Comparing representations for learner models in interactive simulations

Cristina Conati, Lauren Fratamico, Samad Kardan, Ido Roll[1]

University of British Columbia, Vancouver, Canada, V6T1Z4

{conati, fratamic, skardan}@cs.ubc.ca, ido.roll@ubc.ca

Abstract. Providing adaptive support in Exploratory Learning Environments is necessary but challenging due to the unstructured nature of interactions. This is especially the case for complex simulations such as the DC Circuit Construction Kit used in this work. To deal with this complexity, we evaluate alternative representations that capture different levels of detail in student interactions. Our results show that these representations can be effectively used in the user modeling framework proposed in [2], including behavior discovery and user classification, for student assessment and providing real-time support. We discuss trade-offs between high and low levels of detail in the tested interaction representations in terms of their ability to evaluate learning and inform feedback.

Keywords: Educational Data Mining, Clustering, User Modeling, Interactive Simulations, Exploratory Learning Environments

1 Introduction

Interactive simulations are educational tools that can foster student-driven, exploratory learning by allowing students to proactively experiment with concrete examples of concepts and processes they have learned in theory. There is increasing research in Intelligent Tutoring System (ITS) to endow these interactive simulations and other types of Exploratory Learning Environments (ELE from now on) with the ability to provide student-adaptive support for those students who may not learn effectively from these rather unstructured, open-ended activities [2–5]. Providing this support entails building a user-model that can estimate the learner's proficiency in learning via exploration and need for help during interaction. However, building such a model is especially challenging because it is relatively unclear how to operationalize exploration skills and difficult to define a priori which behaviors are conductive to learning.

Some previous work has dealt with the challenge by limiting the exploratory nature of the interaction [4, 6]. In contrast, Kardan and Conati [2] proposed a student modeling framework that learns from action logs which student behaviors should trigger help during interaction with an ELE. *Clustering* is used to identify students who be-

[1] All authors have contributed equally to this work and are listed alphabetically.

adfa, p. 1, 2011.

have and learn similarly from the interaction. Asso*ciation rule mining* is applied to derive distinguishing interaction behaviors from the clusters, and these behaviors are leveraged to drive the provision of adaptive support in real time during interaction. This student modeling framework was successfully applied to provide adaptive support in the CSP applet, an ELE for a constraint satisfaction algorithm [7]. The part of the CSP applet used in [7] involves a limited number of actions and thus it was sufficient to represent student behaviors in terms of raw actions. This simple representation did not scale up when we tried to apply the framework to a more complex simulation that provides over a hundred types of actions for exploring concepts related to electricity, the PhET DC Circuit Construction Kit (CCK). Thus, in [5] we proposed a richer, multi-layer representation of *action-events* that includes information on individual actions (e.g., join), as well as the manipulated components (e.g., light bulbs), the relevant family of actions (e.g., revise), and the observed outcome (e.g., changes to light intensity). We showed that clustering interaction behaviors based on this representation succeeds in identifying students with different learning outcomes in CCK.

In this paper, we provide a comprehensive evaluation of this multi-layer representation as the basis to apply the student modeling framework proposed in [2] to CCK. The evaluation is in terms of ability to *identify learners* with high- or low- learning gains, suitability for *user modeling* (i.e. to classify new students in terms of their learning performance as they work with CCK), and for *defining the content of adaptive support* during interaction. Furthermore, we use these evaluation dimensions to compare alternative representations derived from the multi-layer structure in [5], which capture different aspects of interaction behaviors at different levels of granularity. Our results show both classification accuracies comparable to those reported in [7], as well as that the approach succeeds in discovering association rules that can be leveraged to design interactive support, thus providing evidence on the generality of this student modeling framework across representations. We further discuss tradeoffs between evaluation dimensions that need to be considered when choosing the most suitable representation for assessing and supporting students during interaction with ELE as complex as CCK.

In the rest of the paper we first discuss related work. Then, we describe the CCK simulation and the study used for collecting data. Next, we present the different representations we evaluated, and summarize the user modeling approach we used. After presenting the evaluation results, we conclude with a general discussion of findings, contributions, limitations, and future work.

2 Related work

Most of the work done so far on providing adaptive feedback in interactive simulations has dealt with the challenge of how to identify when and how a student needs support by limiting the exploratory nature of the interaction. For instance, the simulations developed by Hussain et al. [8] provide feedback on how to behave in pre-defined cultural/language-related scenarios with clear definition of correct answers/behaviors. The Chemistry VLab [9] provides help on well-defined steps required to run a scientific experiment. Science ASSISTments [4] provides feedback on the specific problem of controlling for variables in experimental design. Work on designing adaptive support for more open-ended exploratory interactions has relied either on expert knowledge (e.g., [10] and [3]) or on data-mining (e.g. [7] and [11]) to identify suitable feedback strategies. The work in [11], which provides scaffolding to students using an environment that supports learning by teaching an artificial student, relies on knowing a priori which students learned or not from the system to mine the relevant feedback strategies. In contrast, the approach successfully evaluated in [7], and adopted in this paper, groups learners via clustering on their interaction behaviors alone (with little processing), without using additional information.

3 The CCK simulation and User Study

The CCK simulation is part of PhET [12], a freely-available and widely-used suite of simulations in different science and math topics. CCK includes 124 different types of actions to build and test DC circuits by connecting different components including wires, light bulbs, resistors, batteries, and measurement instruments (Figure 1). The available actions include adding, moving, joining, splitting, and removing components, as well as changing the attributes of components (such as resistance). Additional actions relate to the interface (such as changing views) or the simulation itself (such as resetting the simulation). CCK provides animated responses with regard to the state of the circuits on the testbed. For instance, when a light bulb is connected to the circuit, the light intensity and speed of electrons change with variation of the current.

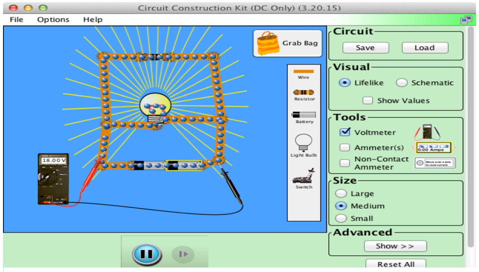

Figure 1. The CCK simulation.

CCK is a tool and instructors define activities outside the environment. Being an inquiry environment, not all students make optimal use of the simulation. Our long-term goal is to assess the effectiveness of students' behaviors in CCK and provide explicit support to foster learning.

Data used in this paper was collected from 96 first-year physics students who participated in a laboratory user study described in [5, 13] and who had less-than-perfect pre-test scores. In the study, participants completed two 25-minute activities. The first activity, on the topic of light bulbs, had different conditions of external scaffolding (using tables and prompts). In contrast, the second activity, on the topic of resistors, was identical for all learners, and included only minimal guidance. Thus, here we focus on data only from the second activity. Students were told to “investigate how resistors affect the behaviors of circuits” and were given advice to combine resistors with different resistances. The students were expected to use CCK to help them explore this learning goal. As this was their second activity with the simulation, all students were proficient with the testbed. Students were assessed on their conceptual knowledge before and after the activities, with the pre-test being a subset of the post-test. To avoid priming students, the pre-test only included questions not related to specific circuits (e.g., comparing the current in two different resistors with different resistances), whereas the post-test also included questions on specific circuit diagrams.

4 Representing the User Actions

Clustering students based on their actions requires a representation that captures important aspects of these actions. However, the large variety of actions available in CCK, together with their contextual nature, makes clustering challenging. Notably, action outcomes depend on the state of the simulation. For example, connecting a wire leads to different outcomes based on the state of the circuit (e.g., existence of a battery) and testing instruments (e.g., how they are connected). The CCK logs information on the type of action, the component used, and the response of the physical model. In addition, actions with one component often affect other components (e.g., changes to batteries affect existing light bulbs).

As described in [5], we created a structured representation that can capture these "action-events" – i.e., user actions and their relevant contextual information - at different levels of granularity. The structure contains four layers: "actions" describe the action that students took, e.g., *join* (25 types). "Components", describe the manipulated component, e.g., *wire* (22 different types). "Family" denotes the general type of action, and there are 8 families in the structure. Common families include: *Build* (describes actions such as adding, removing, and joining components, before the circuit is live), *Test* (describes actions with the measurement instruments), *Organize* (describe actions that re-arrange circuit components without making any structural changes), and *Revise* (describe all build actions that take place on a working circuit). Finally, "outcomes" capture what happens in the circuit after an action is performed. There are 6 types of outcomes, including: *None*, *Deliberate-measure* (the value displayed on a measurement device is updated as a result of using it), *Current-change* (a change in the current reflected in the speed of movement of electrons), and *Light-intensity-change* (the light intensity of a light bulb changes). One action-event may cause more than one outcome. By creating this structure we have added contextual information to the data. For example the action-event current_change.revise.join.wire describes *joining* (action) a *wire* (component) that led to a *current-change* (outcome) when *revising* a circuit (family). In addition, we captured "pauses" longer than 15 seconds as an additional family of actions, with a single type of (in)action.

While in [5] all 4 layers of the structure were used to represent actions-events, subsets of the layers can represent events at different levels of granularity. In turn, each representation can be used to generate different feature sets based on the types of measures used to summarize the action-events for each user. These measures include: (i) frequency of the action-event, i.e., the proportion of each type of action-event over total action-events (denoted by *_f*); (ii) mean; and (iii) standard deviation of the time spent before each action-event.

In [5], we described the performance of the action-event feature set using all 4 layers when used to cluster students who learn similarly with CCK. Here, we investigate the effect of levels of granularity in feature set representation on both generating meaningful clusters, as well as on building effective user models and informing feedback, as in [7]. Thus, we generated feature sets that use different subsets of layers in the action-event structure. For each representation, we also experimented with using only frequencies vs. adding time-related summative measures, for a total of 22 different feature sets

We also tested a feature set that goes beyond actions as units of operation by grouping consecutive actions of the same type into entities called a *block*. We have 6 different blocks, including *Test* (all actions related to using measurement devices), *Construct* (any action that changes the circuit before testing), *Modify* (any action that changes the circuit after testing it), and *Reset* (removing the whole circuit). Each block has two kinds of features: summative features about the block (frequency, average duration and average number of actions within), and specific features about each outcome within the block (for instance, frequency of light-intensity-changes within a construct block).

Of the 23 feature sets described above, only 3 generated meaningful clusters that group students in terms of their learning:

1) *OFAC_f*: Set including all action-events elements (Outcome, Family, Action, Component) with frequency information (210 features)
2) *FAC_f*: Same as the first set, but without Outcome. (202 features)
3) OAC_f: Same as the first feature set, but without the Family layer (90 features).

It should be noted that OAC_f is a feature set that requires less feature engineering, as all the three layers included (outcomes, actions and components) were available in the log files with only minor modifications (e.g., calculation of pauses). The Family layer included in the other two feature sets, on the other hand, was defined via extensive discussion among the authors in terms of how best to conceptualize the various actions available in CCK.

Interestingly, all three feature sets include only information on action frequency, indicating that summative statistics capturing how much time student spend before actions are not contributing to identify different learning outcomes. This can be explained by the fact that we capture significant pauses before actions via a specific action and family (Pauses). Alternatively, there may be important timing information over sequences of actions (e.g., planning a certain circuit or running a series of tests), but not in individual actions [9].

5 Evaluating Representations for Assessment and Support

We applied the user modeling framework for ELE, first proposed in [2], to evaluate how well the three feature sets identified above support building user models. The framework consists of two main phases: *Behavior Discovery* and *User Classification*.

In *Behavior Discovery,* each user's interaction data is first pre-processed into feature vectors. Students are then clustered using these vectors in order to identify users with similar interaction behaviors. The resulting clusters are then analyzed to see whether they identify groups of students with different learning outcomes. If they are, the distinctive interaction behaviors in each cluster are identified via association rule mining. This process extracts the common behavior patterns in terms of class association rules in the form of X → *c*, where X is a set of feature-value pairs and *c* is the predicted class label for the data points where X applies. During the association rule mining process, the values of features are discretized into bins [2].

In *User Classification*, the labeled clusters and the corresponding association rules extracted in Behavior Discovery are used to train a classifier student model. As new users interact with the system, they would be classified in real-time into one of the identified clusters, based on a membership score that summarizes how well the user's behaviors match the association rules for each cluster. Thus, in addition to classifying students in terms of learning, this phase returns the specific association rules describing the learner's behaviors that caused the classification. These behaviors can then be used to trigger real-time interventions designed to encourage productive behaviors and discourage detrimental ones, as described in [7].

Based on this framework, the three measures we use to evaluate the feature sets described in the previous section are: (i) *Quality of the generated clusters*, measured by effect size of difference in learning performance between students in the different clusters. (ii) *Classification accuracy* of user models trained on the obtained clusters. (iii) *Usefulness* of the generated association rules in identifying behavior patterns that can be used to design and trigger support to students.

6 Results

6.1 Quality of the clusters

Table 1 shows the outcome of clustering on the three feature sets. Each row describes one cluster in the optimal number of clusters for that representation. Clusters are named based on their learning performance. The table also reports cluster size (after removing clustering outliers) and the average learning performance of a cluster's members (measured as corrected post-test scores). The last two columns report the p-value and effect size of the difference in learning performance between clusters, obtained via an ANCOVA on the post-test scores, controlling for pre-test. Thus, a larger effect-size suggests a representation that better separates students with different learning levels.

Feature Sets	Cluster	#Members	Average Corrected Post-test	p-value	Effect Size (partial eta squared)
FAC_f	High	67	.596	.048	.041
	Low	29	.534		
OAC_f	High	66	.609	.007	**.076**
	Low	30	.509		
OFAC_f	High	61	.613	.013	.065
	Low	35	.516		

Table 1. Summary statistics for the clustering results

All feature sets generated two clusters, identifying groups of students with high vs. low learning. Effect sizes of the difference in learning performance varied for different feature sets, ranging from small effect size (for *FAC_f*) to medium-small effect (for *OAC_f* and *OFAC_f*). Interestingly, *OAC_f* achieves the highest effect size, showing that the addition of more feature-engineered information (the Family) reduced the differences in learning between the two clusters.

6.2 Classification accuracy

For each of the three feature sets, a classifier user model is trained on the generated clusters, using 8-fold nested cross validation to set the model's parameters and find its cross-validated accuracy[2]. Table 2 reports classification performance of each classifier in terms of overall accuracy, class accuracy for high and low learners, and kappa scores. The classifiers achieved moderate-to-good kappa values between 0.56 and 0.7. All accuracies are significantly above the baseline, indicating that our user-modeling framework can effectively classify students working with CCK with all three feature-sets. The feature set based on the most detailed representation, *OFAC_f*, is superior to the other 2 sets on all accuracy measures, including being the most balanced classifier. This indicates that the additional level of representation added by the Family level is beneficial for classifier accuracy when all information (action, outcome, component) is leveraged. Also, both feature sets that include Outcome show higher accuracy compared with *FAC_f*, suggesting that the outcome of students' actions, rather than the actions themselves, are most beneficial to identify low vs. high learners.

Feature Sets	Baseline	Overall Accuracy % (Std. dev.)	High Learner Class Accuracy	Low Learner Class Accuracy	Kappa
FAC_f	.698	83.3 (5.9)	.851	.724	.564
OAC_f	.688	84.4 (9.4)	.909	.700	.626
OFAC_f	.653	86.5 (8.8)	**.918**	**.771**	**.702**

Table 2. Classifier accuracy measures for different feature sets. Baseline is the accuracy of the most likely classifier.

6.3 Usefulness for providing adaptive support

Association rules identify behavioral patterns that are representative of what students in a given cluster do with CCK (see [14] for a discussion of how patterns are derived from rules). These patterns are useful if they are associated with low (or high) learning performance that can inform adaptive interventions. Specifically, if a student is classified as a "Low Learner" (LL) at any given point of working with CCK, adaptive interventions can be provided to discourage the LL patterns he is showing and to encourage the HL patterns he is not showing. The number of identified patterns varies greatly among feature sets, ranging from 15 in *OAC_f* to 17 in *FAC_f* to 23 in *OFAC_f*, showing that the most complex representation captures finer grained variations in learner behaviors. Example patterns are shown in Table 3.

While the patterns produced by all three feature sets varied, we identified 4 trends that occurred in at least two feature sets each. This shows that our general approach for behavior discovery is able to uncover core behaviors that are stable across repre-

[2] The accuracies reported are calculated at the end of the interaction, which presents an upper bound for the accuracy of the model during the interaction.

sentations. One of these trends is related to addition of light bulbs and changes in light intensity. High Leaners (HL) both add light bulbs infrequently and make infrequent changes in light intensity. Since this activity was focused on understanding how resistors work in circuits, light bulbs were likely distractors at best, and possibly interfered with observing the behavior of other resistors. We see the adding light bulb behavior associated with HL in both the *FAC_f* feature set (*Build.add.lightBulb_f = Low*) as well as in the *OFAC_f* feature set further qualified with the outcome (None.*Build.add.lightBulb_f = Low),* as shown in Table 3. We also see HL making changes to light intensity with low frequency in both *OFAC_f* (*light_intensity.Revise.split.junction_f = Low*) and *OAC_f* (*light_intensity.join.wire_f = Low).* The other trends across feature sets show that high leaners do the following actions more frequently: i) use testing devices (to examine the circuit configuration), ii) change the resistance of resistors (possibly to experiment with a range of resistors, as suggested by the activity), and iii) pause (possibly to plan, reflect, and take notes). The first two are intuitively effective behaviors for understanding how resistors work in a circuit. The last one is an indication of learners taking time to best leverage the learning activity.

Feature Sets	Clus-ter	Pattern [Description]
FAC_f	HL	Build.add.lightBulb_f = Low [When building, they added light bulbs with low frequency]
	LL	Build.changeResistance.resistor_f = Low [When building, they changed the resistance of resistors with low frequency]
OAC_f	HL	light_intensity.join.wire_f = Low [They joined wires resulting in light intensity change with lower frequency]
	LL	deliberate_measure.traceMeasure.nonContactAmmeter_f = Low [They used the non contact ammeter by tracing with low frequency]
OFAC_f	HL	None.Build.add.lightBulb_f = Low [When building, they added light bulbs resulting in no outcome with low frequency]
	LL	deliberate_measure.Test.startMeasure.voltmeter_f = Low [When testing, they used the voltmeter with low frequency]

Table 3. Sample patterns for each feature set (raw form and English description)

Next we evaluate the usefulness of these patterns to inform support. One criterion for doing so is level of detail. Naturally, this depends on the granularity of the corresponding feature set in the different representations. Thus, behaviors in *OFAC_f* give the most contextual information about timing and can be used to give students feedback with regard to the *outcome* of desired actions, *what* to do to achieve that outcome in terms of a high level behavior, and *how* to achieve it using specific action and component. For example, a hint based on the pattern in Table 3 for LL in *OFAC_f* could suggest students to do more deliberate measurements (*outcome)*, achieve this by testing more (*what* to do), and, if necessary, give an even more specific suggestion to use the voltmeter (*how*). In contrast, both of the other two feature sets cannot give one of those layers of hints. *OAC_f* can only tell students the *outcome* of what they need to do and the specifics of *how* to do it, but a more general level of information is missing. For example, based on the LL rule for OAC_f, students can be told to trace with the non-contact ammeter more often, but there is no general “test more” hint. *FAC_f* can only tell students *what* to do and the specifics of *how* to do it, but cannot tell them the *outcome* to achieve. For example, students can be told to change the resistance of their resistors more often, but without emphasizing the desired outcomes.

The richer level of detail available due to the nature of the *OFAC_f* representation lends itself well to provide sequences of hints with narrowing specificity (a well-established approach to hint provision in ITS). For instance, a first level of hint could tell the student the *outcome* that they should try to achieve, then, if needed a second level of hint could suggest the family (*what* to do at the high level), followed by a hint on *how* to do it. The *OAC_f* and *FAC_f* feature sets do not support this hint progression, though missing levels could be inferred. For example, if the detailed hint suggests to trace more with the non-contact ammeter, a hint could still first suggest general testing.

7 Discussion and Conclusion

To summarize our results, we found the *OAC_f* feature set to be the best for identifying high versus low learners. This feature set does not include the knowledge-engineered Family layer. However, it was the feature set based on the most complex representation, *OFAC_f,* that scored highest in terms of classifier accuracy. It was also the set that identified the largest number of behavioral patterns, 23, and that can provide richer levels of feedback to the students. In summary, our comparison of representations that differ in the level of granularity has identified a trade-off between suitability to provide support and quality of the clusters: hints generated by the most complex representation in *OFAC_f* would target the right students due to a high classification accuracy, can give detailed support, and can provide the largest number of hints. On the other hand, the representation with the least amount of feature engineering, *OAC_f,* generates rules that come from higher quality clusters, albeit offers fewer hints, with fewer levels of support. These hints may also be given inappropriately due to lower model accuracy. An experimental evaluation is required to see how this tradeoff impacts the effectiveness of interventions in an adaptive version of CCK. Thus, generating different adaptive versions of CCK based on the classifiers and behavior patterns identified in this paper is one of the next steps of this research.

More importantly, the results in this paper provides evidence on the generality of the user-modeling framework we used for our evaluation. This framework had already been successfully applied for modeling students and providing support in a rather simple simulation for an AI algorithm [7]. Here we show that it can transfer to more complex ELE such as CCK, at least in terms of successfully classifying student learning at the end of the interaction (all classifiers discussed in this paper achieved respectable kappa values, higher than 0.55) and identifying interaction behaviors intuitively associated with more/less effective learning. One of the next steps of this research is investigating how to design real-time hints that can foster the productive patterns and discourage the others as we did in [7]. This includes investigating the overtime accuracy of the classifier user model. Another step of future work is to further test the generality of this modeling framework by applying it to another simulation of the PhET family. This will allow us to identify productive patterns across simulations and domains and bring us closer to addressing the challenge of a general modeling framework for interactive simulations.

8 References

1. Perera, D., Kay, J., Koprinska, I., Yacef, K., Zaiane, O.R.: Clustering and Sequential Pattern Mining of Online Collaborative Learning Data. IEEE Transactions on Knowledge and Data Engineering. 21, 759 –772 (2009).
2. Kardan, S., Conati, C.: A Framework for Capturing Distinguishing User Interaction Behaviours in Novel Interfaces. Proc. of the 4th Int. Conf. on Educational Data Mining. pp. 159-168. , Eindhoven, the Netherlands (2011).
3. Mavrikis, M., Gutierrez-Santos, S., Geraniou, E., Noss, R.: Design requirements, student perception indicators and validation metrics for intelligent exploratory learning environments. Personal and Ubiquitous Computing. 1–16.
4. Gobert, J.D., Pedro, M.A.S., Baker, R.S.J. d, Toto, E., Montalvo, O.: Leveraging Educational Data Mining for Real-time Performance Assessment of Scientific Inquiry Skills within Microworlds. JEDM - Journal of Educational Data Mining. 4, 111–143 (2012).
5. Kardan, S., Roll, I., Conati, C.: The Usefulness of Log Based Clustering in a Complex Simulation Environment. In Intelligent Tutoring Systems. pp. 168–177. Springer (2014).
6. Westerfield, G., Mitrovic, A., Billinghurst, M.: Intelligent Augmented Reality Training for Assembly Tasks. In Artificial Intelligence in Education. pp. 542–551. Springer (2013).
7. Kardan, S., Conati, C.: Providing Adaptive Support in an Interactive Simulation for Learning: an Experimental Evaluation. Proceedings of CHI 2015 (To appear).
8. Hussain, T.S., Roberts, B., Menaker, E.S., Coleman, S.L., Pounds, K., Bowers, C., Cannon-Bowers, J.A., Murphy, C., Koenig, A., Wainess, R., others: Designing and developing effective training games for the US Navy. The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC). NTSA (2009).
9. Borek, A., McLaren, B., Karabinos, M., Yaron, D.: How Much Assistance Is Helpful to Students in Discovery Learning? In Learning in the Synergy of Multiple Disciplines 4th European Conference on Technology Enhanced Learning. pp. 391–404. Springer (2009).
10. Roll, I., Aleven, V., Koedinger, K.R.: The Invention Lab: Using a Hybrid of Model Tracing and Constraint-Based Modeling to Offer Intelligent Support in Inquiry Environments. In Intelligent Tutoring Systems. pp. 115–124. Springer (2010).
11. Leelawong, K., Biswas, G.: Designing Learning by Teaching Agents: The Betty's Brain System. International Journal of Artificial Intelligence in Education. 18, 181–208 (2008).
12. Wieman, C.E., Adams, W.K., Perkins, K.K.: PhET: Simulations That Enhance Learning. Science. 322, 682–683 (2008).
13. Roll, I., Yee, N., Cervantes, A.: Not a magic bullet: the effect of scaffolding on knowledge and attitudes in online simulations. In Proc. of Int. Conf. of the Learning Sciences. pp. 879–886 (2014).
14. Kardan, S., Conati, C.: Evaluation of a Data Mining Approach to Providing Adaptive Support in an Open-Ended Learning Environment: A Pilot Study. AIED 2013 Workshops Proceedings Volume 2. pp. 41–48 (2013).