Mastery Grids Platform for Personalized Practice

Peter Brusilovsky
University of Pittsburgh
peterb@pitt.edu

Kamil Akhuseyinoglu
University of Pittsburgh
kaa108@pitt.edu

Jordan Barria-Pineda
University of Pittsburgh
jab464@pitt.edu

KEYWORDS

personalization,interoperability, smart learning content,online practice system

ACM Reference Format:
Peter Brusilovsky, Kamil Akhuseyinoglu, and Jordan Barria-Pineda. 2019. Mastery Grids Platform for Personalized Practice. In *Proceedings of SPLICE'19.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Mastery Grids[1] is an open source learning platform [13], which provides access to multiple types of interactive learning content through a personalized interface. Mastery Grids interface offers a multi-facet social comparison [5], open learner modeling, adaptive navigation support, and personalized recommendation to help students access most relevant learning content. The personalization offered by the system takes into account course goals provided by the instructors and student current state of knowledge traced by the learner modeling component which is most appropriate. The current version of Mastery Grids is focused on creating *practice support systems* for computer science courses. The idea of a practice system to provide access to a wide variety of interactive content that students can explore on their own to gains knowledge and skills for a specific course. A practice system could be uses in or out of the class, but it is focused on learning by doing and knowledge modeling rather than assessment and grading. For each course, instructors could create their own installations of Mastery Grids interface by organizing the course courses info a sequence of topics and selecting most relevant smart content items of multiple types for each topic. The instructors could also start with one of the existing installations (see below) for Java, Python and SQL making necessary updates. The platform offers a drag-and-drop authoring system with a content recommendation component to support instructors in creating their own courses [6].

The interface of Mastery Grids is formed by a flexible topic grid. Each column represents one course topic (ordered along with their position in the course). The cells in the topic column provide access to various kinds of learning content for the topic while the coloring of these cells represent knowledge progress for the topic. Different views of the grid allows to view learner knowledge for the topic gained by working with each type of content, visualize knowledge progress of a peer group (for example, the whole class) and compare learner's own knowledge progress with the progress of the peer group. As shown in the figures below, higher level of knowledge of the topic are visualized by darker colors (green for the learner, blue for the peer group). A click on a topic opens a panel, which visualizes all smart learning content selected by the teacher for this topic. Content cells are also colored according to the knowledge progress level. The knowledge-based visualization on the topic and content level supports self-regulated learning and helps learners to focus on most relevant topics and items. A click to a specific content item provides an immediate access to this item for practice. On the top of adaptive navigation support offered by content and topic link annotations, the system also offers several kinds of content recommendation [10].

Mastery Grids is based on communication-based personalization infrastructure ADAPT2 (formerly Knowledge Tree [2]) and supported by adaptive social learning framework Aggregate. ADAPT2 supports smart content invocation with single sign-on, extensive data collection, and student modeling. Aggregate adds several kinds of open student modeling, social comparison, and recommendation.

2 EXISTING INSTALLATIONS

In this section we review several existing model installations of Mastery Grids developed for Java, Python, and SQL programming. A review of these installations helps to understand most critical features of the platform while also showing possible ways to start with a practice system for a course in one of the supported domains.

2.1 Integrated Python Practice System

The Python practice system known also as Python Grids is a collaboration between University of Pittsburgh PAWS Lab and Aalto University LeTech group [4] The Python Grids provides access to three types of interactive learning content developed by PAWS Lab and two types developed by LeTech lab. PAWS content includes parameterized problems focused on understanding code behavior delivered by QuizPET system and interconnected worked examples and challenges (faded examples) focused on program construction delivered by PCEX system [8]. LeTech content includes animated code examples and Parson's coding problems. More details about these kinds of content is provided in the next section.

The Python Grids[2] has been evaluated in several large-scale studies at Aalto University. A report of one of these studies and more details about the system could be found in [4].

2.2 Integrated Java Practice System

This system is a collaboration between the University of Pittsburgh PAWS Lab, Aalto University LeTech group, and University

[1] More information about Mastery Grids is available at PAWS wiki: http://adapt2.sis.pitt.edu/wiki/Mastery_Grids_Interface

[2] See a demo of the system at https://www.youtube.com/watch?v=roCkp7QVfB

SPLICE'19, August 11th, 2019, Toronto, ON, Canada
© 2019 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

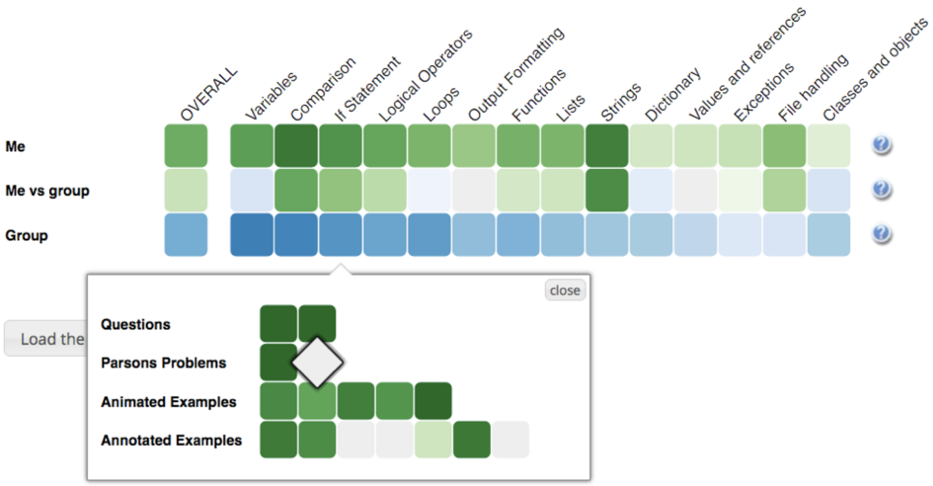

Figure 1: Integrated Python Practice System

of Toronto CSEd Group. Several versions of the Java installations provide access to different set of smart content. It provides access to integrated QuizJet parameterized problems, PCEX examples and challenges, LeTech animated examples, and coding problems devlieved by PCRS system developed at CSEd Group. Various versions of the Java Practice system were evaluated in several studies [1, 7, 13].

2.3 Integrated SQL Practice System

This system was developed in collaboration between University of Pittsburgh PAWS Lab and University of Canterbury Intelligent Computer Tutoring Group with contributions from Sibel Somyürek (Gazi Unversity) and Ayça Çebi (Karadeniz TU).

The SQL practice system provides access to SQL code examples delivered by WebEx system, SQL animated examples, and two types of SQL programming problems delivered by SQL-KnoT system developed at PAWS lab and SQL-Tutor from the University of Canterbury. The system has been evaluated in several studies. A report of a study and a more information about the system could be found in [5].

3 SMART CONTENT AVAILABLE IN MASTERY GRIDS

Mastery Grids could be used to provide access to a range of smart learning content [3] which is compatible either with ADAPT2 communication protocols [2] or IMS LTI protocol [12]. Protocol compliance is critical for Mastery Grids to provide access to requested learning items and receive the flow of information about student interaction with learning content to update the state of learner knowledge and history.

This section reviews a range of learning content types, which are readily available in the system and connected to its authoring interface. This content is created and connected with collaboration from several research group and typically delivered directly from a content server maintained by this group. The architecture of Mastery Grids support seamless delivery and knowledge tracking for learning content coming from different servers located in different countries. Due to its support of LTI protocol, the system can provide access to a much larger variety of content, however other types of content are not yet supported by the authoring system and concept-level student modeling component.

3.1 Smart content from Aalto University

The following types of learning content created by the LeTech lab at the Aalto University is available for the use with Mastery Grids. These types of content is compatible with both ADAPT2 and LTI protocols and delivered by the ACOS server [14]

- Animated examples for Java and Python
- Parson's problems for Python

3.2 Smart content from University of Pittsburgh

The following types of learning content created by the PAWS lab at the University of Pittsburgh is available for the use with Mastery Grids.

- Interactive code examples for C, Java, Python, and SQL served by WebEx system.
- Worked code examples and challenges for Java and Python served by PCEX system [8, 9].
- Parameterized code tracing problems for Java and Python delivered by QuizJET and QuizPET systems [11]
- SQL coding problems delivered by SQLKnoT system [5]
- SQL code animations [5]

3.3 Smart content from University of Toronto

Mastery Grids provides access to learning content created for PCRS system developed at the University of Pittsburgh of Toronto [15]. PCRS offers several types of problems for different languages. Currently, only Java problems are readily available.

REFERENCES

[1] J. Barria-Pineda, J. Guerra-Hollstein, and P. Brusilovsky. A fine-grained open learner model for an introductory programming course. In *26th Conference on*

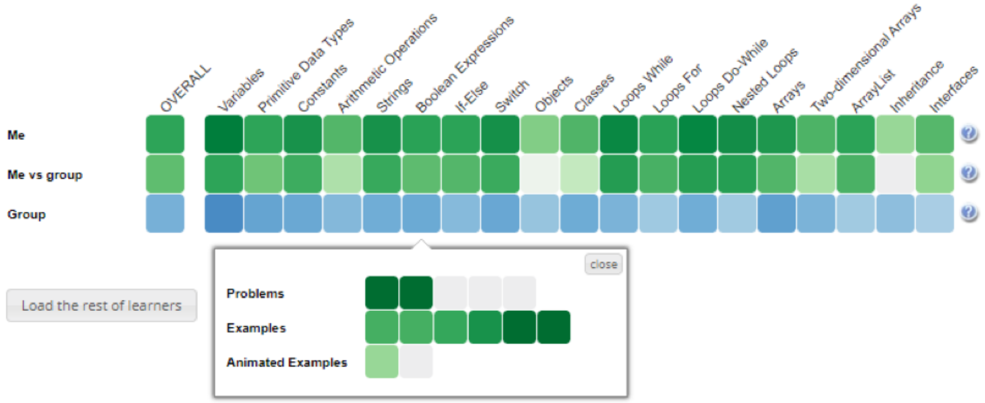

Figure 2: Integrated Java Practice System

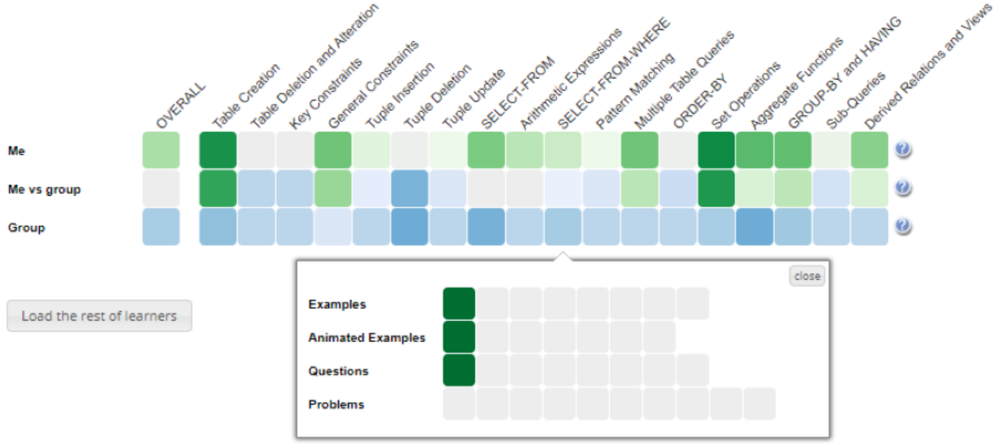

Figure 3: Integrated SQL Practice System

User Modeling, Adaptation and Personalization, UMAP ’18, pages 53–61. ACM, 2018.

[2] P. Brusilovsky. Knowledgetree: A distributed architecture for adaptive e-learning. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 104–113. ACM, 2004.

[3] P. Brusilovsky, S. Edwards, A. Kumar, L. Malmi, L. Benotti, D. Buck, P. Ihantola, R. Prince, T. Sirkiä, S. Sosnovsky, et al. Increasing adoption of smart learning content for computer science education. In *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference*, pages 31–57. ACM, 2014.

[4] 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*, 13(1):18, 2018.

[5] 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*, 4(3):450–461, 2016.

[6] H. Chau, J. Barria-Pineda, and P. Brusilovsky. Course-adaptive content recommender for course authoring. In V. Pammer-Schindler, M. Pérez-Sanagustín, H. Drachsler, R. Elferink, and M. Scheffel, editors, *13th European Conference on Technology Enhanced Learning, EC-TEL 2018*, Lecture Notes in Computer Science, pages 437–451. Springer, 2018.

[7] J. Guerra, R. Hosseini, S. Somyurek, and P. Brusilovsky. An intelligent interface for learning content: Combining an open learner model and social comparison to support self-regulated learning and engagement. In *the 21st International Conference on Intelligent User Interfaces (IUI ’16)*, pages 152–163. ACM, 2016.

[8] R. Hosseini. *Program Construction Examples in Computer Science Education: From Static Text to Adaptive and Engaging Learning Technology.* Doctoral dissertation, 2018.

[9] R. Hosseini, K. Akhuseyinoglu, A. Petersen, C. D. Schunn, and P. Brusilovsky. Pcex: Interactive program construction examples for learning programming. In *Proceedings of the 18th Koli Calling International Conference on Computing Education Research*, page 5. ACM, 2018.

[10] 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 G. Conole, T. Klobučar, C. Rensing, J. Konert, and Lavoué, editors, *10th European Conference on Technology Enhanced Learning (EC-TEL 2015)*, Lecture Notes in Computer Science, pages 155–168, 2015.

[11] 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*, 26(4):270–283, 2010.

[12] IMS Global Learning Consortium. Learning tools interoperability, 2015.

[13] T. D. Loboda, J. Guerra, R. Hosseini, and P. Brusilovsky. Mastery grids: An open source social educational progress visualization. In *European conference on technology enhanced learning*, pages 235–248. Springer, 2014.

[14] T. Sirkiä and L. Haaranen. Improving online learning activity interoperability with acos server. *Software: Practice and Experience*, 47(11):1657–1676, 2017.

[15] D. Zingaro, Y. Cherenkova, O. Karpova, and A. Petersen. Facilitating code-writing in pi classes. In *Proceeding of the 44th ACM technical symposium on Computer science education*, pages 585–590. ACM, 2013.