Invited Paper

Intelligent Technologies for Personalized Practice Systems

Peter Brusilovsky *

(Received XX XXXXX 20XX and accepted in revised form XX XXXXX 20XX)

Abstract Personalized practice systems focus on supporting self-organized learning in a free practice mode. Adapting to the learners' knowledge and goals, these systems help them navigate the increasing volumes of smart learning content, guide them to practice opportunities that are most appropriate to their level of knowledge and increase their motivation to practice. In this paper, we distill the experience generated by 20 years of research on personalized practice systems into a set of AI-based technologies that make these systems efficient, engaging, and user-friendly.

Keywords personalized practice, adaptive learning, navigation support, content recommendation, computer-science education

1. Introduction

With the increasing popularity of Computer Science, the size of the introductory programming classes and the diversity of students in the classes have increased remarkably challenging traditional pedagogies and learning tools. Programming skills can't be mastered by reading alone, to gain them students need a considerable amount of practice studying worked examples and solving programming problems. However, practice opportunities offered by traditional classes and modern MOOCs through labs and assignments are rarely sufficient for less prepared and non-traditional students. To master programming skills, this rapidly expanding cohort of students need to study programming examples and solve problems on their own, beyond the required minimum of labs and assignments. This work known as free practice enables less prepared students to level the playing field practicing as much as necessary to achieve mastery while also focusing on the most important or least studied topics.

The importance of free practice has been long recognized by the Computer Science Education community, which developed a large variety of advanced tools to help students practice on their own. These tools frequently referred to as interactive or "smart" learning content [1] include a variety of "worked example" tools (annotated examples [2], program visualizations [3], and coding tutorials [4]) focused on communicating code understanding and code construction knowledge to the learners and various types of automatically assessed problems (code tracing problems [5], Parson's problems [6], coding problems [7]) engaging students in applying and mastering this knowledge.

Unfortunately, the use of these advanced tools to support free practice is hampered by two problems. First, despite the need to practice and the confirmed effectiveness of modern tools in supporting it, only a fraction of students use these tools for free practice and a smaller fraction use them regularly [5; 8; 9]. Second, even for well-motivated students who are ready to allocate time for free practice, it becomes increasingly harder to efficiently use this time due to rapidly increasing volumes of available smart content. Large collections of smart content are now frequently provided by publishers as additions to traditional or interactive textbooks [10]. Developers of smart content tools increasingly release collections of smart content for free practice [11; 12]. These increasing volumes of available practice content make it harder to choose the content for practice, which due to the known paradox of choice, further decreases their engagement with practice [13].

We believe that to make free practice efficient, students need *personalized practice systems*, a new type of learning tools focused on supporting student work in a free practice mode. It should help students navigate the increasing volumes of smart learning content, guide them to practice opportunities that are most appropriate to their level of knowledge, and increase their motivation to practice.

For over 20 years our research team has been exploring technologies, interfaces, and infrastructures for building personalized practice systems. In this paper, we share a part of our experience by distilling a set of

* School of Computing and Information, University of Pittsburgh, USA

Artificial Intelligence technologies that make these systems efficient, engaging, and user-friendly. Following this introduction, each section of the paper highlights one specific technology that we currently use in our personalized programming practice systems and which, as we believe, one of the components of their success. We introduce the features "historically" – focusing on the context in which this feature was originally developed, the motivation for its development, and the first rounds of its evaluation. This approach helps stress the importance and the value of each feature in research on personalized practice. In cases when a specific feature has been developed further in future work, we provide necessary references to more recent designs and results.

2. Adaptive Navigation Support

The first technology that could be used for turning a collection of practice-oriented smart learning content into a personalized practice system is *adaptive navigation support* [14], a personalized guidance technology developed in the field of adaptive hypermedia. Adaptive navigation support could be helpful to adapt to the students' varying levels of knowledge at the start of the course as well as to support more flexible free practice. Indeed, some students start the course with partial knowledge of the subject, while others are complete beginners. Some try to practice every week while others might skip a week or two or delay all practice until exam preparation. When entering a practice system after each lecture, different students might need to work with different problems and examples prepared by the instructor for this lecture. Moreover, the most relevant practice for a specific student could be only found in earlier lectures since the student lacks prerequisite knowledge or failed to practice in time.

A useful technology that can help students navigate in a maze of complex learning content with multiple prerequisites is *prerequisite-based navigation support* [14], which was developed in early adaptive hypermedia systems ELM-ART [15] and InterBook [16]. This technology traces the changing level of student knowledge for multiple concepts of the course, decides whether the target student is ready to attempt a specific item of learning content or lacks prerequisite knowledge, and makes this judgment visible to the student through adaptive link annotation.

An example of using this personalized guidance approach for personalized practice is NavEx system, which added adaptive navigation support to a collection of worked examples delivered by WebEx system [17]. NavEx provided links to all worked examples in the course on the left side of its interface adding a personalized icon (known as adaptive link annotation) to each example link (Figure 1 left). A click on an example link opens an example in the right frame (Figure 1 right). The personalized icon estimates whether the example is ready to be explored by the target student at the given time. A red cross sign indicates that example is not yet ready to be explored since the student likely misses the prerequisite knowledge that is necessary to understand it. A green bullet indicates that an example is ready to be studied. Moreover, the filling of the bullet indicated how many lines of a specific example had been explored by the student in the past. A filled bullet indicates a fully explored example with all explanations being examined by the student.

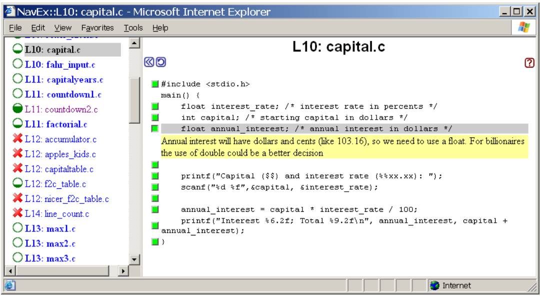

Figure 1. Access to WebEx examples through NavEx interface. A list of all course examples augmented by adaptive link annotations is shown on the left.

Table 1. Student engagement with NavEx and WebEx

	WebEx	NavEx+ WebEx	p-value
Lines	34.76±6.66	171.90±65.56	<.001
Examples	5.66±0.871	18.10±4.32	<.001
Lectures	3.52±0.42	8.20±1.23	<.001

We explored NavEx in several classroom studies comparing it to the baseline condition (WebEx examples were used without adaptive navigation support) [2]. Despite its simplicity, the effect of adaptive navigation support was remarkable (Table 1). With personalized guidance, the number of explored examples increased 3 times, the number of explored lines 5 times (i.e., examples were explored deeper) and the lecture coverage of explored examples more than doubled.

3. Open Learner Modeling

Open learner modeling [18] is a technology that makes the learner model, the heart of every personalized learning system, visible to the learners. Open Learner Models

(OLM) show how much knowledge students gain for different topics or concepts of the course. With OLM, students are constantly aware of their performance and can better decide what to focus on. In our early work, we focused on a coarse-grain OLM known as a topic-level OLM. An example of a system with a topic-level OLM is QuizGuide [19], an adaptive hypermedia service for personalized access to code-tracing problems, generated by the previously developed system, QuizPACK [5].

QuizGuide groups practice problems into coarse-grained course topics (Figure 2 left). The link to each topic is annotated with an icon showing a target with (or without) arrows. The number of arrows (from 0 to 3) reflects the student's performance on the quizzes of that annotated topic (from no arrows representing very little or no progress to three arrows representing good comprehension). With this approach, the topic targets serve as a topic-level OLM. QuizGuide combines this OLM with additional navigation support. The topic targets also serve as links to tracing problems for the topic assembled into 5-problem *quizzes*. A click on a topic link unfolds/folds links to the quizzes available for this topic. The color of a target encodes the relevance of a topic to the current learning goal of the class. The topics form a prerequisite-outcome structure. Every time new topics are introduced in a lecture, they are annotated with a bright-blue target. Topics that serve as prerequisites for any of the current topics have a pale-blue target. Completed topics are assigned grey targets. Finally, topics that belong to learning goals not yet covered in class are annotated with crossed targets.

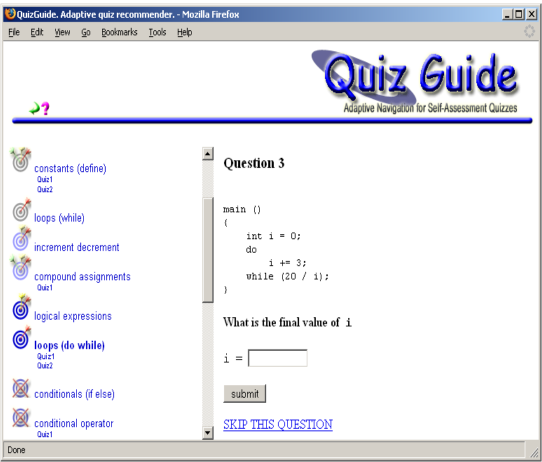

Figure 2. QuizGuide: Access to QuizPACK code tracing problems using OLM and Navigation support.

One of the goals of combining OLM with navigation support used in QuizGuide was to help a student choose which topic is most appropriate to work bringing them to the right topic at the right time and preventing them from leaving insufficiently studied topics behind, which could hinder their problem-solving performance. The result of system evaluation in several classroom studies demonstrated that QuizGuide affected student learning in several positive ways [19]. Similar to the NavEx case, adaptive navigation support significantly increased student engagement with learning content, the number of quizzes used for practice doubled. OLM helped students increase topic coverage and pay significantly more attention to practices with insufficiently explored topics from past lectures. In turn, it significantly increased the learning gain.

Following the success of topic-level OLM in QuizGuide [19] we used topic-level OLM as a key component for all of our personalized practice systems including JavaGuide [20] and SQL-Guide [21], which used a QuizGuide interface to guide learners to Java and Database practice problems, and several other designs [22; 23]. Our most recent work on personalized practice systems for Java, Python, and SQL uses a topic-level open-source OLM interface Mastery Grids [24]. This interface was designed to accommodate a varying number of course topics and multiple kinds of smart content. We used this design to develop personalized practice systems for Java, Python, and SQL courses taught in many universities across several countries.

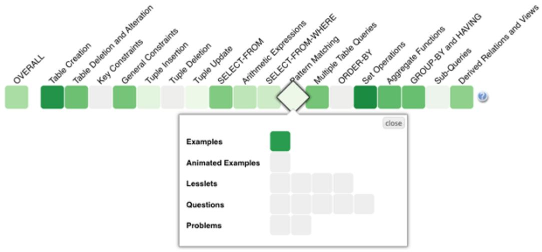

Figure 3. Mastery Grids interface for personalized SQL practice system.

Figure 3 shows a Mastery Grids interface for a personalized SQL practice system. A progression of 18 SQL topics here is presented horizontally as a row of green cells. The learner's current knowledge of the course topics is shown as a row of square cells using green colors of different intensities. The darker the topic cell color is, the more knowledge progress was achieved by the learner in the topic. Clicking on a topic cell opens a table of practice content for this topic organized by type. Here the color intensity indicates the amount of work already done by the learner with this content. In

Figure 3, the learner noticed that their knowledge gained on the *Pattern Matching* topic is very low and opens available content to practice. The content table shows five types of content to practice with only one example explored so far.

A known problem of otherwise efficient topic-level OLM is its low granularity level for reflecting learners' knowledge. For example, our Java ontology developed for practice content indexing and knowledge modeling includes over 200 concepts for which learner knowledge could be individually tracked [25]. However, topic-level OLMs for Java in JavaGuide [20] and Mastery Grids use less than 20 topics to visualize learner knowledge on the topic level. Since many topics include multiple concepts, reflecting learner knowledge on such a coarse level could lead to neglecting important concepts and gaining unbalanced knowledge of course topics. In our recent work, we frequently complement a topic-level OLM with a much finer-grained OLM representing learner knowledge on the *concept* level.

To determine the best approach to visualize a concept-level OLM, we performed a comparative study [26]. Based on this study, we selected a bar chart OLM representation (Figure 4 top), which we now use in our recent personalized practice systems in addition to the topic-level model (Figure 3). To support the parallel use of topic-level and concept-level models, the order of concepts in the concept-level OLM follows the order of their introduction in topic-level models, and a synchronized visualization shows which concepts are associated with each course topic (Figure 4 bottom).

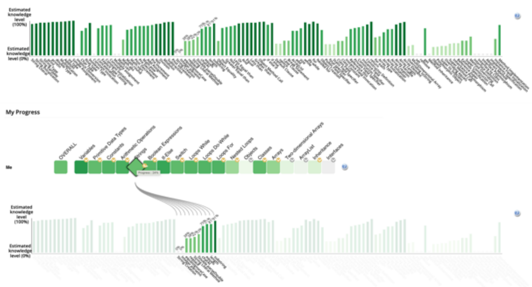

Figure 4. Concept-Level OLM for Java. Top: Current level of learner knowledge of Java concepts. Bottom: Connecting topic-level and concept-level OLM.

4. Open Social Learner Modeling

The growing popularity of online learning with multiple students interacting directly or indirectly in the learning process motivated a large stream of research on social learning technologies that leverage the power of the learner community [27]. Among other technologies, social comparison [28] emerged as an efficient approach to increase user engagement and participation [29]. In our work on personalized practice systems, we used social comparison aas a component of the Open *Social* Learner Model (OSLM). With OSLM, students can compare the current state of their knowledge of the domain shown by an OLM with the models of other learners or the whole class visualized in a similar way.

Figure 5. OSLM in Progressor: Peers' progress is displayed as thumbnails next to the user's OLM.

We explored several approaches to design OSLM and achieved considerable success with the design known as Progressor [23]. Progressor visualized the state of the learner's topic-level progress as uneven segments of a circle where the color of each segment (red to green) indicated the amount of knowledge for the topic accumulated through practice (Figure 5). This design was different from the OLMs in our earlier personalized practice systems such as QuizGuide (Figure 2) and was specifically focused on comparing two OLMs side by side. When checking their OLMs, the learners could see thumbnails of OLMs of other learners in their class (peers) and can select any of these models for a side-by-side comparison of peer's knowledge with their own.

Progressor was used to provide access to parameterized tracing questions for Java programming language delivered by QuizJET [30], a Java version of QuizPACK. To examine the value of OSLM we compared the effectiveness of personalized practice with Progressor interface to non-personalized practice with QuizJET system in several classroom studies [23]. As the data shows (Table 2), the use of OSLM lead to a remarkable increase of engagement with practice questions. The number of sessions with the system and the number of question attempts more than doubled, the coverage of course topics increased considerably, and the success rates in solving tracing problems increased from frustrating 42.63% to 68.39%. A follow-up study with Progressor+, an extension of Progressor, confirmed that OSLM could significantly increase learner engagement with several types of practice content used in parallel [31].

Table 2. Practicing with QuizJET vs Progressor

	QuizJET	Progressor
Attempts	80.81±22.06	205.73±40.46
Success Rate (%)	42.63±1.99	68.39±4.32
Distinct Topics	7.81±1.64	11.47±1.34
Distinct Questions	33.37±6.50	52.7±6.92
Sessions	3.75±0.53	8.4±1.39

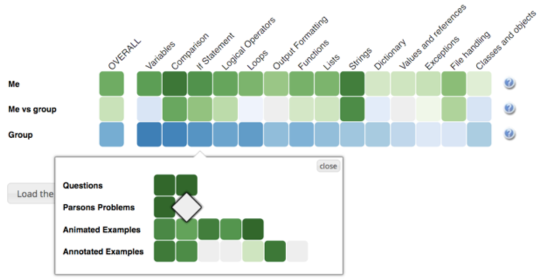

Figure 6. A Mastery Grids interface with multiple kinds of Python practice content and enabled OSLM

Recognizing the importance of Open Social Learner Modeling, an OSLM component was integrated into our Mastery Grids interface for personalized practice [24] at the design time. Figure 6 shows Mastery Grids installation for a personalized Python practice system with the OSLM component enabled [32]. Here Mastery Grids provides personalized access to four types of practice content (two types of problems and two types of worked examples) organized into 14 topics. The learner's knowledge of course topics is shown on the top row of the grid using green colors of different intensities, as explained in the previous section. The bottom row displays the average knowledge of the class using blue colors of different intensities. The middle row provides an easy way to compare learners' knowledge progress with the progress of the class – the green color indicates that the learner is ahead of the class in this topic while the blue color indicates that the learner is lagging behind the class. A click on a topic cell opens a panel that provides access to four kinds of smart learning content items for this topic. With this design, the learners can easily locate course topics that need more work while the ability to compare their progress with the class increases their motivation to practice.

Mastery Grids OSLM interface has been explored in several studies, which confirmed the remarkable engaging power of OSLM technology [33; 34; 35]. The studies also demonstrated that OSLM enabled learners to work more efficiently and increased their knowledge monitoring ability, which is an important skill in the context of self-regulated learning.

5. Learning Content Recommendation

Despite of demonstrated effectiveness of navigation support provided by prerequisite-based link annotation, OLM, and OSLM reviewed above, these technologies don't offer the learner a direct answer which learning content item is the best to practice at the given moment. The personalized guidance provided by these technologies is *indirect* – it offers several kinds of information that help select the next item but leaves the choice to the learner. From our past research on personalized guidance [15], we know that this indirect guidance works well for better-prepared learners with starting knowledge of the domain. In contrast, learners with little to no knowledge of the domain work best with a more direct form of guidance such as content sequencing [36] or learning content recommendation [37].

Due to the complementary nature of adaptive navigation support and learning content recommendations, for the last 10 years we frequently used these two technologies in our personalized practice systems side-by side by integrating recommendations into Mastery Grids interface. In our first attempt to combine recommendations and adaptive navigation support, we decided to avoid the traditional ranked list, the dominant approach to present recommendations. Instead, we used stars icons to mark topics where recommended content is located and, once the topic is opened, to mark the recommended content itself (Figure 7).

We evaluated this design in a classroom study [38] combining it with a "knowledge maximizer" approach for learning content recommendation that we developed earlier [39]. With this approach, a concept-based recommender algorithm prefers learning content that maximizes the knowledge gained by practicing with this content without breaking prerequisites. The results of the study demonstrated that adding direct recommendations to the indirect guidance provided by the navigation support helped to break suboptimal sequential navigation pattern and encouraged non-sequential work with learning content. However, we also discovered that "maximizer" focused on a rapid advance through course concepts was mostly beneficial for stronger students helping them significantly shorten their learning paths.

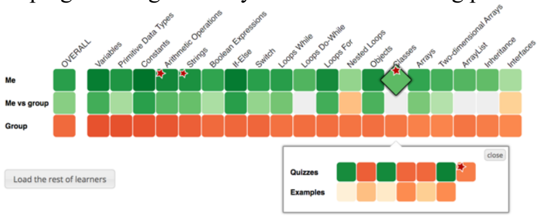

Figure 7. Recommending learning content using "Knowledge Maximizer" approach in Mastery Grids. Star icons are used to mark recommended content.

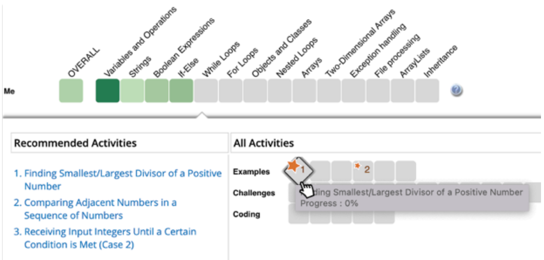

Figure 8. Content recommendation for a selected topic based on balanced knowledge expansion using a combination of a ranked list and stars icons.

In our follow-up work on content recommendation, we replaced the "maximizer" approach with a balanced knowledge expansion approach [40; 41]. This approach attempted to select learning content with an optimal balance of for prerequisite and target concepts. It assures that prerequisite concepts for a practice content item are sufficiently learned and prevents the learner from practicing too many new concepts at the same time. Following student feedback, we also started present recommended items for each topic as a ranked list in addition to marking them with stars (Figure 8). We also developed a remedial recommendation approach focused on error remediation [42]. The idea of the remedial recommendation was to isolate concepts associated with problems that the learner failed to solve correctly. Assuming that these concepts might be the source of troubles, the remedial recommendation approach focused on examples and problems that offer the best opportunity to practice the troublesome concepts. The studies of both approaches demonstrated that direct recommendation has a strong impact on student work, encouraging them to practice with recommended content. In contrast to the "maximizer" approach, the balanced recommendation approach has not left weaker students behind. We observed that low-pretest students preferred to practice with recommended items and achieved significantly higher success rates by following recommendations [40].

6. Transparency and Explanations

Transparency and explanations are considered important features of modern AI [43]. To make human decision-making assisted by AI, the behavior and recommendations of the AI system need to be understandable to the human using the system. In the context of personalized information access (i.e., adaptive navigation support and recommendation) transparency is frequently achieved by making the decision steps of the personalized guidance algorithm visible or on-demand *scrutable* by the end users. Multiple studies demonstrated that visual transparency and scrutability increase user confidence and trust in the provided recommendation [44; 45; 46].

The original design of the Mastery Grids interface for personalized access to practice content included several transparency features. For example, the amount of knowledge gained by the learner on a specific topic could be scrutinized by examining the amount of progress over practice content offered for this topic shown by the intensity of the green color (Figure 6). The average knowledge of the whole class for a specific topic could be scrutinized by opening a panel showing topic-level performance of every student in the class (Figure 9). These details help students understand how the estimation of average class knowledge works and build trust in the data shown by the OSLM.

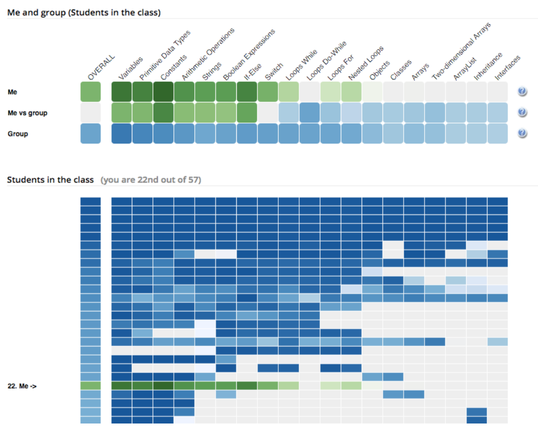

Figure 9. Scrutinizing average knowledge progress of the class by viewing the progress of individual peers.

In our most recent work, we attempted to add another layer or transparency to support the main goal of a personalized practice system, i.e., helping learners to identify the most relevant learning content to practice. The first step in making the content recommendation more transparent is the concept-level model of learner knowledge visualized as OLM (Figure 4) since this model is the main source of information for personalized content selection approaches [47]. However, this model is not sufficient to provide content selection transparency because it doesn't explain how a specific recommendation approach works and how it uses the current state of the model to identify the most relevant content.

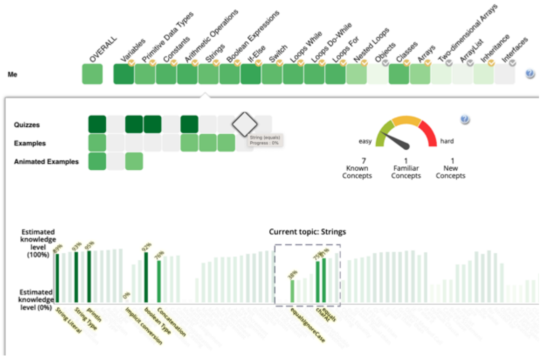

Figure 10. Transparent content selection for the balanced knowledge expansion approach**.**

An example of making the content selection process transparent for the case of balanced knowledge expansion approach (Figure 8) is shown in Figure 10. Here the interface uses concept-level OLM on the bottom to highlight concepts associated with a smart content item that the learner considers practicing with. In the figure, the learner considers working with a code tracing quiz for the topic "Strings" (second item from the left in the top content row). Once the mouse is placed over the content cell (the currently examined content cell turns 45 degrees and looks like a diamond), the interface highlights all concepts associated with the content items in the OLM. The idea of the balanced approach is to consider separately *prerequisite* and *target* concepts that are associated with the selected item and recommend content with optimal balance. To make the process more clear the visualization also separates target concepts showing them within "current topic" box. In Figure 10, we see that these concepts are *equalIgnoreCase* (current knowledge 38%), *equal* (current knowledge 81%), and *charAt* (current knowledge 75%). Six prerequisite concepts, which are expected to be learned earlier, are highlighted on the left. Five of these concepts are already well known, but *implicit conversion* is new with 0% knowledge. The gauge on the right assesses the balance of these concepts and with 7 known, 1 familiar (*equalIgnoreCase*) and one new concept (*implicit conversion*) assesses the selected quiz as an easy one for the current knowledge of the learner. More details about this transparency-oriented visualization for the balanced recommendation approach can be found in [40; 48] while a different transparency visualization for remedial recommendation approach is presented in [42].

With this level of transparency, the learner can scrutinize any recommended content item to confirm that it features an optimal balance of target and prerequisite concepts. Alternatively, this visualization could be used as an advanced adaptive navigation support approach helping the learners select the best practice items themselves [41]. Our studies of this transparency approach [48] indicated that the visualization shown in Figure 10 was positively assessed by the learners helping them find new concepts to practice while avoiding items that are too easy or unnecessarily hard. However, it also indicated that the designers of transparency-oriented visualization need to balance the complexity of visualization with the support provided by it to the user needs. In our case, an attempt to add concept-level OSLM to Figure 10 was judged negatively since it made the visualization considerably more complex without adding useful additional support. At the same time, the addition of the gauge, which efficiently summarized the information presented by the OLM was assessed very positively – adding little complexity it provided efficient support in locating content to practice.

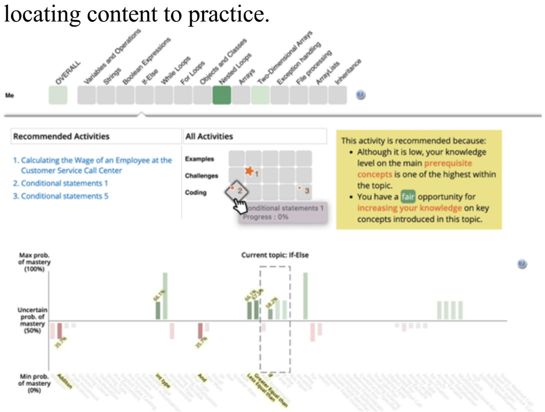

Figure 11. Explaining recommendations using transparency visualization and text explanations

A positive effect of the gauge also indicated that even a well-designed transparency visualization could be hard to understand so it has to be complemented by a more clear message stressing how a recommended item matches the learner's needs. In recommender systems, this message is typically provided by explaining recommendations [49]. In our most recent work [40; 41], we compared transparency visualization with text-based explanations of recommended items applying these approaches separately and in combination (Figure 11). Here the interface does both – explains why the balanced approach considers the recommended problem good for the learner (yellow note) and highlights the current level of learner knowledge for the target and prerequisite concepts associated with this problem (bottom part).

A classroom study of transparency visualization and explanations [41] demonstrated that both functionalities were extensively used by the learners to select practice content. We also found adding text-based explanations increases user engagement with learning content and helps low-pretest students understand what the system recommends them to practice and how that relates to their current state of learning.

8. Summary and Conclusion

In this paper, we attempted to summarize over 20 years of experience in developing and evaluating personalized practice systems for computer science education. We distilled several AI technologies, which make personalized practice systems more efficient and engaging for learners, and presented these technologies in the order they were developed and explored by our team. As we noted, once a successful approach has been discovered and its effectiveness has been confirmed in several contexts, we usually include these top-performing technologies in each of our personalized practice systems. In particular, our oldest well-explored technologies such as adaptive navigation support, OLM, and OSLM are used in most of our recent systems and form the basis of our open-source platform Mastery Grids, which has been extensively used over the last 10 years. Our recently developed technologies including specific content recommendation approaches, transparency interfaces, and explanations require further research to distill the most efficient approaches for different contexts and audiences.

Our current work on personalized practice focuses on improving support for self-regulated learning, learner control, and human-AI collaboration [50], which offer opportunities for learners to exercise their agency in the AI-assisted learning process. In this context, we plan to continue research on transparency and explainable recommendations.

9. Contact Address

Peter Brusilovsky, University of Pittsburgh, School of Computing and Information, 135 North Bellefield Ave., Pittsburgh, PA 15260, USA
E-mail: peterb@pitt.edu

Acknowledgements

The work presented in this paper is a results of research efforts of many students in Personalized and Adaptive Systems Lab at the University of Pittsburgh. The author especially wants to acknowledge important contributions of Sergey Sosnovsky, Michael Yudelson, I-Han Hsiao, Tomasz Loboda, Roya Hosseini, Jordan Barria-Pineda, and Kamil Akhuseyinoglu. This work has also been partially supported by NSF awards 0310576, 0447083, 0633494, 1740775, and 2213789.

References

[1] P. Brusilovsky et al., "Increasing Adoption of Smart Learning Content for Computer Science Education," in: *Proc. Working Group Reports of the 2014 on Innovation and Technology in Computer Science Education Conference*, Uppsala, Sweden, pp. 31-57, doi: 10.1145/2713609.2713611.

[2] P. Brusilovsky, and M. Yudelson, "From WebEx to NavEx: Interactive Access to Annotated Program Examples," *Proceedings of the IEEE*, Vol. 96, No. 6, pp. 990-999, 2008, doi: 10.1109/JPROC.2008.921611.

[3] J. Sorva, V. Karavirta, and L. Malmi, "A Review of Generic Program Visualization Systems for Introductory Programming Education," *ACM Transactions on Computing Education*, Vol. 13, No. 4, 2013, doi: 10.1145/2490822.

[4] K. Khandwala, and P. J. Guo, "Codemotion: Expanding the Design Space of Learner Interactions with Computer Programming Tutorial Videos," in: *Proc. Fifth Annual ACM Conference on Learning at Scale*, New York, NY, USA, pp. 57:1--57:10, doi: 10.1145/3231644.3231652.

[5] P. Brusilovsky, and S. Sosnovsky, "Individualized Exercises for Self-Assessment of Programming Knowledge: An Evaluation of QuizPACK," *ACM Journal on Educational Resources in Computing*, Vol. 5, No. 3, pp. Article No. 6, 2005, doi: 10.1145/1163405.1163411.

[6] D. Parsons, and P. Haden, "Parson's programming puzzles: a fun and effective learning tool for first programming courses," in: *Proc. the 8th Australasian Conference on Computing Education*, pp. 157-163.

[7] P. Ihantola, T. Ahoniemi, V. Karavirta, and O. Seppälä, "Review of recent systems for automatic assessment of programming assignments," in: *Proc. 10th Koli Calling International Conference on Computing Education Research*, pp. 86-93, doi: 10.1145/1930464.1930480.

[8] T. Naps et al., "Evaluating the educational impact of visualization," *ACM SIGCSE bulletin*, Vol. 35, No. 4, pp. 124–136, 2003.

[9] R. Hosseini, K. Akhuseyinoglu, P. Brusilovsky, L. Malmi, K. Pollari-Malmi, C. Schunn, and T. Sirkiä, "Improving Engagement in Program Construction Examples for Learning Python Programming," *International Journal of Artificial Intelligence in Education*, Vol. 30, No. 2, pp. 299-336, 2020, doi: 10.1007/s40593-020-00197-0.

[10] B. Ericson, M. Guzdial, B. Morrison, M. Parker, M. Moldavan, and L. Surasani, "An eBook for Teachers Learning CS Principles," *ACM Inroads*, Vol. 6, No. 4, pp. 84--86, 2015, doi: 10.1145/2829976.

[11] V. Barr, and D. Trytten, "Using turing's craft codelab to support CS1 students as they learn to program," *ACM Inroads*, Vol. 7, No. 2, pp. 67–75, 2016, doi: 10.1145/2903724.

[12] T. Sirkiä, and L. Haaranen, "Improving online learning activity interoperability with Acos server," *Software: Practice and Experience*, Vol. 47, No. 11, pp. 1657–1676, 2017, doi: 10.1002/spe.2492.

[13] A. Hicks, K. Akhuseyinoglu, C. Shaffer, and P. Brusilovsky, "Live Catalog of Smart Learning Objects for Computer Science Education," in: *Proc. Sixth SPLICE Workshop "Building an Infrastructure for Computer Science Education Research and Practice at Scale" at ACM Learning at Scale 2020*, Virtual, August 12, 2020.

[14] P. Brusilovsky, "Adaptive navigation support," in: *The Adaptive Web: Methods and Strategies of Web Personalization*, P. Brusilovsky, A. Kobsa and W. Neidl, Eds., Lecture Notes in Computer Science, 4321, Berlin Heidelberg New York: Springer-Verlag, 2007, pp. 263-290.

[15] G. Weber, and P. Brusilovsky, "ELM-ART: An adaptive versatile system for Web-based instruction," *International Journal of Artificial Intelligence in Education*, Vol. 12, No. 4, pp. 351-384, 2001.

[16] P. Brusilovsky, J. Eklund, and E. Schwarz, "Web-based education for all: A tool for developing adaptive courseware," in: *Proc. Seventh International World Wide Web Conference*, H. Ashman and P. Thistewaite, Eds., Brisbane, Australia, 14-18 April 1998, pp. 291-300.

[17] M. Yudelson, and P. Brusilovsky, "NavEx: Providing Navigation Support for Adaptive Browsing of Annotated Code Examples," in: *Proc. 12th International Conference on Artificial Intelligence in Education, AI-Ed'2005*, C.-K. Looi, G. McCalla, B. Bredeweg and J. Breuker, Eds., Amsterdam, the Netherlands, July 18-22, 2005, pp. 710-717.

[18] S. Bull, "There are open learner models about!," *IEEE Transactions on Learning Technologies*, Vol. 13, No. 2, pp. 425 - 448, 2020, doi: 10.1109/TLT.2020.2978473.

[19] S. Sosnovsky, and P. Brusilovsky, "Evaluation of Topic-based Adaptation and Student Modeling in QuizGuide," *User Modeling and User-Adapted Interaction*, Vol. 25, No. 4, pp. 371-424, 2015, doi: 10.1007/s11257-015-9164-4.

[20] I.-H. Hsiao, S. Sosnovsky, and P. Brusilovsky, "Guiding students to the right questions: adaptive navigation support in an E-Learning system for Java programming," *Journal of Computer Assisted Learning*, Vol. 26, No. 4, pp. 270-283, 2010, doi: 10.1111/j.1365-2729.2010.00365.x.

[21] P. Brusilovsky, S. Sosnovsky, D. Lee, M. Yudelson, V. Zadorozhny, and X. Zhou, "Learning SQL programming with interactive tools: from integration to personalization," *ACM Transactions on Computing Education*, Vol. 9, No. 4, pp. Article No. 19, pp. 1-15, 2010, doi: 10.1145/1656255.1656257.

[22] I.-H. Hsiao, F. Bakalov, P. Brusilovsky, and B. König-Ries, "Open Social Student Modeling: Visualizing Student Models with Parallel Introspective Views," in: *Proc. 19th International Conference on User Modeling, Adaptation, and Personalization, UMAP 2011*, J. Konstan, R. Conejo, J. Marzo and N. Oliver, Eds., Girona, Spain, July 11-15, 2011, pp. 171-182.

[23] I.-H. Hsiao, F. Bakalov, P. Brusilovsky, and B. König-Ries, "Progressor: social navigation support through open social student modeling," *New Review of Hypermedia and Multimedia*, Vol. 19, No. 2, pp. 112-131, 2013, doi: 10.1080/13614568.2013.806960.

[24] T. Loboda, J. Guerra, R. Hosseini, and P. Brusilovsky, "Mastery Grids: An Open Source Social Educational Progress Visualization," in: *Proc. 9th European Conference on Technology Enhanced Learning (EC-TEL 2014)*, S. de Freitas, C. Rensing, P. J. Muñoz Merino and T. Ley, Eds., Graz, Austria, September 16-19, 2014, pp. 235-248.

[25] S. Sosnovsky, P. Brusilovsky, I.-H. Hsiao, and M. Yudelson, "Java Ontology," PAWS Lab, 2010, https://sites.pitt.edu/~paws//ont/java.owl.

[26] J. Guerra Hollstein, J. Barria Pineda, C. Schunn, S. Bull, and P. Brusilovsky, "Fine-Grained Open Learner Models: Complexity Versus Support," in: *Proc. 25th Conference on User Modeling, Adaptation and Personalization*, Bratislava, Slovakia, pp. 41-49, doi: 10.1145/3079628.3079682.

[27] J. Vassileva, "Toward Social Learning Enviroments," *IEEE Transaction on Learning Technologies*, Vol. 1, No. 4, pp. 199-214, 2008.

[28] P. Dijkstra, H. Kuyper, G. van der Werf, A. Buunk, and Y. van der Zee, "Social Comparison in the Classroom: A Review," *Review of Educational Research*, Vol. 78, No. 4, pp. 828-879, 2008, doi: 10.3102/0034654308321210.

[29] J. Vassileva, "Motivating participation in social computing applications: a user modeling perspective," *User Modeling and User-Adapted Interaction*, Vol. 22, No. 1-2, pp. 177-201, 2012.

[30] I.-H. Hsiao, P. Brusilovsky, and S. Sosnovsky, "Web-based Parameterized Questions for Object-Oriented Programming," in: *Proc. World Conference on E-Learning, E-Learn 2008*, Las Vegas, USA, November 17-21, 2008, pp. 3728-3735.

[31] I.-H. Hsiao, and P. Brusilovsky, "Guiding and Motivating Students Through Open Social Student Modeling: Lessons Learned," *Teachers College Record*, Vol. 119, No. 3, 2017, doi: 10.1177/016146811711900302.

[32] P. Brusilovsky, L. Malmi, R. Hosseini, J. Guerra, T. Sirkiä, and K. Pollari-Malmi, "An integrated practice system for learning programming in Python: design and evaluation," *Research and Practice in Technology Enhanced Learning*, Vol. 13, No. 18, pp. 18.1-18.40, 2018, doi: 10.1186/s41039-018-0085-9.

[33] S. Somyurek, and P. Brusilovsky, "Impact of Open Social Student Modeling on Self-Assessment of Performance," in: *Proc. E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, Kona, HI, USA, October 19-22, 2015, pp. 1181-1188.

[34] P. Brusilovsky, S. Somyurek, J. Guerra, R. Hosseini, V. Zadorozhny, and P. Durlach, "Open Social Student Modeling for Personalized Learning," *IEEE Transactions on Emerging Topics in Computing*, Vol. 4, No. 3, pp. 450-461, 2016, doi: 10.1109/TETC.2015.2501243.

[35] S. Somyürek, P. Brusilovsky, and J. Guerra, "Supporting knowledge monitoring ability: open learner modeling vs. open social learner modeling," *Research and Practice in*

Technology Enhanced Learning, Vol. 15, No., pp. Article number 17, 2020, doi: 10.1186/s41039-020-00137-5.

[36] D. McArthur, C. Stasz, J. Hotta, O. Peter, and C. Burdorf, “Skill-oriented task sequencing in an intelligent tutor for basic algebra,” *Instructional Science*, Vol. 17, No. 4, pp. 281-307, 1988.

[37] H. Drachsler, K. Verbert, O. Santos, and N. Manouselis, “Panorama of Recommender Systems to Support Learning,” in: *Recommender Systems Handbook*, F. Ricci, L. Rokach and B. Shapira, Eds., Boston, MA: Springer, 2015 pp. 421-451.

[38] R. Hosseini, I.-H. Hsiao, J. Guerra, and P. Brusilovsky, “What Should I Do Next? Adaptive Sequencing in the Context of Open Social Student Modeling,” in: *Proc. 10th European Conference on Technology Enhanced Learning (EC-TEL 2015)*, G. Conole, T. Klobučar, C. Rensing, J. Konert and É. Lavoué, Eds., Toledo, Spain, September 15–18, 2015, pp. 155-168, doi: 10.1007/978-3-319-24258-3_12.

[39] R. Hosseini, P. Brusilovsky, and J. Guerra, “Knowledge Maximizer: Concept-based Adaptive Problem Sequencing for Exam Preparation,” in: *Proc. the 16th International Conference on Artificial Intelligence in Education (AIED 2013)*, Memphis, USA, pp. 848-851.

[40] J. Barria-Pineda, K. Akhuseyinoglu, S. Želem-Ćelap, P. Brusilovsky, A. Klasnja Milicevic, and M. Ivanovic, “Explainable Recommendations in a Personalized Programming Practice System,” in: *Proc. 22nd International Conference on Artificial Intelligence in Education, AIED 2021*, I. Roll, D. McNamara, S. Sosnovsky, R. Luckin and V. Dimitrova, Eds., Utrecht, The Netherlands, June 14-18, 2021, pp. 64-76.

[41] J. Barria-Pineda, K. Akhuseyinoglu, and P. Brusilovsky, “Adaptive Navigational Support and Explainable Recommendations in a Personalized Programming Practice System,” in: *Proc. 34th ACM Conference on Hypertext and Social Media*, Rome, Italy September 4 - 8, 2023, pp. 1-9, doi: 10.1145/3603163.3609054.

[42] J. Barria-Pineda, K. Akhuseyinoglu, P. Brusilovsky, K. Pollari-Malmi, T. Sirkiä, and L. Malmi, “Personalized Remedial Recommendations for SQL Programming Practice System,” in: *Proc. Workshop on Adaptation and Personalization in Computer Science Education at the 28th ACM Conference on User Modeling, Adaptation and Personalization, ACM UMAP 2020*, Genoa, Italy, Jul. 18, 2020, pp. 135-142, doi: 10.1145/3386392.3399312.

[43] National Academies of Sciences Engineering and Medicine, “Human-AI Teaming: State-of-the-Art and Research Needs,” Washington, DC: The National Academies Press, 2022.

[44] S. Bostandjiev, J. O'Donovan, and T. Höllerer, “TasteWeights: A Visual Interactive Hybrid Recommender System,” in: *Proc. 6th ACM Conference on Recommender System*, Dublin, Ireland, September 9, 2012, pp. 35-42.

[45] K. Verbert, D. Parra-Santander, P. Brusilovsky, and E. Duval, “Visualizing Recommendations to Support Exploration, Transparency and Controllability,” in: *Proc. the 2013 International Conference on Intelligent User Interfaces, IUI '2013*, Santa Monica, USA, March 19–22, 2013, pp. 351-362, doi: 10.1145/2449396.2449442.

[46] H. Cramer et al., “The effects of transparency on trust in and acceptance of a content-based art recommender,” *User Modeling and User-Adapted Interaction*, Vol. 18, No. 5, pp. 455-496, 2008.

[47] J. Barria Pineda, and P. Brusilovsky, “Making Educational Recommendations Transparent through a Fine-Grained Open Learner Model,” in: *Proc. Workshop on Intelligent User Interfaces for Algorithmic Transparency in Emerging Technologies at the 24th ACM Conference on Intelligent User Interfaces, IUI 2019*, Los Angeles, USA, March 20, 2019.

[48] J. Guerra, C. Schunn, S. Bull, J. Barria-Pineda, and P. Brusilovsky, “Navigation support in complex open learner models: assessing visual design alternatives,” *New Review of Hypermedia and Multimedia*, Vol. 24, No. 3, pp. 160-192, 2018, doi: 10.1080/13614568.2018.1482375.

[49] N. Tintarev, and J. Masthoff, “A Survey of Explanations in Recommender Systems,” in: *Proc. 3rd International* Workshop *on Web Personalisation, Recommender Systems and Intelligent User Interfaces at IEEE 23rd International Conference on Data Engineering (ICDE 2007)* Istanbul, Turkey, pp. 801-810.

[50] P. Brusilovsky, “AI in education, learner control, and human-AI collaboration,” *International Journal of Artificial Intelligence in Education*, Vol. 34, No. 1, pp. 122-135, 2024.

Author1 Photograph
(2.7cm *3.3cm)

Peter Brusilovsky is a Professor of at the School of Computing and Information, University of Pittsburgh where he also directs Personalized Adaptive Web Systems lab. Peter has been working in the field of personalized learning, student and user modeling, recommender systems, and intelligent user interfaces for over 30 years. He is a board member of several journals including the International Journal of AI in Education and User Modeling and User-Adapted Interaction.