Concept-Level Knowledge Visualization For Supporting Self-Regulated Learning

Jordan Barria-Pineda
School of Information Sciences,
University of Pittsburgh,
Pittsburgh, PA, USA
jab464@pitt.edu

Julio Guerra
Instituto de Informática,
Universidad Austral de Chile,
Valdivia, Chile
School of Information Sciences,
University of Pittsburgh,
Pittsburgh, PA, USA
jdg60@pitt.edu

Yun Huang
Intelligent Systems Program,
University of Pittsburgh,
Pittsburgh, PA, USA
yuh43@pitt.edu

Peter Brusilovsky
School of Information Sciences,
University of Pittsburgh,
Pittsburgh, PA, USA
peterb@pitt.edu

Abstract

Mastery Grids is an intelligent interface that provides access to different kinds of practice content for an introductory programming course. A distinctive feature of the interface is a parallel topic-level visualization of student progress and the progress of their peers. This contribution presents an extended version of the original system that features a fine-grained visualization of student knowledge on the level of the detailed concepts that are associated with the course. The student model is based on a Bayesian-network which is built using students performance history in the learning activities.

Author Keywords

Student Modeling; Open Student Model; Social Comparison; Information Visualization; Competency Visualization

ACM Classification Keywords

H.5.4 [Information interfaces and presentation]: Navigation

Introduction

Mastery Grids [4] is an intelligent interface that combines open student modeling (OSM), social comparison, and adaptive navigation support to provide access to different kinds of practice content for an introductory programming course. It has been shown that this tool has positive effects on both student engagement and efficiency [1, 4]. This pa-

per presents an extended version of Mastery Grids that features both a coarse-grained topic-level student progress visualization and a fine-grained concept-level knowledge visualization. While the original interface focused mostly on progress visualization, our experience indicated that showing just the progress of learners on the level of learning activities was not enough. As indicated in the literature on self-regulated learning [3], some students tend to focus more on thoroughly acquiring the knowledge taught (mastery-oriented students) rather than focusing on just achieving the completion of each one of the learning stages (performance-oriented students). To support both kinds of learners, it is necessary to provide more fine-grained level feedback to all students about their learning progress and knowledge.

Mastery Grids: The Interactive Visual Interface

Topic-Level Progress Visualization
The original Mastery Grids interface [4] focuses on visualizing student progress on the level of course topics. The core of the visualization is a grid of cells where the horizontal dimension represented the progression of course topics and the color density of a cell represented the level of knowledge gained on the topic: the darker the color, the higher the knowledge (Figure 1 A). The green cells represent the learner's own progress, while the blue cells represent the progress of a group of peers (e.g. the whole class or a subgroup of students). The topic knowledge was calculated by analyzing student progress within the activities belonging to each topic. These activities could be accessed by clicking on the topic cell and moving to the "activity level" (Figure 1 B). While topic/activity progress tracking is useful to students who wish to understand their overall progress and to identify topics that need work, it is not sufficient to understand knowledge of domain concepts or to provide guidance on selecting an activity to work on. The concept-level
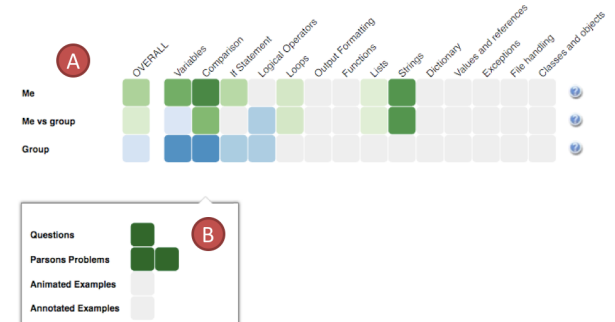

Figure 1: Previous version of Mastery Grids.

extension of Mastery Grids combines topic-based progress visualization with concept-level knowledge visualization for knowledge monitoring and activity selection, which are critical components of self-regulated learning.

Concept-Level Knowledge Visualization
This new version of the system presents student concept-level knowledge visualization as a bar chart, in which each bar represents a programming concept, such as *"For Loop"* (See Figures 3a and 3b). The height and the color density of each bar indicates the estimated level of knowledge of the user (in green) or of the group (in blue): the longer and darker the bar, the higher the level of estimated knowledge. These concepts are grouped and sorted in the x-axis by topic, according to the order in which topics are covered in the course. Within the overarching group of topic concepts, the topics are ordered by their inferred level of knowledge. The main interaction features of the visualization are:

1. Exploring topic-concept relationships: When students mouse over a grid cell that represents a course topic, the interface highlights the concepts that the topic covers (Figure 3a). Students can check their estimated current knowledge of the related concepts and map their observed strengths

or weaknesses in a specific topic to the presence or absence of knowledge in specific concepts.

2. Exploring the activity-concept relationship: When clicking in a topic, the set of learning activities associated are shown and the concept-level knowledge visualization can be used to examine concepts covered by each activity (problem, animated example, and others). When the student mouses over an activity cell, the concepts that can be practiced by performing this activity are highlighted in yellow in the bar chart (Figure 2). Note that when working with a topic, the core concepts of the current topics are always shown (Figure 3b). As a result, the students can see to which extent each activity can contribute to master the current topic, to bridge the gaps in their knowledge, or to connect to concepts in other topics. With this visualization, students obtain a kind of "x-ray vision" to select the most appropriate activity to practice.
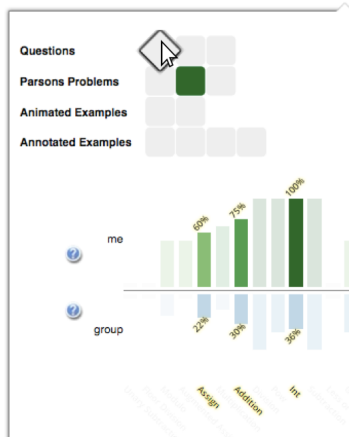
The Difficulty Gauge and the Learning Gauge
We added an additional visual feature, in the form of a *gauge* within the activity-level context, with the goal of easing the understanding of the information. The gauge has two versions: the *Difficulty Gauge* shows to which extent the highlighted activity is easy or hard for the student (top of Figure 4), and the *Learning Gauge* estimates how much learning the student will acquire by doing the activity (bottom of Figure 4). Both values are computed using the number of *known* (mastered), *familiar* (learned but not mastered), and *new* concepts (those that are not yet learned).

The Student Model

Students' concept-level knowledge is inferred, based on two components. The main component is a Bayesian network (Figure 5) that models a student's general knowledge on each concept across different application con-
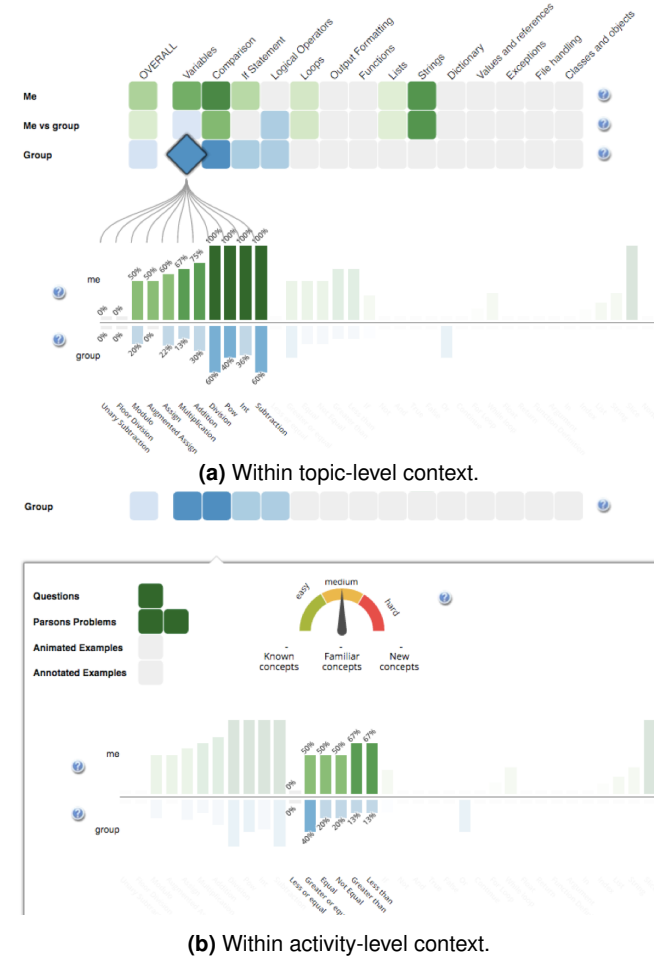

Figure 2: Details of the concepts that can be practiced if a student addresses the hovered activity

(a) Within topic-level context.

(b) Within activity-level context.

Figure 3: Concept-Level Knowledge Visualization.

texts (*context-general knowledge*). This network, called *conjunctive knowledge modeling (CKM)*, was proposed in our prior work [5]. The binary O nodes represent observed performance on exercises (correct/incorrect), and the binary K nodes represent latent knowledge levels of concepts

(learned/unlearned). The multiple-to-multiple relation between concepts and exercises was specified by experts beforehand. We learn parameters based on existing data and subsequently perform dynamic knowledge updates in the real system.

The second component models a student's specific knowledge in applying a concept in each important application context (*context-specific knowledge*). For example, for the concept *"WhileStatement"*, we construct a combination *"WhileStatement+AddAssignment"* to monitor student performance on exercises by using add assignment within a while loop. If a student has at least one correct attempt on at least half of the exercises that require a specific combination, this combination is marked as *learned*. Then, we compute the context-specific knowledge of a concept as the ratio of its combinations that are marked as learned. This idea is based on our prior work [5], which demonstrated that incorporating such combinations could improve mastery inference accuracy and guide students to a more varied practice. To obtain the final knowledge level on a concept, we compute the average of both the context-general and context-specific knowledge.

The parameters were learned from data collected from previous classroom studies (2013-2015) at the University of Pittsburgh (256 students). The BN contains 100 concepts and 102 exercises, and was implemented based on SMILE [2].

Evaluation In-Progress

We performed two controlled experiments (within-subject designs) to investigate the effects of the different visual features and the perception of the user model, and conducted two longitudinal classroom studies (between-subject designs) in a course of Java programming and a course of Python programming. In these studies, the treatment

groups experienced different sets of visual features that combined the social comparison features, the learning and difficulty gauges, and the concept visualization. We expect to observe different navigational patterns within the system among the groups, as well as different levels of engagement.

REFERENCES

1. P. Brusilovsky, S. Somyurek, J. Guerra, R. Hosseini, V. Zadorozhny, and P. Durlach. 2016. Open Social Student Modeling for Personalized Learning. *IEEE Transactions on Emerging Topics in Computing* 4, 3 (2016), 450–461.

2. M. Druzdzel. 1999. SMILE: Structural Modeling, Inference, and Learning Engine and GeNIe: a development environment for graphical decision-theoretic models. In *Proc. 16th National Conf. on Artificial Intelligence (AAAI)*. 902–903.

3. A. Elliot and K. Murayama. 2008. On the measurement of achievement goals: Critique, illustration, and application. *Journal of Educational Psychology* 100, 3 (2008), 613–628.

4. J. Guerra, R. Hosseini, S. Somyurek, and P. Brusilovsky. 2016. An Intelligent Interface for Learning Content Combining an Open Learner Model and Social Comparison to Support Self-Regulated Learning and Engagement. In *Proceedings of the 21st International Conference on Intelligent User Interfaces - IUI '16*. 152–163.

5. Y. Huang, J. Guerra, and P. Brusilovsky. 2016. Modeling skill combination patterns for deeper knowledge tracing. In *Proc. 6th Workshop on Personalization Approaches in Learning Environments (PALE)*. The 24th conference on User Modeling, Adaptation, and Personalization (UMAP).
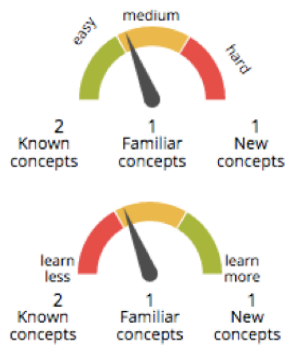
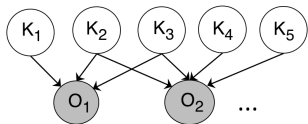**Figure 4:** Difficulty Gauge (top) and Learning Gauge (bottom)

Figure 5: Bayesian network structure