) to provide smart learning services [2]. This model consists of four basic services: pull, prospect, content, and push. The researchers rely on built-in sensors in mobile devices to define the user's behavior or environment. The pull service will extract the type of content to be delivered to the user. The prospect service is responsible for the preparation of the learning content to comply with the user's context. The content service generates the content and establishes the connection between the server and the target device. Finally, the push service performs the synchronized delivery of the generated content to the target device.

In order for systems to adapt to changes in environments, the Technical University of Cluj-Napoca designed a self-adapting algorithm for context-aware systems. *"The algorithm is characterized by a closed feedback loop with four phases: monitoring, analyzing, planning, and execution,"* [6]. This algorithm uses the RAP (Resources, Actions, Policies) context model to programmatically describe the sensed environment, a task which is part of the monitoring phase. The analyzing phase involves evaluating the changes in the context using the context entropy concept in order to determine how much the context follows a predefined set of policies. The planning phase explores all the system's states to select the proper adaptation action which the system should take to respond to context changes. The execution phase implements the adaptation action as defined in the planning phase to change the system's state accordingly.

Zhiwen Yu et al. [9] discuss about a semantic infrastructure, the Semantic Learning Space, for context-aware e-learning. The Semantic Learning Space *"supports semantic knowledge representation, systematic context management, interoperable content integration, expressive knowledge query, and adaptive content recommendation"*. [9] recognizes the need to adapt the learning content to the user's context which is a challenge distinct from flexible content delivery. It also defines the e-learning context as *"a user's prior knowledge, learning style, speed of learning, current activities, goals, available learning time, location and interests."* For example, in a smart learning environment, the system will track the knowledge gap between the current user's knowledge and the targeted learning outcomes and provide the user with the proper learning content to fill that gap taking into account the user's context. In another article, Zhiwen Yu et al. proposed an ontology-based approach for semantic content recommendation in order to get one step further toward sophisticated context-aware e-learning. The recommender takes into account knowledge about the learner, knowledge about content, and knowledge about the learning domain in order to offer the right thing to the right person in the right way at any time, at any place, and in the right form [10]. The recommender goes through the following sequence of steps: semantic relevance calculation, recommendation refining, learning path generation, and recommendation augmentation.

Kosba and his associates have developed the Teacher ADVisor (TADV) framework which uses LCMS tracking data to elicit student, group, and class models, and using these models help educators gain better understanding of their distance students [17]. It uses a set of predefined conditions to recognize situations that require educators' intervention, and when such a condition is met, TADV generates an advice for the educator, as well as a recommendation for what is to be sent to students. Whereas TADV is focused on the educators' day-to-day activities, our approach aims at helping them rethink the quality of the employed learning content and learning design. Our approach also helps students share their experience, reuse additional learning resources collected by their peers, and get fine-grained feedback about their progress.

3. SCALE Framework and Smart Learning

SCALE is a mixed-initiative learning analytics framework aimed at collecting learning traces from any learning domain and analyzing those learning traces to extract the underlying competency levels in the same learning domain. The SCALE framework has been designed for a full integration with a learning management system such as Moodle as well as a suite of automated grading and testing tools such as Web-CAT[1] to make sensed learning traces reliable and associable to learning outcomes. SCALE does not focus as much on the physical context of a student as it does with the student's learning context (i.e., background knowledge).

SCALE's layered architecture consists of a sensing layer, an analysis layer, a competency layer and a visualization layer. The sensing layer is implemented through the Hackystat[2] framework which provides a collection of preset and customized sensors embedded in learning analytics tools. The analysis layer consists of parsers and analysis tools pertaining to the learning domain. For instance, SCALE's analysis layer applied in the programming domain will consist of compilers and static/dynamic code analysis tools. The output of the analysis will then be converted and stored in a comprehensive competency ontology. The competency layer will associate competencies with learning outcomes and show the evidences that the student is progressing or not toward those learning outcomes. Ontologies, implemented using Semantic Web technologies, along with inference engines will pave the way towards discovering new patterns and trends in the learning styles and learning paths of students. Competency ontologies will hold and define the knowledge background of students, a prerequisite to offer customized

[1] Web-CAT (http://web-cat.org/group/web-cat)
[2] Hackystat (https://code.google.com/p/hackystat/)

learning materials. Finally, the visualization layer will provide a graphical interface consisting of a set of visualization and communication tools to play back the student's performance, display the student's competencies in relation to the latest learning activities, provide an environment where all learning stakeholders (i.e., instructors, students, peers, parents, recruiters, etc.) can meet and discuss how to set new goals and how to reach them, and give the student the opportunity to comment his/her learning to optimize the system's understanding of the student. We have designed the framework with a plug-and-play architecture to allow any data-centric learning-oriented application to be plugged in to SCALE.

We aim at making SCALE context-aware in terms of user's prior knowledge, regulated learning (self-regulated and co-regulated), learning style, learning efficiency, current activities, goals, available learning time, location and interests, as partly defined by Zhiwen Yu [9]. Programming is by far one of those easily traceable (not necessarily analyzable) domains due to the availability of explicit data that identify stepwise progressions made by programmers as they complete their coding tasks. In other words, the number of sensing hours spent by the system to track and update the learning context of the student may be much greater than in a chemistry course (depending on how e-learning is applied to the chemistry field). Due to the great number of environments in which Hackystat sensors (one of the SCALE cornerstones) may be embedded, the system may have a better representation of the user context.

One important question concerns the degree of ubiquity of SCALE. This may be a tricky question to answer. Consider the Java programming domain as an example. We have integrated three different programming tools within the SCALE system to capture coding related activities of learners – Eclipse IDE sensor, Virtual Programming Lab[3] (VPL) IDE sensor, and MI-LATTE reflection and regulation sensor. The dilemma consists in reducing the physical learning environment to certain computing devices due to the specificity of the software to be used as well as the competencies that the student must develop to master coding habits and competencies. The mastery of the software may even be a learning outcome of the course so that the student may become proficient. Eclipse, which is a professional integrated development environment, cannot be run on mobile devices. However, in the setting of the experiments that will be conducted at Athabasca University and Madras Institute of Technology, students will have to work through the assignments and many of the programming exercises using Eclipse. On the other side, students may install Eclipse and work on as many computers as they wish. We intend to make the SCALE framework available everywhere on the planet where an Internet connection is available. The SCALE system will guarantee reliable data collection through the Hackystat sensors (contributing to context awareness) despite low-speed Internet connections and inevitable connectivity issues. On the other side, we have other tools which may be accessible through the Web (VPL and MI-LATTE). SCALE could offer programming exercises on smart phones through these alternative tool technologies. These programming tools will nevertheless support only small-scale programming exercises.

In the future, we plan to provide students with a gamut of recommendations about which learning paths to take, which course or career to select, how to prepare for job

[3] Virtual Programming Lab (http://vpl.dis.ulpgc.es/)

interviews, which students have similar cognitive profiles, and which steps to undertake to compete with top-level students. Currently, SCALE provides students with minimalist recommendations about the programming concepts that should be deepened in order to maximize their success in the upcoming assignments or exams. The SCALE framework is a work in progress and will implement more sophisticated recommendations as we develop the system further.

4. Experiment with SCALE Forerunner e-Learning Technologies

Athabasca University, Canada, and the Madras Institute of Technology (MIT), Anna University, India, have conducted an experiment to analyze the introduction of tracing-oriented learning technologies among first year engineering courses. The traces target enhancement of the student's learning experience and possibly his/her performance within a course. The experiment was conducted at the MIT campus in the setting of a C programming course among 767 participating students and 10 professors (one professor per classroom). Students belonging to nine different classrooms received traditional lectures while a randomly chosen 10th classroom received a traceable online learning environment in Moodle in addition to classroom lectures. The e-learning technologies introduced in the course include the Moodle learning management system, the Virtual Programming Lab, the Eclipse IDE sensor, and CTAT tutors. The CTAT tutors guide students to solve programming exercises at a finer level of one line of code at a time. The study content was presented to students using a quadrant-based framework [18,19,20,21,22]. The new design followed a four-step process: watching, discussing, conceptualizing, and trying out. It also provided guidelines to the student as to how to study instead of what to study. All of these technologies trace study habits of learners at finer levels of granularity. Further, collected data were integrated in a singular framework to associate datasets originating from different sensors.

The objective of this study consists in discovering new trends and examining how the student's performance behaves when elements of smart learning environments are part of his/her learning experience. Since this experiment did not occur in a controlled environment, the reader should note that we will not claim anything from the results of this experiment except that we will pay attention to potential patterns and confirm them in upcoming experiments which will include more state-of-the-art e-learning technologies and a cutting-edge design of the learning process oriented toward self-regulated and co-regulated learning.

The experiment involved 10 different classrooms. Approximately 75 students attended each classroom and each classroom had a distinct professor. Classrooms were numbered from classroom0 to classroom9. Classroom3 is the classroom of interest in this experiment. The performance of classroom3 will be compared to the average performance of all the other classrooms. All classrooms teach the same course using the same structure. The course consists of three assessments, one theory exam, and one practical exam. Assessment1 consisted of theoretical questions while Assessment2 and Assessment3 consisted of programming exercises. Hence, classroom3's students were not yet exposed to the Virtual Programming Lab tool in Assessment1 while they had access to the entire course content in Moodle (in which was implemented the new instructional design). We will analyze or rather observe the performance of classroom3 in comparison to the other classrooms at the end of the semester.

Figure 1 displays the average marks for each assessment for both classroom3 and the other classrooms. Figure 2 shows the percentages of students who passed the assessments. Both figures show that classroom3 seemed to perform less well in the first assessment. We elaborate some hypotheses which could partially explain the reason(s) why classroom3 got inferior average grade marks. 1) Did it take classroom3 students more time to adapt to the new instructional design? 2) Was the new instructional design not optimal for theoretical parts? 3) Would the new instructional design have been optimal if it had been supported by the proper e-learning technologies? We will strive to validate/invalidate these hypotheses in future experiments in more controlled environments. However, the graphs show that

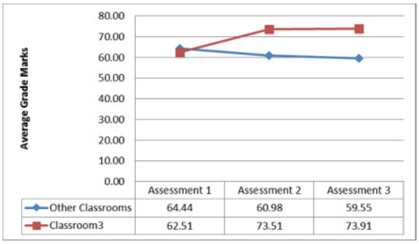

Fig. 1. Assessment Average Grade Marks

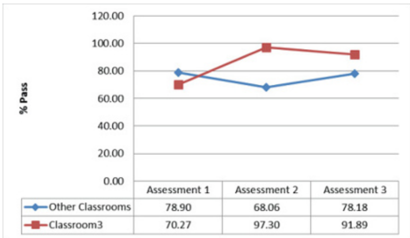

Fig. 2. Assessment % Pass

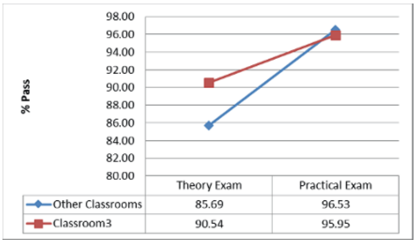

Fig. 3. Exam % Pass

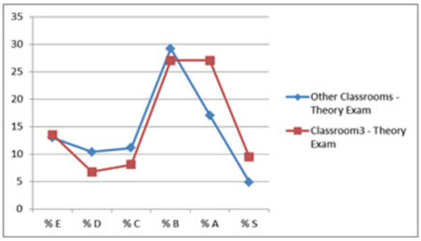

Fig. 4. Theory Exam % GPA

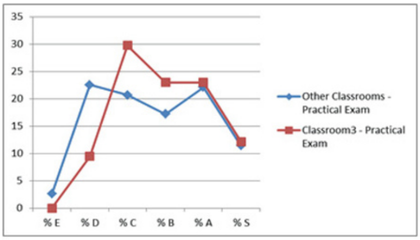

Fig. 5. Practical Exam % GPA

average marks and pass percentages of classroom3 are significantly higher than those of the other classrooms in Assessments 2 and 3. We would like to validate if classroom3's students started benefitting from the new instructional design and new e-learning technologies after Assessment1.

As for the theory and practical exams, Figure 3 denotes that the percentage of students in classroom3 who have passed the theory exam (almost 5% over) is greater than the percentage of students in the other classrooms. This observation may support the hypothesis that students took more time to adapt to the new instructional design but that this design optimized in some way the student’s learning experience afterward. As for the practical exam, the percentage of students in the other classrooms who passed the practical exam is very slightly greater than classroom3’s student pass percentage. However, both classroom3 and the other classrooms perform quite well in the practical exam. We may, nevertheless, observe that the pass percentage gap between the theory and practical exams is greater in the other classrooms.

Finally, in Figures 4 and 5 we see that fewer students in classroom3 have GPAs C, D, or E, and more students in classroom3 get GPAs S, A, and B. Note that the order of GPAs from best to worst is S, A, B, C, D, and E. All these are mere observations and suggest some trends. More experiments will be conducted in the near future in several universities across the world to understand the impact of smart learning environments on student performance and to confirm our hypotheses following this experiment's results.

5. Future Work and Conclusion

SCALE will also incorporate a Causal Learning Analytics (CLEAN) extension to determine the causes of various learning-related occurrences. This will include among other things identifying the causes of the successes and failures in learning outcomes and determining the impact that those factors have on the learning outcomes to name a few. Furthermore, SCALE will track the type and sequence of programming activities (debugging, compiling, testing, documenting, code writing, etc.) typical for every student category (at risk, average, and top students). SCALE will also look for the learning approaches and behaviors which are the most effective as well as the conceptual causes of student's errors. The CLEAN extension will be implemented as a rule-based subsystem using pattern-matching techniques (i.e., production rules). In summary, SCALE aims at tracking a student's competencies in as much learning activities as possible and at explaining the factors contributing to the strengthening of those competencies.

References

[1] Burghardt, C., Reisse, C., Heider, T., Giersich, M. & Kirste, T. (2008). Implementing Scenarios in a Smart Learning Environment. 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (pp. 377-382).

[2] Kim, S., Song, S.-M. & Yoon, Y.-I. (2011). Smart Learning Services Based on Smart Cloud Computing. Sensors 11, 7835-7850.

[3] Aion, N., Helmandollar, L., Wang, M. & Ng, J. (2012). Intelligent Campus (iCampus) Impact Study. 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT), Vol. 3 (pp. 291-295).
[4] Lee, J., Jung, Y. J., Park, S. R., Yu, J., Jin, D.-s. & Cho, K. (2012). A Ubiquitous Smart Learning Platform for the 21st Smart Learners in an Advanced Science and Engineering Education. 2012 15th International Conference on Network-Based Information Systems (NBiS) (pp. 733-738).
[5] Huang, R., Hu, Y., Yang, J. & Xiao, G. The Functions of Smart Classroom in Smart Learning Age.
[6] Cioara, T., Anghel, I., Salomie, I., Dinsoreanu, M., Copil, G. & Moldovan, D. (2010). A self-adapting algorithm for context aware systems. 2010 Ninth Roedunet International Conference (RoEduNet) (pp. 374-379).
[7] Koo, D.-H. (2012). Trends and Revitalization of Smart-Learning in Elementary and Middle Schools. Asian Journal of Information Technology, 160-168.
[8] Mikulecký, P. (2012). Smart Environments for Smart Learning.
[9] Yu, Z., Zhou, X. & Shu, L. (2009). Towards a semantic infrastructure for context-aware e-learning. (pp. 71-86).
[10] Yu, Z., Nakamura, Y., Jang, S., Kajita, S. & Mase, K. (2007). Ontology-Based Semantic Recommendation for Context-Aware E-Learning. (pp. 898-907).
[11] Akiyoshi, M. & Nishida, S. (1993). A qualitative simulation-based learning environment: how to enhance causal understanding of complex phenomena in large-scale plants. 1993 Twelfth Annual International Phoenix Conference on Computers and Communications (pp. 531-537).
[12] Blikstein, P. (2011). Using Learning Analytics to Assess Students' Behavior in Open-ended Programming Tasks. In Proceedings of the 1st International Conference on Learning Analytics and Knowledge (pp. 110-116). ACM.
[13] Castro, F., Mugica, F. & Nebot, A. (2007). Causal Relevancy Approaches to Improve the Students' Prediction Performance in an e-Learning Environment. 2007 Sixth Mexican International Conference on Artificial Intelligence - Special Session (pp. 342-351).
[14] Fancsali, S. E. (2011). Variable Construction for Predictive and Causal Modeling of Online Education Data. In Proceedings of the 1st International Conference on Learning Analytics and Knowledge (pp. 54-63). ACM.
[15] Santos, J. L., Govaerts, S., Verbert, K. & Duval, E. (2012). Goal-oriented Visualizations of Activity Tracking: A Case Study with Engineering Students. In Proceedings of the 2nd International Conference on Learning Analytics and Knowledge (pp. 143-152). ACM.
[16] Yu, T. & Jo, I.-H. (2014). Educational Technology Approach Toward Learning Analytics: Relationship Between Student Online Behavior and Learning Performance in Higher Education. In Proceedings of the Fourth International Conference on Learning Analytics and Knowledge (pp. 269-270). ACM.
[17] Kosba, E., Dimitrova, V., Boyle, R. (2005). Using Student and Group Models to Support Teachers in Web-Based Distance Education. In Proceedings of the 10th International Conference on User Modeling, Edinburgh, UK (pp. 124-133).
[18] Lee, S., Barker, T., Kumar, V. (2014). Self-Directed Learning. Educational Technology & Society, in preparation.
[19] Lee, S., Barker, T., Kumar, V. (2011). Learning Preferences and Self-Regulation - Design of a Learner-Directed e-Learning Model. International Conferences ASEA, DRBC and EL 2011, Software Engineering, Business Continuity and Education. Communications in Computer and Information Science, Springer Link, Volume 257, pp. 579-589.
[20] Lee, S., Barker, T., Kumar, V. (2011). Models of eLearning: The Development of a Learner-Directed Adaptive eLearning System. In Sue Greener and Asher Rospigliosi (eds.), Proceedings of the European conference on e-learning held in University of Brighton, Brighton, UK, November 10-11, 2011 (pp. 390-398).
[21] Lee, S., Barker, T., Kumar, V. (2010). Approaches to Student Modeling in the Context of e-Learning 2.0. In Paula Escudeiro (ed.), Proceeding of the 9th European conference on e-learning, pp. 59-66.
[22] Kumar, V., Manimalar, P., Somasundaram, T. S., Sidhan, M., Lee, S., El-Kadi, M. (2009). Open Instructional Design. Technology for Education (T4E 2009) (pp.43-50).
[23] Seanosky, J., Boulanger, D., Kumar, V. & Kinshuk (2014). Unfolding Learning Analytics for Big Data. ICSLE 2014, in preparation.

[April 14, 2014]