# TutorGen SCALE® - Student Centered Adaptive Learning Engine

**Ted Carmichael, PhD**
TutorGen, Inc.
ted@tutorgen.com

**Mary Jean Blink**
TutorGen, Inc.
mjblink@tutorgen.com

**John Stamper, PhD**
Carnegie Mellon University, TutorGen, Inc.
john@stamper.org

**ABSTRACT**: TutorGen's Student Centered Adaptive Learning Engine (SCALE®) represents a breakthrough in developing adaptive educational systems by using big data collected from new or existing educational software systems to automatically generate intelligent tutoring capabilities. This work aims to make adaptive learning widely available; to give students real-time, useful feedback; and to provide tools to teachers for assessing student performance. SCALE does this by collecting student data from new or existing digital learning systems and then automatically generating adaptive capabilities based on this data. In this way SCALE is able to efficiently turn any edtech product into an intelligent tutoring system, with very little need for new software customization or expensive and time-intensive manual input. Adaptive learning has long been proven to decrease time to learn and increase retention and understanding for students, but has not been widely adopted due to the high cost of implementation. SCALE solves this challenge with a human-centered, data-driven approach by using Artificial Intelligence and machine learning techniques to generate adaptability in a way that is content and system agnostic. Here we report on our approach for creating and implementing SCALE, and the refinements created to bring this technology from the research lab into the classroom.

**Keywords**: ITS, Intelligent Tutoring System, EDM, Educational Data Mining, Artificial Intelligence, Big Data, Adaptive Learning, Teacher dashboard, edtech.

## 1 INTRODUCTION

TutorGen's Student Centered Adaptive Learning Engine (SCALE®) represents a breakthrough in developing adaptive educational systems by using big data collected from new or existing educational software systems to automatically generate intelligent tutoring capabilities. SCALE collects data from existing computer/web based training software, and uses educational data mining and artificial intelligence techniques to automatically generate student models. SCALE improves these models over time as more data is collected, and tracks student progress on specific concepts or skills (knowledge tracing). This allows for easy assessment at any point in time. The system also dynamically selects the next best problem to maximize student learning and minimize time needed to master a set of skills (problem selection). For complex multi-step problems, SCALE can provide context specific, just-in-time hints. SCALE also provides data adapters so edtech developers can easily hook into the SCALE system using Web APIs. Finally, a main differentiator of our system is our transparent process of data curation and the related visual tools that expose the workings of the problem and student-model generation process. **By building on award-winning learning science and data mining research, we have designed and developed a system that makes adding adaptive capabilities to existing systems easy and affordable.**

SCALE grew out of a recognized need in the marketplace, for automatic adaptability that can be added to any edtech product, without having to custom design and implement a non-generalizable solution. We found that many companies are designing new edtech delivery and content products, and new platforms, but that adaptability is often missing, or only done in a cursory way. Yet methods for using Artificial Intelligence (AI) and machine learning to generate adaptability in a way that is content and system agnostic have already been proven in university research labs, including our own work [e.g., 1-4]. This work has been supported by multiple grants from the NSF and the Commonwealth of Kentucky, to get the SCALE technology out of the lab and into the hands of students, teachers, administrators, and developers of edtech products.

## 2 BUILDING THE FOUNDATION FOR SCALE

Following extensive discussions with potential customers, we identified the features necessary to make the TutorGen SCALE system a success. The key innovation of SCALE is the automatic generation of adaptive learning capabilities. These include: 1) Problem selection 2) Skill tracking (and skill modeling) 3) Hints and feedback on multi-step problems 4) Assessment of student learning.

In addition to the ability to automatically generate these features, SCALE also includes a "feedback loop" to continue the improvement of the features over time as more data is collected. Several potential customers have noted that existing providers of adaptive student learning software do not have any way to explain why the system behaves as it does. In contrast, SCALE provides tools that let the instructors and developers explore the data using meaningful visualizations that will provide insights into student learning. In order to achieve this vision we identified the following technical objectives to be completed during the NSF SBIR Phase I and Phase II projects for SCALE:

1. Research and implement overall data and system architecture
2. Research, design, and implement knowledge tracing algorithms and handler
3. Research, design, and implement the hint and feedback mechanism
4. Design and implement student assessment and support tools for teachers
5. Design and implement integration of support tools for computer-based training systems and content designers/developers
6. Pilot test SCALE in the classroom
7. Perform Full Integration and Load Testing

### 2.1 Data and System Architecture

The high level design of the system can be seen in Figure 1, and the main product is designed around the SCALE Engine, which implements the model discovery and creates the mechanisms for automated and expert feedback. All information collected and generated is stored in a database, which can be accessed by developers and educators through a set of tools. These tools allow for exploring models, improving models, and assessing student performance. A set of universal data connectors provides a simple API for connecting software to the system for data collection through web services. This documented communication method provides an open architecture for existing computer-based training system providers to connect to SCALE in a seamless fashion.

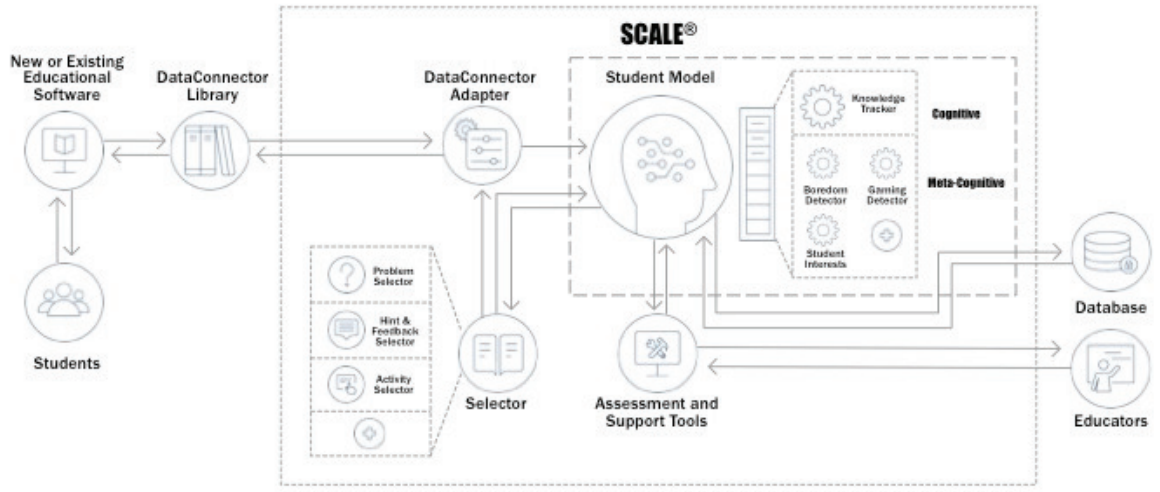

**Fig 1. Design of TutorGen SCALE showing how new or existing educational software can use our data connector libraries to connect to the SCALE system which includes three main components: the knowledge tracker with student models, the hint and feedback handler, and assessment and support tools.**

The database has been designed to be flexible enough to manage data for students, knowledge tracing, hint generation, student models, custom attributes, and transaction log. These capabilities require data to be managed at a very granular level while still providing exceptional performance for real-time processing of tutor data and analytics. This led to the design of a multidimensional data model, allowing for the highest performance. The design of the knowledge storage has extended the common format used in the PSLC DataShop repository (http://pslcdatashop.org) and centers on storing student knowledge in the form of Knowledge Component (KC) models. In these models, a KC represents a piece of trackable knowledge: skills or concepts. There is also a hierarchical component to expert models that is captured, which can be influenced by both the type of knowledge being taught and instruction used. The database design takes into account these hierarchical relationships, and also has the ability to compare models against various metrics.

## 2.2 Knowledge Tracing Algorithms and Handler

Student models have traditionally been developed by domain experts applying manual analysis of course content. Further refinements of these models have utilized Cognitive Task Analysis (CTA). A number of studies have demonstrated how detailed CTA can result in dramatically better instruction [5,6]. CTA methods are useful for creating student models, but they have several limitations. First, CTA is more of an art than a science. Structured interviews, think aloud protocols, and rational analysis are all highly subjective and different analysts may produce very different results. Second, the most successful CTA approaches are heavy in human effort. Structured interviews, think-alouds, or student model simulation all require high level of psychological and subject-area expertise in addition to significant time investments. This is both time-consuming and costly.

Our research into finding expert models is founded on our belief that the expert models are discovered, not created. This means that the correct model exists for a specific domain and set of instruction, and it must be identified in the search space of all possible models. A correct expert model is one that is consistent with actual student behavior. It predicts task difficulty, as well as transfer between instruction and test. Finding the exact model for a set of instruction is difficult because some students may be relying on a flawed model that they have created internally [3]. With this in mind, our system is able to discover the expert model that best describes the data and leads to the most robust learning.

To discover better expert models, we apply human and machine learning techniques and evaluate the resulting model using statistical analyses. This forms the basis of our "data-driven, human-centered" approach. Our core statistical model is based on the Additive Factors Model (AFM), which has been previously implemented in DataShop. AFM is a generalization of the log-linear test model (LLTM) [7]. It is a specific instance of logistic regression, with student-success (0 or 1) as the dependent variable and with independent variable terms for the student, the KC, and the KC-by-opportunity interaction. By clustering student knowledge areas, we have augmented the traditional AFM model, adding additional parameters that weight learning rates for these clusters, which, as more data is collected, makes an impact into optimizing an individual's learning.

We have confirmed that our new technique for model discovery will implement the best features of the studied techniques and result in a human readable model. We start with human generated models, if they exist, and from these existing models we will split or merge KCs in the search of alternative models. Because the resulting models will have a hierarchy from existing labeled KCs, it is much easier for humans to evaluate the results (with the help of our visualization tools). These models can then be evaluated using the AFM statistical model and a variety of metrics to score the models including variants of Akaike information criterion, Bayesian information criterion, and cross validation [8].

## 2.3 Hint and Feedback Handler

The hint and feedback handler is based on the Hint Factory, our novel method of automatically generating context specific, just-in-time (JIT) hints for multi-step problems [10]. The method is designed to be specific, on-demand, and to provide the right help at the right time. In order to deliver hints and feedback, the Hint Factory first constructs a graph of states and actions that represents all previous student approaches to a particular problem. The state-action graph is transformed into a Markov decision process (MDP). A MDP is defined by its state set S, action set A, transition probabilities T, and a reward function R [11]. The goal of using an MDP is to determine the best policy (i.e., the best path through this graph) that corresponds to solving the given problem. This is achieved by calculating a "value," the expected discounted sum of the rewards to be earned by following an optimal policy from state s, calculated recursively using value iteration [12]. When the hint button is pressed, the hint provider searches for the current state in the MDP and checks if that a successor state exists. If it does, the successor state with the highest value is used to generate a hint sequence. Once a student performs a correct step, the hint sequence is reset.

Barnes and Stamper (2008) demonstrated the feasibility of this approach on historical data, showing that extracted MDPs with the proposed hint-generating functions could provide correct next-step hints towards the problem solution over 80% of the time [10,12,15]. A pilot study showed that students were able to solve more problems when hints were included [1]. Since the Hint Factory is data-driven, the system can be bootstrapped with expert solutions [15]. And it can evolve, providing some automatically-generated hints initially, and improving as additional expert information and student attempts are added to the model.

## 2.4 Student Assessment and Support Tools

It is important for educators to have excellent visualization tools for student assessment. Further, research has shown that interaction with intelligent tutors could possibly be a better predictor of a student's knowledge than standard tests [17]. Such models are the basis for the kind of student-customized adaptive instruction that intelligent tutoring systems can provide [18].

On the knowledge tracing side, the use of visualizations to assess student performance using cognitive models [2] will be used to give educators the ability to assess student knowledge at a given point in time. Croy, Barnes, and Stamper applied a technique to visualize student proof approaches to allow teachers to identify problem areas for students [19]. The goal of implementing these assessment and support tools is to provide educators and administrators the view of student learning and how the system works to support their initiatives and improve learning. The idea that educators can identify areas in SCALE that seem to contradict what they see in the actual classroom will allow for the refinement of the student models in ways that make sense to the educator.

## 2.5 Support Tools for Content Designers and Developers

A key differentiator of SCALE is the "data-driven, human-centered" approach that achieves superior results to existing systems. The human-centered portion allows developers and educators to explore and improve the models discovered. To implement this, we have designed tools that are based on existing tools in the EDM community (www.educationaldatamining.org). These include:

1. Learning Curve Analysis tool
2. Performance Profiler tool
3. Model Exploration and Tagging tool
4. The Replay Tutor student simulator

The Learning Curve Analysis tool allows researchers to identify smooth learning curves. We expect that the learning curve for each well defined KC will be reasonably smooth. When the learning curve of a purported KC is noisy, with upward or downward "blips," the student model is suspect. This can often mean a KC needs to be split, since we know that when two KCs are represented as one, the learning curve will not be smooth [3]. If the student model is accurate, we expect the error rate to decline over the number of opportunities a student has to both learn and apply a KC. Thus, a flat learning curve is another indication of a potentially flawed student model. This is especially true if data supports the idea that students are learning a specific KC but the model does not reflect this.

The Performance Profiler tool allows researchers to view error rates by problem or KC and also view the predicted value of one or more models. Problems will be displayed in a column by error rate with shading representing the error. We also will show the predicted values of two proposed models with the lines and points. This will show that one model does a better job predicting some problems than others. This is useful both to identify problems that may have an excessively high error rate, and to identify problems where the models have a difficult time correctly predicting the student's answer.

The Model Exploration and Tagging tool allows educators and researchers to explore log data in a graph. This tool allows researchers a way to identify places in the instruction where KCs are tagged incorrectly as well as see areas where improvements can be made.

The Replay Tutor allows experts and educators to validate new models using existing log data. The simulator predicts what students might do by using a Bayesian knowledge tracing model [20]. This tool allows researchers to better understand how changes in a model might affect student performance without having to test the model in an actual classroom environment.

In addition to the ability to automatically generate these features, SCALE also includes a "feedback loop" to continue the improvement of the features over time as more data is collected. Several potential customers have noted that existing providers of adaptive student learning software do not have any way to explain why the system behaves as it does. In contrast, SCALE provides tools that let the instructors and developers explore the data using meaningful visualizations that will provide insights into student learning. Often this means identifying areas where the existing models contradict the data collected. Built around the concept of data curation, these tools can also be used to prompt the developers, educators, and users of the educational software for more human input in order to improve the underlying models that the system generates.

The pilot for SCALE has been conducted in multiple courses with our partners HarvardX and the Open Learning Initiative (OLI), and has been reported in citations [21-22]. The full integration and load testing for SCALE, including the dashboard and the problem selection, is discussed in the next section.

## 3 CONTINUED DEVELOPMENT AND ENHANCEMENTS

After completing pilot testing on multiple platforms and in classroom and online courses, we instituted our iterative feature driven design (FDD) process to evaluate feedback from content and platform developers. This feedback, and SCALE-generated quantitative data, was analyzed and used to determine key enhancements to realize a viable, market-ready version of the research findings on which SCALE is based. Here we discuss four primary feature areas:

1. SCALE Dashboard Visualizations and Functionality
2. Learner Mode Support
3. Problem Selection Extensions
4. Multiple Learner Models

### 3.1 SCALE Dashboard Visualizations & Functionality.

We have added dashboard support for all of the SCALE APIs, providing visual representations of content structures, including problems and knowledge components (KCs), such that tagging and relationships can easily been entered or maintained. SCALE import/export capabilities have been expanded for support of all data elements and refactored to improve performance. Finally, visualization features were extended to provide more options for viewing learning curves, such as by allowing any level of content groupings for all students or sets of students. This is helpful when performing testing with experimental and control groups to compare and contrast the learning outcomes for students.

### 3.2 Learner Mode Support.

In working with various learning platforms, we determined that edtech is often used differently by students, depending on which stage they are in in their learning, and how the technology is integrated with the teaching strategies in the classroom. We have therefore added the ability to support multiple learner modes, triggering SCALE to react in a tailored way to the student's current mode. Although the architecture is built to support any number of learner modes, we have currently identified and implemented the following three: Mastery Learning, Review, and Static Assessment. Mastery Learning mode is used to support the formative stage, so that the learning process and activities can be modified, adapting to the students' current state of knowledge, and thus increasing the efficacy of the learning system and promoting improved student attainment. Review mode is used to support students when they are reviewing previously learned concepts, such as during their preparation for a formal assessment. Review mode is still adaptive, in that the system adjusts what the student will see based on the current student mode. However, during review mode the student will have the opportunity to see every concept (skill or KC) at least once, essentially resetting the thresholds used by the system to determine what the student knows. Static Assessment mode can be used for a summative assessment of the students. Unlike Mastery Learning mode or Review mode, by default Static Assessment does not perform adaptive problem selection. We envision adding support for several additional learner modes as we complete additional research.

### 3.3 Problem Selection Extensions.

In order to provide the best learning experience for students, the problem selection methodology and algorithms have been enhanced and expanded. First, we have added the ability for the system to determine and record problem difficulty. While a theoretical view would manage this metric based on the number of KCs required to solve a problem, in practice, problems are not always tagged at the level of granularity or thoroughness required to capture difficulty level. So, SCALE now provides a means for problems to be categorized with a difficulty level to help align the problems, in order to deliver content to students at the correct level for each student. This determination is made based on analysis of the logged student data, using KC information and updated student models, and can also be informed by expert entry and tagging.

Second, we have also created a new ensemble selection method. Originally, SCALE used an optimized problem selection methodology, that integrated a moderate amount of random selection, in order to drive efficiency in learning. However, we have found that different students will thrive and excel in distinct ways, based on a variety of problem selection approaches. And so now, SCALE provides the framework for managing different problem selection methodologies using our patent-pending ensemble architecture. This involves supporting multiple learner personas associated with corresponding problem selection methods. While interacting with the learning platform, SCALE will align the problem selection methods based on the learning persona that is connected to each student. This alignment is determined automatically and in real-time as the student uses the SCALE-enabled learning system. Over time, and potentially depending upon content groupings, student personas may change. In this way the problem selection method will adapt as necessary, to align students with the appropriate persona, and thus the associated problem selection method that is best for each student while learning the subject matter.

### 3.4 Multiple Learner Models.

Models are built using the data as described previously. SCALE supports managing multiple student models for the same platform and content such that SCALE will learn about the student with each student interaction. On a period or ad-hoc basis, log data is used to refine existing models, sometimes splitting existing models or creating new models. This could be automated to be done in a real-time manner, but to date, content developers and teachers seem to want students to experience the learning through controlled model releases.

## 4 CONCLUSIONS AND FUTURE WORK.

The process of taking cutting edge research out of the university labs and using it to create a new product has been illuminating in numerous ways. Regardless of how “finished” new technology seems in the experimental stage, there are always refinements that only the real-world testing can discover. We received invaluable feedback and insight from both developers - who implement SCALE to seamlessly incorporate it into their own products - and teachers, who will be using SCALE-enabled technology and dashboards in the classroom itself. On the developer side, we found ways to streamline and improve how SCALE works, what information it delivers, and how both students and problem sets can be grouped together in different ways for deeper analysis. For teachers, we discovered that multiple learner modes is very important, due to the fact that students and teachers use the learning system in different ways for different purposes. Sometimes the material is new and being learned for the first time; sometimes the student needs to review for a test; and sometimes the teacher needs an objective, fixed assessment, that won’t change and adapt. (Of course, it is entirely possible to have adaptive assessments, too, such as with standardized test like the SAT, GRE, and versions of the ACT. However, to create a robust adaptive assessment requires much more student data that is generally available for a particular set of content material. Further, this requirement puts constraints on how and how quickly the material can be updated. And so a static assessment may work better in many situations.)

Our ensemble method of problem selection is quite promising, and also grew out of discussions and feedback with both teachers and developers. For example, students who learn Algebra I in 7th grade may be fundamentally different than student who learn the same content in 10th grade. And thus, these two different groups of students may need different support structures, and our models may require different assumptions in order to optimize learning. This work is very promising but still preliminary. We are in the process of conducting further experiments and refinements on these methods for determining the right selection method for individual students, including retrospective studies using our Replay Tutor and already-collected data; and new studies in participating classrooms. The ultimate goal will be to properly refine the personas: how many are appropriate or necessary, and in what ways is learning improved through their use? Is it more effective, more efficient, or both? And, importantly, to what degree are these personas generalizable, across content areas and across Learning Systems?

We will continue to address the needs of the market and will continue to conduct targeted pilot tests to assess results of new features and functionality. We are also addressing the need for adding activity selection, and not just problem selection, to find the next best activity to maximize learning. And finally, we have been working to integrate non-cognitive factors into the engine to provide additional data points of student affect and behaviors, to continue to contribute ways to keep students actively engaged and on improved trajectories of learning.

## 5 ACKNOWLEDGEMENTS

TutorGen gratefully acknowledges support of SCALE® from the National Science Foundation award numbers 1346448 and 1534780 and from the Commonwealth of Kentucky Cabinet for Economic Development, Kentucky Science and Engineering Foundation, and The Kentucky Science and Technology Corporation, award numbers KSTC-184-512-14-182 and KSTC-184-512-16-241.

## REFERENCES

1. Barnes, T. & Stamper, J. (2009). Automatic hint generation for logic proof tutoring using historical data. In Journal of Educational Technology & Society, Special issue on Intelligent Tutoring Systems, Vol. 13, Iss. 1. 2010.
2. Koedinger, K., Stamper, J. (2010). A Data Driven Approach to the Discovery of Better Cognitive Models. In Baker, R.S.J.d., Merceron, A., Pavlik, P.I. Jr. (Eds.) Proceedings of the 3rd International Conference on Educational Data Mining. (EDM 2010), pp. 325-326. Pittsburgh, PA.
3. Stamper, J., Koedinger, K.R. (2011) Human-machine Student Model Discovery and Improvement Using DataShop. In Kay, J., Bull, S. and Biswas, G. eds. Proceeding of the 15th International Conference on Artificial Intelligence in Education (AIED2011)
4. Blink, M.J., Stamper, J., and Carmichael, T. (2014) SCALE: Student Centered Adaptive Learning Engine. In S. Trausan-Matu et al. (Eds.) Proceedings of the 12th International Conference on Intelligent Tutoring Systems (ITS 2014), pp. 654-655, 2014. Springer.
5. Clark, R. E., Feldon, D., van Merriënboer, J., Yates, K., & Early, S. (2007). Cognitive task analysis. In J. M. Spector, M. D. Merrill, J. J. G. van Merriënboer, & M. P. Driscoll (Eds.), Handbook of research on educational communications and technology (3rd ed., pp. 577–593). Mahwah, NJ: Lawrence Erlbaum Associates.
6. Lee, R. L. (2003). Cognitive task analysis: A meta-analysis of comparative studies. Unpublished doctoral dissertation, University of Southern California, Los Angeles, California.
7. Wilson, M., & De Boeck, P. (2004). Descriptive and explanatory item response models. In P. De Boeck, & M. Wilson, (Eds.) Explanatory item response models: A generalized linear and nonlinear approach. New York: Springer-Verlag.
8. Stamper, J., Koedinger, K., McLaughlin, E. (2013) A Comparison of Model Selection Metrics in DataShop. In Proceedings of the 6th International Conference on Educational Data Mining (EDM 2013). Memphis, USA. Jul 6-9, 2013. pp. 284-287.
9. Cen, H., Koedinger, K. R., & Junker, B. (2006). Learning Factors Analysis: A general method for cognitive model evaluation and improvement. In M. Ikeda, K. D. Ashley, T.-W. Chan (Eds.)

Proceedings of the 8th International Conference on Intelligent Tutoring Systems, 164-175. Berlin: Springer-Verlag.

10. Stamper, J., Barnes, T., and Croy, M. (2011) Experimental Evaluation of Automatic Hint Generation for a Logic Tutor. In Kay, J., Bull, S. and Biswas, G. eds. Proceeding of the 15th International Conference on Artificial Intelligence in Education (AIED2011). pp. 345-352. Berlin Germany:Springer.
11. Sutton, S. and Barto, A.(1998). Reinforcement Learning: An Introduction. Cambridge: MIT Press.
12. Barnes, T., Stamper, J. (2008). Toward Automatic Hint Generation for Logic Proof Tutoring Using Historical Student Data. In Procs of the 9th International Conference on ITS, pp. 373-382. Berlin, Germany: Springer.
13. Stamper, J., Barnes, T., Croy, M. (2011). Enhancing the Automatic Generation of Hints with Expert Seeding. In The International Journal of Artificial Intelligence in Education (IJAIED), Special Issue on the Best of ITS 2010.
15. Stamper, J., Barnes, T., and Croy, M. (2010) Enhancing the Automatic Generation of Hints with Expert Seeding. In Aleven, V., Kay, J., and Mostow., J eds. Proceeding of the 10th International Conference on Intelligent Tutoring Systems (ITS2010). vol. II, pp. 31-40. Berlin, Germany: Springer Verlag.
16. Fossati, D., Di Eugenio, B., Ohlsson, S., Brown, c., Chen, L., Cosejo, D. I learn from you, you learn from me: How to make iList learn from students. In V. Dimitrova, R. Mizoguchi, B. Du Boulay and A. Graesser (Eds.), Proc. 14th Intl. Conf. on Artificial Intelligence in Ed, AIED 2009, pp. 186—195., Brighton, UK. IOS Press (2009)
17. Feng, M., Heffernan, N.T., & Koedinger, K.R. (2009). Addressing the assessment challenge in an online system that tutors as it assesses. User Modeling and User-Adapted Interaction: The Journal of Personalization Research (UMUAI). 19(3), pp. 243-266.
18. Koedinger, K. R. & Aleven, V. (2007). Exploring the assistance dilemma in experiments with Cognitive Tutors. Educational Psychology Review, 19 (3): 239-264.
19. Croy, M., Barnes, T., and Stamper, J. (2008). Towards an Intelligent Tutoring System for Propositional Proof Construction. In A. Briggle, K. Waelbers, & E. Brey (Eds.) Current Issues in Computing and Philosophy, pp. 145-155. IOS Press: Amsterdam, Netherlands.
20. Corbett, A.T. and Anderson, J.R.: Knowledge tracing: Modeling the acquisition of Procedural knowledge. User Modeling and User-Adapted Interaction, 4. (1995) 253-278.
21. Rosen, Y., Rushkin, I., Ang, A., Fredericks, C., Tingley, D., Blink, M.J., Lopez, G. (2017) Adaptive Assessment Experiment in a HarvardX MOOC. Proceedings of the 10th International Conference on Educational Data Mining. (EDM), Wuhan, China, June 25-28, 2017. pp. 466-471.
22. Rosen, Y., Rushkin, I., Ang, A., Fredericks, C., Tingley, D., Blink, M.J. (2017) Designing Adaptive Assessments in MOOCs. Proceedings of the 4th ACM Conference on Learning @ Scale (L@S 2017). (ACM, 2017), Cambridge, MA, pp. 233-236, ACM.