21ST INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED17
21-25 AUGUST 2017, THE UNIVERSITY OF BRITISH COLUMBIA, VANCOUVER, CANADA

EXPLOITATION OF MICRO-LEARNING FOR GENERATING PERSONALIZED LEARNING PATHS

Rusak, Zoltan
Delft University of Technology, The Netherlands

Abstract

Personalization of learning experience in engineering courses is a key to successfully engage students in any type of learning activity. Personalization is needed to achieve optimal learning experiences taking into account the pace of learning influenced by the background and capability of the learners, their personal interest, and optimal timing of learning exercises. This paper presents the development of an algorithmic solution to personalize learning content and learning paths for teaching Android software development to design students. Our solution recommends micro-learning sessions to students based on their background knowledge, their preferences and ranking of alternative learning contents, and their performance of completing the tests of micro-learning sessions. The recommender algorithm has been applied in an e-learning environment by 68 students of an elective course and the goodness of recommendations was evaluated with the goal to further tune the learning content and the recommendation mechanism. Our results show that ca. 60% of the learning content of the course requires personalization, while the remaining 40 % is suitable for all students without any adjustment.

Keywords: Education, Design learning, Life-long learning, Personalization of learning, Micro-learning

Contact:
Dr.-Ing. Zoltan Rusak
Delft University of Technology
Design Engineering
The Netherlands
z.rusak@tudelft.nl

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 9: Design Education, Vancouver, Canada, 21.-25.08.2017.

1 INTRODUCTION

The potential of personalizing learning content and learning experience have been comprehensively explored by educational researchers and practitioners. It has been shown that due to the extreme amount freely accessible information and knowledge sources on the Internet, engineering education at universities takes place at an unpredictable pace as students have different background knowledge and learning capabilities. Self-training has become an essential part of engineering education due to the massive amount of multimedia tutorials, MOOCs and webinars. Despite these positive developments, the proliferation of personalized learning is still in its infancy. Addressing personalized learning, on the one hand, is essential to achieve optimal learning experiences taking into account the pace of learning influenced by the background and capability of the learners, their personal interest, and optimal timing of learning exercises. On the other hand, it was found that learners are more likely to be engaged if they are more active, autonomous and have full control over their learning process (Brady, 2004). This implies that adaptable systems with recommendations are expected to be more accepted by learners than self-adaptive learning systems capable to automatically adjust learning exercises to the needs of learners.
This paper reports on development of a personalizable e-learning application developed for industrial design students of the bachelor program of Faculty of Industrial Design Engineering at the Delft University of Technology, the Netherlands. This software application is used for teaching industrial design students basic programming skills that would enable them to design, conceptualize and prototype software applications. In the Software course, they learn how to design and develop Android apps using UML, XML, and Java. Students enrolled to the course have different background knowledge and skills of programming that is ranging from novice to expert level. This, however, creates exceptional challenges for the teachers in providing appropriate course content that suits the students' background knowledge and pace of learning. To address this challenge, a recommendation mechanism for personalizing the learning content has been developed that can guide the students through the content of the course taking into account their actual programming skills, pace of learning and preferences. An accompanying Android application organizes the course content for the students into micro learning sessions and feeds these sessions in a personalized way. Truly personalized content should be driven by micro-learning sessions that enable more accurate composition of learning content and higher flexibility and adaptation to personal need of students.
This paper reports on a concept developed by the author for personalizing course content for teaching programming to design students. First, a state of the art review of methods and tools for personalizing learning experiences is presented. Section 3 discusses the concept of our approach and the mathematical formulation of our recommendation mechanism. Section 4 presents the verification of the recommendation mechanism, and the implementation of our e-learning environment. Finally, we present our findings and conclusions.

2 STATE OF THE ART

Several recommendation systems or adaptive learning systems have been developed to provide personalized learning materials to individual students via analysing their profiles or learning portfolios (Reategui, Boff, and Campbell, 2008). Some studies have employed data mining or statistical methods to analyse the students' learning portfolios and profiles in order to determine the learning materials to be recommended to individual students (Tzouveli, Mylonas, and Kollias, 2008). Researchers have indicated that such technology-enhanced learning environments can make positive contributions to students' learning outcomes (Harri-Augstein and Thomas, 2005).
User models are essential to e-learning systems, giving students learning continuity, tutors evidence of students' progress, and both a way to personalize students' learning materials to their abilities, progress and preferences. Personalizing information has been the motivation behind developing e-learning systems. Adaptive educational systems attempt to maintain a learning style profile for each student and use this profile to adapt the presentation and navigation of instructional content to each student. Student (whose characteristics are: knowledge level, technical education, educational goals, interests, motivation, learning style, personal characteristics, general knowledge, etc. (Markovi 2013) Markovi (2013) developed an e-learning system, which creates an adaptive test that gradually increases the difficulty of questions. Though this approach has potential if the learning content is concise, it has limitation in deriving insights from the experience of other users as it creates similar learning patterns

and paths as the e-learning system does not learn from previous experiences of users with similar profiles.

The recommendation mechanism that facilitates personalization is an elementary need in for implementing content adaptations in e-learning systems. Various methods have been developed to make recommendations to users. Most of the methods are based on: (i) collaborative filtering, (ii) content based filtering, (iii) ant colony optimization, (iv) particle swarm optimization, and (v) different combinations of these techniques (Nilashi, et al. 2013).

Collaborative filtering methods operate with a large amount of information concerning users' behaviours, activities or preferences and, based on their similarity to other users, make predictions on what users will like (Elahi et al. 2016). A benefit of the collaborative filtering method that it is capable to accurately recommend content and items without requiring a model of the content itself. Algorithms such a Jaccard similarity index, k-nearest neighbour or Pearson Correlation has been used to measure user similarity or item similarity in recommender systems. The use of collaborative filtering methods in the domain of e-learning, however, raises many issues. Collaborative filtering is sensitive to data coming from unexperienced users, who may have little or no experience with the content evaluated by them. Their input can typically skew the reliability of the recommendation mechanisms.

Content-based filtering methods use data collected or modelled based on (i) content (e.g. learning material, therapeutic training) or items (e.g. products) and (ii) a profile of the user's preference (Lobo, et al., 2009). In a content-based recommender system, keywords are used to describe the items and a user profile is built to indicate the type of item a particular user likes. In other words, these algorithms try to recommend items that are similar to those that a user liked in the past (or is examining in the present). Content based filtering methods, however, are not able to incorporate the order of items to be recommended for the users. Due to these limitations of collaborative and content based filtering, hybrid approaches are introduced to benefit from their positive properties.

Other methods such as ant colony optimization have been also introduced for developing more comprehensive recommendation mechanisms. Ant colony optimization (ACO) is a metaheuristic technique that operates with a set of software agents, called artificial ants, to search for sufficient solutions to a given optimization problem (Yang and Wu. 2009). The optimization problem is transformed into a path search problem on a weighted graph. The ants (i.e. agents) explore possible paths by moving on the graph and gradually construct a stochastic solution influenced by the pheromones (i.e. weights of the nodes or edges of the graph). Krynicki, et al., (2016) have developed an evolutionary algorithm, which extends ACO with three important features to be utilized in recommending personalized learning programs for people with brain injury. Their algorithm (i) takes into account the state of the individuals to drive the optimization search, (ii) dynamically adapt the recommendation according to the recent behaviour of individual user or a group of users with similar characteristics, and (iii) models the cognitive state of the user as the set of deficits (deductive reasoning, sustained attention, short-term memory, etc.). The solution offers a hybrid model for personalizing possible routes through a learning program, which found to be effective in composition of personalized programs. These methods are especially powerful in situations, when the learning capability of the learner is significantly changes (i.e. during stroke rehabilitation), but in our application the learning capabilities of the students do not significantly change. Therefore, we have applied combination of collaborative and content based filtering approach extended with a prioritizing mechanism of the learning content.

3 CONCEPT OF PERSONALIZED E-LEARNING APPLICATION

The concept of our personalized e-learning application is presented in Figure 1. This figure depicts the data and information flows between the modules of an adaptive e-learning software, that consist of (i) a user profiler, which creates and manages learner profiles based on students personal preferences, their progress of learning and their performance in online tests, (ii) a content profiler that sorts micro-learning sessions based on the rankings by learners with similar profiles, (iii) content recommender that implements a recommendation mechanism able to rank micro-learning sessions based on user profiles, progress of learning, and similarity of content, and (iv) a micro-learning session composer, which selects content from alternatives and questions for user tests based on performance of users with similar profilers.

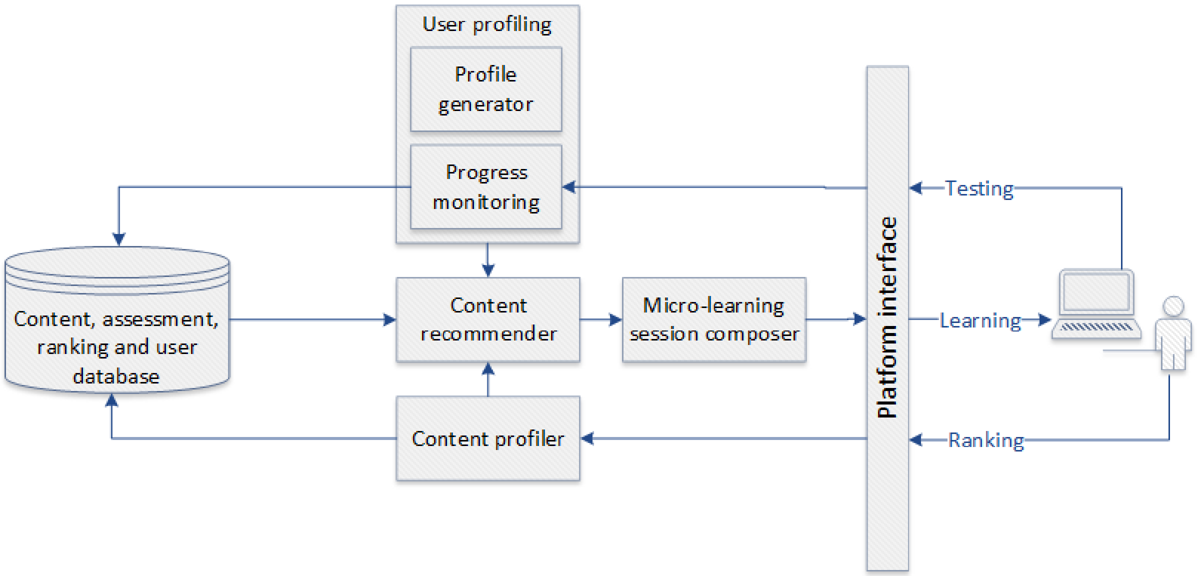

Figure 1. Concept of e-learning application

3.1 Principle of recommendation mechanism

Recommendation of contents for micro-learning sessions is computed based on the similarity indexes of the users, contents and learning progress. Typical recommendation mechanisms do not have to cope with the challenge of ordering the content according to some learning lines. For instance, product recommendations made by marketing software, such as recommendation mechanism of Amazon, do not consider in which order the products should be purchased. In the context of learning, the order of the contents is, however, an important issue. For this reason, we proposed a recommendation mechanism that not only considers the similarity of users, contents and learning progress, but also the dependency of learning content of previous visited micro-learning sessions.

To calculate the similarity indexes, we use the Jaccard index formula, which compares two sets and produces simple decimal statistics between 0 and 1.0. Similarity index of user profiler is defined in Equation (1):

$$Sim(U_1, U_2) = (|E_1 \cap E_2| + |T_1 \cap T_2|) \div |E_1 \cup E_2 + T_1 \cup T_2| \tag{1}$$

where, U_1, U_2 are two users compared, $E_1 \cap E_2$ is the similarity of the users' experiences with course content, $T_1 \cap T_2$ is the similarity of their test results, and $E_1 \cup E_2 + T_1 \cup T_2$ is the total number of similarities. In this formula, the experience of the user represents the background of the student entering the course, while the test results represent the learning progress. Similarity of experience means, if the student has or lacks similar experiences with programming technologies (e.g. basic knowledge of object oriented programming), while similarity of the test results means if the student could/could not answer specific questions of the tests.

Similarity index of contents of micro-learning sessions is defined in Equation (2):

$$Sim_{C0}(C_1, C_2) = \left(\left|L_1^M \cap L_2^M\right| + \left|D_1^M \cap D_2^M\right| + \left|L_1^R \cap L_2^R\right| + \left|D_1^R \cap D_2^R\right|\right) \div |L_1^M \cup L_2^M + D_1^M \cup D_2^M + L_1^M \cup L_2^M + D_1^R \cup D_2^R| \tag{2}$$

where C_1 and C_2 are alternative contents of the same micro-learning session, $L_1^M \cap L_2^M$-are content liked by two users from the same group profile, $D_1^M \cap D_2^M$ disliked session by two users from the same group profile, $L_1^R \cap L_2^R$ -likes of recommendation of micro-learning session C_1, C_2 to follow micro-learning session C_0, $D_1^R \cap D_2^R$ -dislikes of recommendation of micro-learning session C_1, C_2 to follow micro-learning session C_0. The first two components of this similarity index are related to the content of micro-learning sessions liked by majority of users with the same user profile, and the second two components represents one step in learning paths liked/disliked by the users with the same profile.

The content recommender utilizes the probability value determined based on user and content similarities as defined by Equation (3):

$$P(U_m, C_n) = \left(\sum_{i=1}^{k} sim_L(U_n, U_i) + \sum_{i=1}^{l} sim_D(C_n, C_i)\right) \div (|\mathrm{N}_L| + |\mathrm{N}_D|) \tag{3}$$

where $P(U_m, C_n)|C_{n-1}$ is the probability that user U_m will like content C_n after C_{n-1}, $\sum_{i=1}^{k} sim_L(C_n, C_i)$ is the sum of all user similarities and $\sum_{i=1}^{l} sim_D(C_n, C_i)$ is the sum of all content similarities and $|N_L| + |N_D|$ is the total number of users and contents liked and disliked. The probability that the user will like a given session as a follow up of the previous micro-learning session is computed for each micro-learning session. The sessions with the highest probabilities are recommended for the user of the e-learning environment.

3.2 Composing micro-learning sessions

Formal curricular setting of the course includes lectures and workshops addressing the topics in a structured and active way. Informal learning elements are online micro-learning sessions that are composed of a short explanation of the learning goals, tutorials discussing the theoretical and practical nature of the discussed topics, and an assessment tool evaluating the knowledge of the students and obtaining feedback from the students on each session. Micro-learning sessions were designed based on general guidelines reported in the literature, but with consideration to their adaptability to particular user profiles and to the progress of learning. The content of the course was decomposed into 25 topics that are relatively short, so that they can be presented in sessions that are not longer than 5-6 minutes, or can be explained by texts of 2-3 paragraphs, or 2-3 PowerPoint slides followed by online tests consisting of 4-5 questions. The modularized structure of the course is presented in Figure 2.

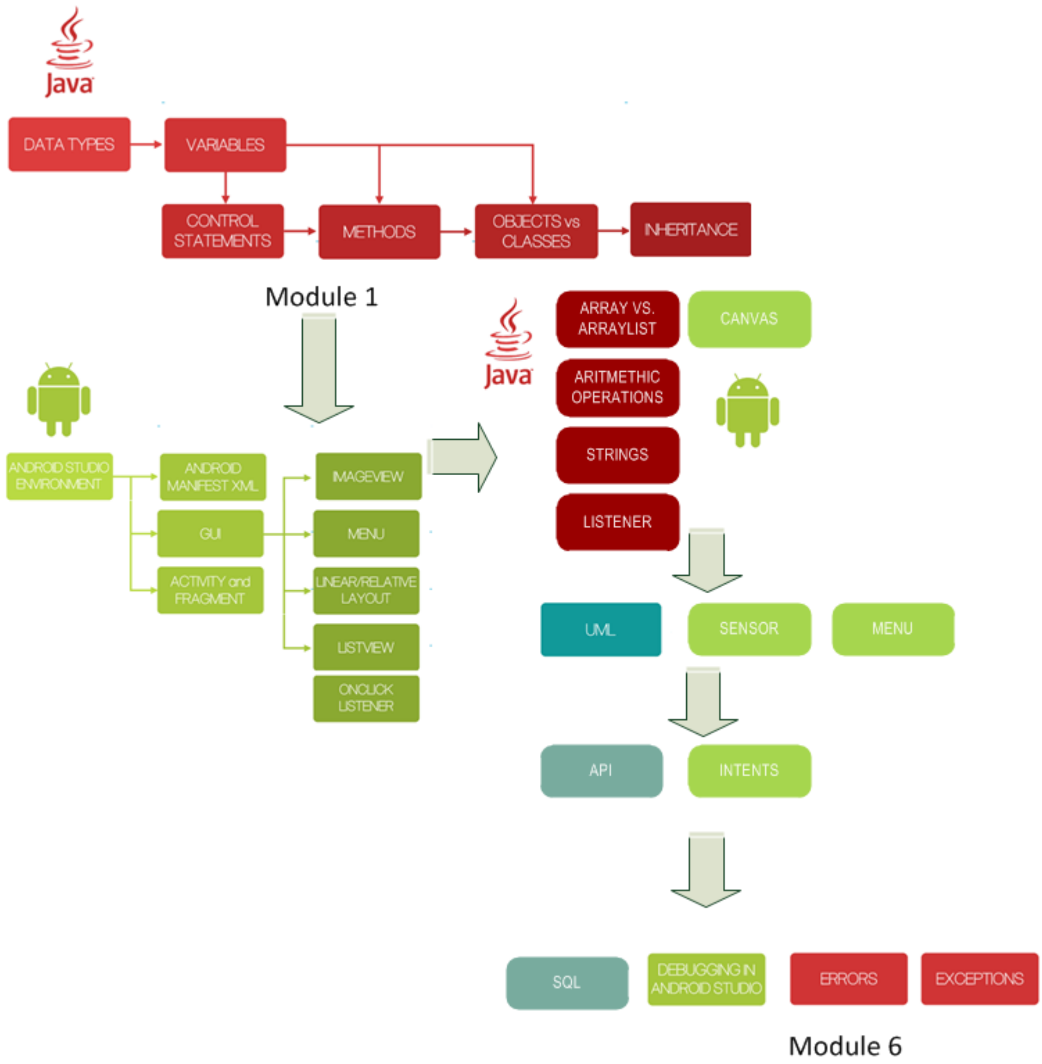

Figure 2. A typical learning path of the course. Line arrows show dependencies of micro-learning sessions and block arrow showing order of modules of formal learning sessions.

Micro-learning session composer adjusts the content for a particular topic to personal preferences for video, textual or PowerPoint material or a composition of these. Each of these content elements is classified based on the depth of explanation in order to consider the programming level of learners and their pace of progress in the composition of a micro-learning session. Test questions are also pre-sorted to classes of easy, medium and difficult. Since the goal of the course is not to train industrial design students to be professional programmers, the learning objective is to provide knowledge and skills that enables them to design and prototype software applications. The test questions are compositions, which in their nature require students to (i) reproduce knowledge, (ii) analyse existing codes, i.e. apply their insights in interpreting programming codes and (iii) create concept solutions, such as generating concepts of code snippets for a given problem. This type of segmentation of the course enables the continuous and semi-automated update of the course content. Classification of content and test questions is updated based on the data collected from the students by the e-learning solution during the course. For instance, questions that are pre-sorted for beginners, and found by them to be too challenging can be identified and assigned to the user profile of advanced programmers.

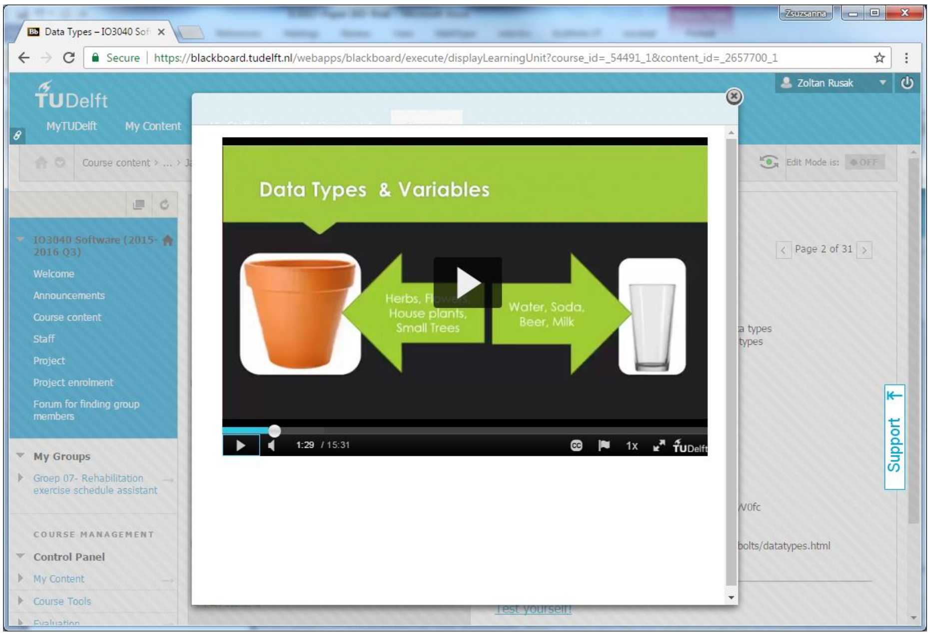

Figure 3. On-line micro-learning sessions: (a) typical learning path generated for beginner students, (b) micro-learning session including explanation of learning objectives, multimedia tutorial, and test quiz

4 VERIFICATION THE RECOMMENDATION ALGORITHM

Before implementing the recommendation mechanism to be deployed in Software course, we have used the data collected from a previous course running in the academic year 2015/2016 in order to verify the proposed concept. The verification of the recommendation mechanism separately addressed the concept of (i) student clustering, i.e. how well the developed questionnaires are able to separate clusters of students based on the similarity index defined in Equation (1), (ii) the content clustering, i.e. how well the similarity index of contents distinguishes preferences of different users and (iii) content recommendation, if probability value of recommendation makes distinctions for making dedicated recommendations for different user groups.

4.1 Clustering students

User profiling is an important aspect of personalizing learning content. Profiling of users of the course was done by evaluating the results of two questionnaires. The first questionnaire focussed on users' earlier experience with programming (e.g. the level of knowledge of various programming languages, years of programming experience), while the second questionnaire assessed their progress of learning by online tests about the topics of micro-learning sessions.

The results of these questionnaires have been processed in Matlab. First, a similarity matrix was generated that represents the distance of answers between all students. Figure 4a and 5a shows the similarity matrices of the user experience and the learning progress, respectively. Using classical multidimensional scaling, clusters within the similarity matrices were explored. The results of this analysis are illustrated in Figure 4b and 5b. The results of the cluster analysis show that the experience of the users forms four clusters of students. We identified these clusters as: beginner, mid-level, advanced and expert programmers. Figure 5b, also shows similar clustering, but the differences in the pace of progress within the clusters are also more visible. This means that the similarity index of the tests is an important aspect to be considered in computing the recommendation probability.

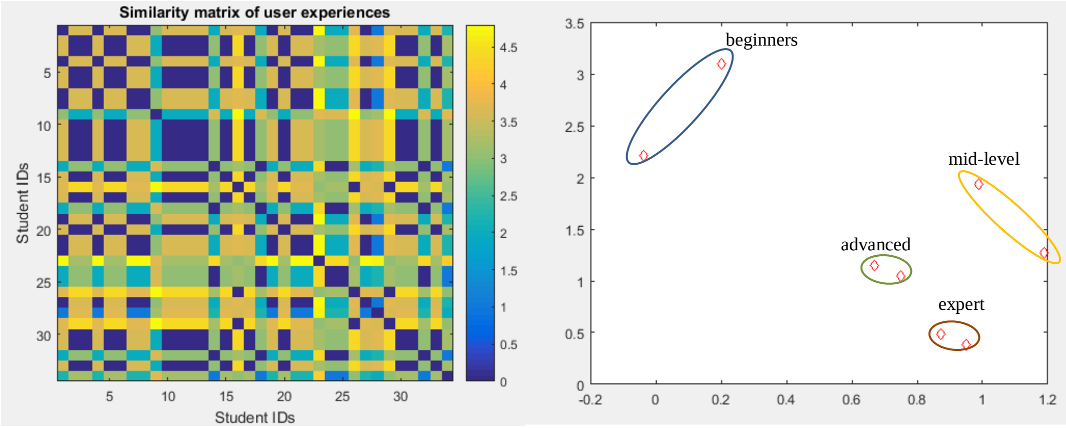

Figure 4. Users clustering based on background knowledge and experience in programming.

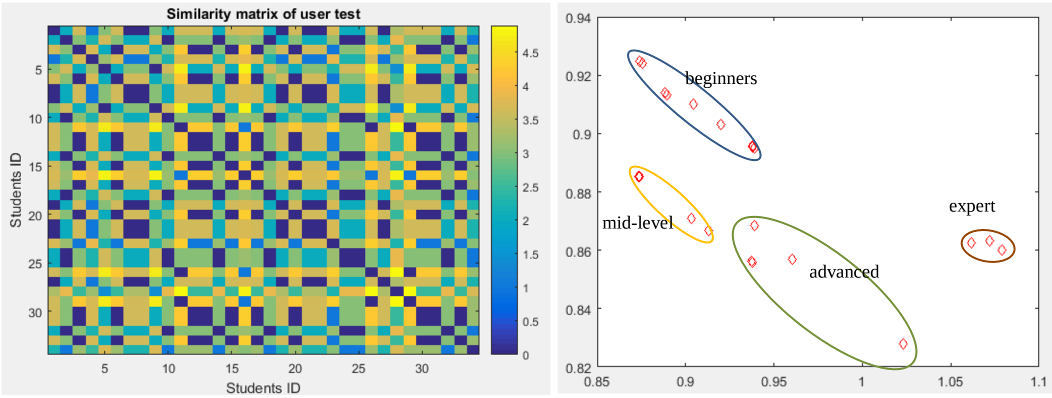

Figure 5. Clustering students based on test results

4.2 Clustering content

Figure 6. Pearson correlation of student skills and learning content, and rankings per user group

Clustering of content was tested based on the ranking of students of the contents of each micro-learning session. This component of the similarity index was verified by computing the Pearson correlation between learning content alternatives and the user profiles. Pearson correlation together with the rankings made by the students is shown in Table 1 for one micro-learning session. The Pearson correlation helped to explore how well a particular micro-learning session suits different user profiles. Correlation values between -0,5 and 0,5 show that the learning content is equally suitable or unsuitable for each user group. In the range of 0,5 and 1,0 the correlation values show that the content is more suitable for expert users, while in the range of -0,5-1,0 it indicates that the content is better fitting beginners.

4.3 Exploring probability value of recommendations

Name of micro-learning-session	Beginners	SD	Mid	SD	Advanced	SD	Expert	SD
Data Types	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Variables	0,90	0,00	0,60	0,01	0,50	0,17	0,40	0,02
Control statements	0,80	0,02	0,60	0,19	0,60	0,07	0,50	0,05
Methods	0,70	0,07	0,80	0,05	0,80	0,20	0,50	0,05
Objects vs Classes	0,40	0,04	0,90	0,12	0,70	0,08	0,70	0,13
Inheritance	0,40	0,04	0,80	0,18	0,70	0,10	0,80	0,13
Android Studio Environment	0,80	0,20	0,90	0,20	0,70	0,11	0,60	0,02
Android Manifest	0,40	0,03	0,80	0,15	0,80	0,04	0,90	0,07
XML	0,80	0,16	0,80	0,19	0,70	0,18	0,70	0,07
GUI elements	0,90	0,13	0,90	0,08	0,80	0,04	0,80	0,04
Activity and Fragment	0,80	0,09	0,70	0,14	0,80	0,11	0,90	0,12
Images (ImageView)	0,80	0,19	0,70	0,07	0,70	0,12	0,80	0,06
Menu	0,80	0,14	0,80	0,20	0,60	0,07	0,50	0,10
Linear/Relative layout	0,70	0,12	0,50	0,19	0,50	0,00	0,50	0,05
ListView	0,30	0,06	0,60	0,03	0,60	0,00	0,70	0,16
OnClick Listener	0,40	0,10	0,50	0,01	0,60	0,19	0,80	0,19
Strings	0,50	0,20	0,60	0,07	0,60	0,20	0,70	0,07
Array vs ArrayList	0,60	0,11	0,70	0,05	0,80	0,14	0,70	0,14
Arithmetic Operations	0,80	0,17	0,40	0,11	0,50	0,00	0,50	0,20
UML	0,60	0,17	0,60	0,10	0,70	0,12	0,80	0,06
Sensors	0,70	0,12	0,60	0,03	0,60	0,01	0,70	0,00
SQLite	0,20	0,19	0,40	0,09	0,50	0,12	0,70	0,16
Intents	0,50	0,01	0,60	0,20	0,80	0,09	0,70	0,14
Debugging	0,60	0,11	0,80	0,13	0,70	0,19	0,70	0,02
Errors	0,40	0,07	0,60	0,13	0,50	0,09	0,80	0,20
Exceptions	0,40	0,16	0,50	0,19	0,60	0,01	0,80	0,04

Figure 7. Probability of recommendation per user profile. The

The current content of the course was evaluated based on the set of equations defined in section 3. Table 1 summarizes the probability of recommendations for each micro learning session per user profile. The probability of recommendation was computed for each user and the results were analysed by descriptive statistics. Table 1 contains the mean and the standard deviation of the probability of recommendation for each user profile. The results in general show that approximately 40 percent of the total number of micro-learning sessions has the same suitability for each user profile, while the rest of the micro-learning sessions are not fitting some user profiles. This implies that the course could benefit from personalized learning tracks supported by the proposed recommendation mechanism.

From our analysis, it can also be concluded that difference between mid and advanced level programmers is not significant, therefore a distinction between these user profile groups is not meaningful. Table 1 also presents a substantial number of micro-learning sessions with low probability of recommendations. This can be caused by the following reasons: (1) the users did not like the content of the session and gave low rankings, (2) there is no consensus among the user group of liking or disliking a session and the data presents large variation as in the case of the session about "Strings", (3) the position of the micro-learning session is not logical or ideal as a follow up of previous session.
The third finding of our analysis is that some of the course contents for expert users may be too easy, which results in high performance in the online tests, but low ranking of the learning material. This is mainly visible in the first three micro-learning sessions.

4.4 Implementation and plans for further studies

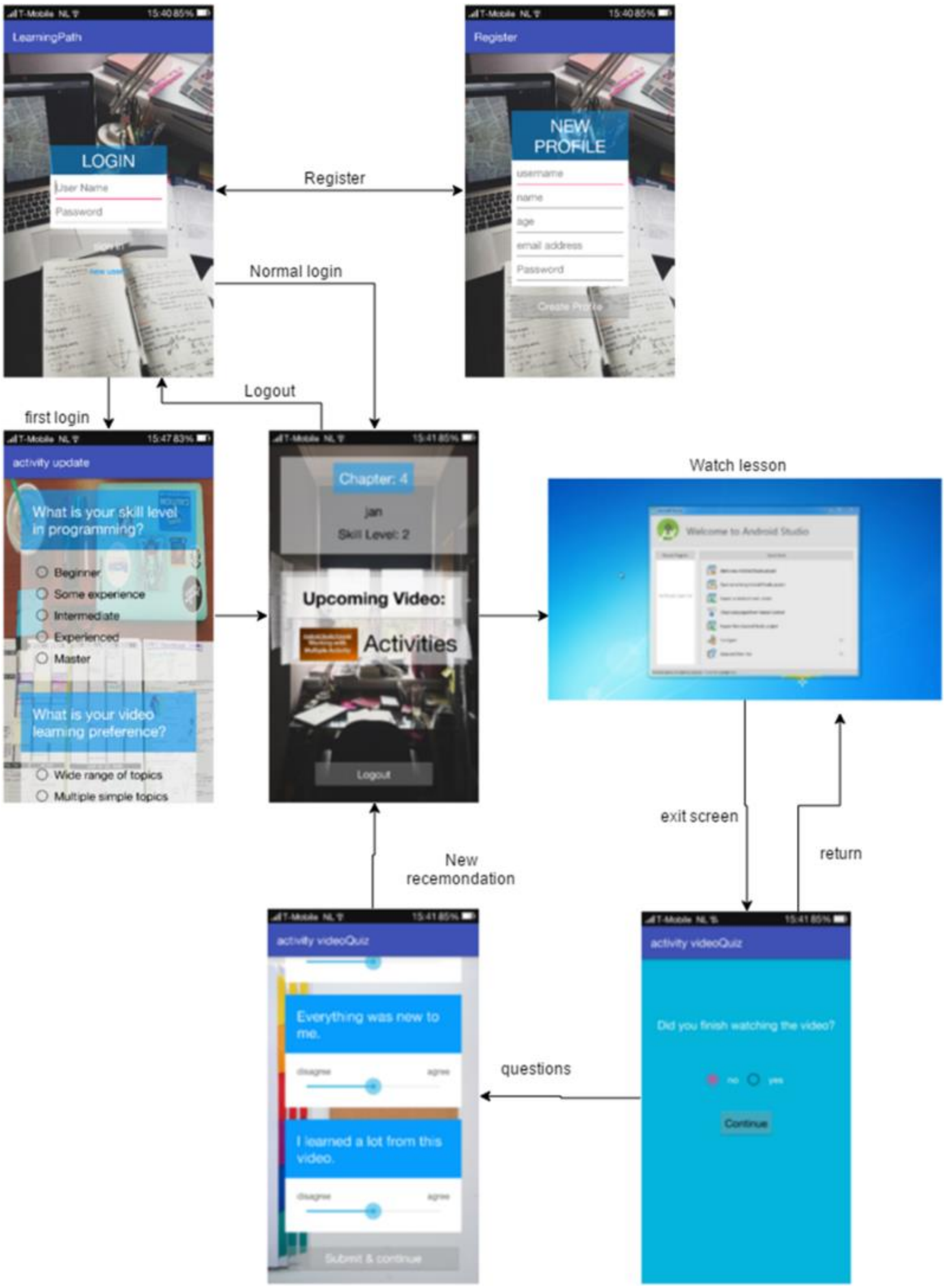

Figure 8. Implementation of recommendation mechanism as an Android app

Making this course personalizable had many challenges. Collecting the initial material for the micro-learning sessions was relatively easy as the topic of Android programming is well documented in the form of online courses. The segmentation of the learning content into 4-5 minutes videos and development of short tests are, however, very a labour intensive and time consuming exercise. The course content for the first time was offered to the students in their standard e-learning environment, i.e. blackboard. The implementation of the recommendation mechanism was, however, not possible in blackboard, therefore, the recommendation mechanism was implemented as an Android application. Figure 3 shows the app developed by students of the Software course. The app will be tested in the upcoming course next year, in order to validate and fine tune the recommendation mechanism proposed in this paper.

5 CONCLUSIONS

The paper presented a new approach for teaching programming to industrial design students considering the difference in their background knowledge, experience and pace of learning. The course content was decomposed into micro-learning sessions consisting of the description of learning goals, multimedia material explaining the learning content, and a short online-test. A content recommendation mechanism was developed that proposes the follow up micro-learning sessions to the students taking into account the profile of learners, their progress of learning, and their preferences. The novel aspect of this recommendation mechanism that it extends the content and collaborative recommendations with content dependency enabling development of learning paths for the course. The concept of the recommendation mechanism has been verified based on data collected in a former course. Analysis of the collected data suggests that (i) the questionnaire on user profiles and the similarity index proposed in this paper is able to distinguish the clusters of students (the clusters identified for our course were beginner, mid-level, advanced, and expert programmers), (ii) suitability of some of the micro-learning sessions for particular clusters of students exposable by determining the Pearson correlation between the learners profiles and their ranking of the micro-learning sessions (iii) the approximately 60% of the total number of micro-learning learning sessions of the analysed course content should be personalized to address the specific needs of the learners clusters. Our further research aims to deploy the developed Android application in the future course in order to validate the recommendation mechanism.

REFERENCES

Brady, L. (2004), "Towards optimal student engagement in teacher education". *Australian Journal of Teacher Education*, 29(2). Retrieved August 1, 2014, from http://ro.ecu.edu.au/ajte/vol29/iss2/3/.

Harri-Augstein, E. S., and Thomas, L. F. (2005), "The Kelly repertory grid as a vehicle for eliciting a personal taxonomy of purposes for reading". *Journal of Research in Reading*, 1(1), 53–66.

Krynicki, K., Jaen, J., and Navarro, E. (2016), "An ACO-based personalized learning technique in support of people with acquired brain injury". *Applied Soft Computing*, 47(6), 316–331. Liu, et al., (2015)

Lobo, L.M., Sunita, R.J., Aher, B., "Mining association rule in classified data for course recommender system in e-learning", *Int J Comput Appl*, 39 (7) (2012), pp. 1–7 (Yang and Wu. 2009)

Markovi, S., et al. "Adaptive distance learning and testing system Computer Applications in Engineering Education", Volume 21, Issue S1. *Computer Applications in Engineering Education*, 2013. 21, E2-E13.

Mehdi Elahi, Francesco Ricci, Neil Rubens, "A survey of active learning in collaborative filtering recommender systems", *Computer Science Review*, Volume 20, May 2016, Pages 29-50, ISSN 1574-0137,

Nilashi, M., bin Ibrahim, O., and Ithnin, N. (2014), "Hybrid recommendation approaches for multi-criteria collaborative filtering", *Expert Systems with Applications*, 41(8), 3879-3900.

Reategui, E., Boff, E., and Campbell, J. A. (2008), "Personalization in an interactive learning environment through a virtual character". *Computers & Education*, 51, 530–544.

Tzouveli, P., Mylonas, P., and Kollias, S. (2008), "An intelligent e-learning system based on learner profiling and learning resources adaptation". *Computers & Education*, 51, 224–238

Yang, Y.J., and Wu, C. (2009), "An attribute-based ant colony system for adaptive learning object recommendation". *Expert Systems with Applications*, 36(2), 3034-3047.