What Should I Do Next? Adaptive Sequencing in the Context of Open Social Student Modeling

Roya Hosseini[1], I-Han Hsiao[2], Julio Guerra[3], and Peter Brusilovsky[3]

[1] Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA, USA
roh38@pitt.edu
[2] School of Computing, Informatics & Decision Systems Engineering, Arizona State University, Tempe, AZ, USA
sharon.hsiao@asu.edu
[3] School of Information Sciences, University of Pittsburgh, Pittsburgh, PA, USA
{jdg60,peterb}@pitt.edu

Abstract. One of the original goals of intelligent educational systems was to guide each student to the most appropriate educational content. In previous studies, we explored both knowledge-based and social guidance approaches and learned that each has a weak side. In the present work, we have explored the idea of combining social guidance with more traditional knowledge-based guidance systems in hopes of supporting more optimal content navigation. We propose a greedy sequencing approach aimed at maximizing each student's level of knowledge and implemented it in the context of an open social student modeling interface. We performed a classroom study to examine the impact of this combined guidance approach. The results of our classroom study show that a greedy guidance approach positively affected students' navigation, increased the speed of learning for strong students, and improved the overall performance of students, both within the system and through end-of-course assessments.

Keywords: personalized guidance, open social student modeling, adaptive navigation support, E-learning, Java programming

1 Introduction

One of the original goals of intelligent educational systems was to guide each student to the most appropriate educational content. Starting with the first reported ITS system SCHOLAR [7], a range of knowledge-based guidance technologies have been reported. Different technologies in this group include instructional planning [1], course sequencing [3], course generation [14], and adaptive navigation support [2]. All these knowledge-based approaches were based on the same principles: by using a combination of domain models, course goals, and overlay student models, the sequencing engine decides which content is the most appropriate for an individual student at any given moment and delivers it to the student through the interface, which either directly brings the student to the right content (as in sequencing), or delivers the content through suggested links (as in course generation and navigation support). Despite the known power of this technology, there are few practical applications, due to the large amount of effort required to build the domain models and analyze the content.

In our recent research, we discovered and evaluated a new approach to guide students to the "right" content, based on the ideas behind open social student modeling (OSSM) [10]. OSSM is a recent expansion of open student modeling (OSM), a popular approach that makes traditionally hidden student models available for students to explore [5,11,13]. OSM is known for its ability to increase student engagement, motivation, and knowledge reflection. The goal of OSSM is to integrate its cognitive aspects with social aspects by allowing students to explore each others' models or to view a cumulative model of the class [4]. In our studies, we explored several versions of a visual OSSM, based on comparative visualization of the student's own open knowledge model and the models of students with similar learning goals. While our original goal was to increase student engagement, which is a known value of social approaches, our studies have also demonstrated the navigational support power of OSSM. Our system was able to guide students to the most appropriate self-assessment problems [10] almost as efficiently as the knowledge-based guidance that we previously explored [9]. Since the main power of OSSM comes from the overall community of learners, it also requires considerably simpler domain and user models in order to reach maximum efficiency. However, our studies also revealed that the OSSM approach can make students more conservative in their work with content, which decreases the 'personalization' power of such 'social' guidance.

This paper explores the idea of combining social guidance with more traditional knowledge-based guidance in the hope of supporting more optimal content navigation. This idea was motivated by the success of hybrid approaches in recommender systems that demonstrated several efficient ways to combine content-based and collaborative filtering approaches [6]. We introduce a greedy sequencing approach for selecting learning activities that could maximize student's level of knowledge, and demonstrate the ways in which this approach could be implemented in the context of OSSM. We also present a classroom study that examines the added impact of this combined guidance approach.

The remainder of this paper presents the sequencing approach and its implementation in the OSSM interface and reports the results of the evaluation. We conclude with a discussion of the results and plans for future work.

2 Adaptive Sequencing in the Context of OSSM

In our study, adaptive guidance was implemented in the context of a specific OSSM interface called Mastery Grids. To explain the technology, we will start with a brief presentation of Mastery Grids, follow by explaining how the suggestions generated by the sequencing algorithm were added to the OSSM interface, and finally explain the details of our specific sequencing approach, which we call Greedy Sequencing.

2.1 Mastery Grids, an OSSM Interface

Mastery Grids is an OSSM interface that combines a visual open student model presentation with an interface to access online course materials. The design of Mastery Grids was informed by our earlier studies of OSSM [10], where we discovered that students achieve higher success rates and engage with non-mandatory content more frequently in the presence of OSSM. A classroom study confirmed these effects for the first version of Mastery Grids [12].

Figure 1 shows a screenshot of the Mastery Grids interface. The system organizes course content into topics, which are displayed as columns of the grid. The first row shows the current student's topic-by-topic knowledge progress by using shades of green in different densities; the darker the color, the higher the progress. The third row shows the aggregated progress of the rest of the students of the class in shades of orange. The second row presents a differential color that compares the current student's progress to the overall class progress. For example, in Figure 1, the student shows a higher progress than the rest of the class in most of the topics where the cells in the second row are green, but the class has advanced farther in two of the topics (13th and 20th column) where the cells in the second row are orange. The student has the same progress as the class in the four topics shown in a light gray color (11th, 15th, 18th, and 19th column). By clicking in a cell, the student can access the content that falls inside the topic. For example, in Figure 1, the student has clicked the topic *Classes*, and the system displays cells to access questions and examples related to this topic. Additionally, by clicking the button "Load the rest of learners", a grid shows an anonymized, ranked list of individual student models (Figure 2).

2.2 Enhancing OSSM Interface with Sequencing

To implement adaptive sequencing in the context of the Mastery Grids interface, we used the top three content item recommendations generated by the adaptive sequencing approach and displayed their presence in the topic using red stars that appear on both recommended items and the topics that they contain. The size of the stars shows the position of the recommended items in the top-three list. Note that our approach to *sequencing* is consistent with the navigational support nature of the interface: it does not force students to go to the sequenced content, but simply informs the students and helps them to make their next navigational step. The resulting interface combines the social guidance of OSSM with the personal guidance that sequencing provides.

2.3 Greedy Sequencing

The intelligence behind the sequencing interface is provided by a sequencing algorithm that we call *greedy sequencing* (GS). This algorithm was specifically developed to compensate for the conforming nature of the OSSM on student navigation. The goal of GS is to guide students through learning materials by proactively recommending student activities that could *maximize* the chance to

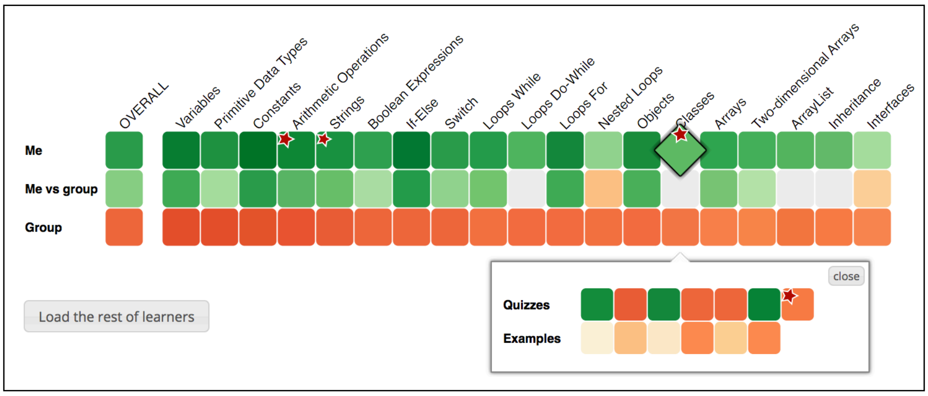

Fig. 1. The presentation of recommendations in the context of Mastery Grids' OSSM interface; a cell with a star symbol represents a recommended item

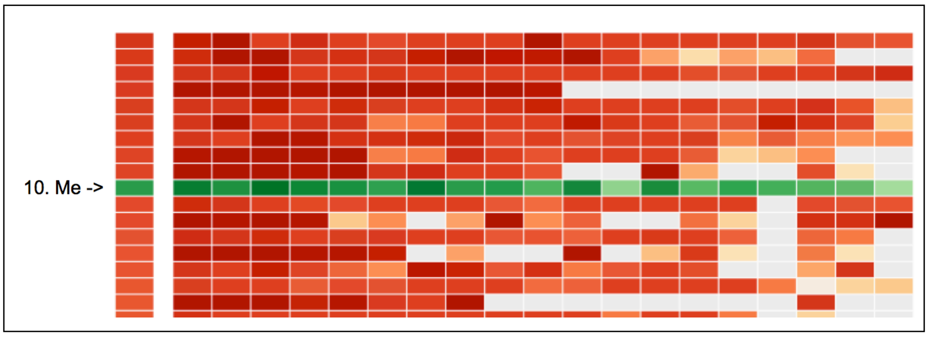

Fig. 2. List of peer models ordered by progress on topics in the course. The student with the highest progress appears at the top of the list. This list is anonymized and the current student can see herself in the position that she has obtained, according to her topic-based progress. In this case, the student, who is shown by the label "Me", is in position 10.

gain new knowledge while avoiding content that is too complex for the student to comprehend. As with other knowledge-based sequencing approaches, GS uses information about concepts associated with content: more specifically, it focuses on prerequisite and outcome concepts for each activity. Prerequisites are the concepts that students need to master before starting to work with a given activity. Outcomes are the concepts that are being learned in the process of work with the activity. In our work, all concepts associated with an activity were determined using our concept parser [8]. The parser indexes the activities with concepts of Java ontology[1]. The extracted concepts for each activity are then separated into prerequisites and outcomes. In the activity, we marked a concept as a prerequisite if it has appeared in prior topics, and as an outcome if it is the first topic where the concept appears.

The GS algorithm ranks activities by balancing the knowledge level of each student in the prerequisite concepts with the knowledge that can be gained from the outcome concepts. The rank of an activity is calculated using (1) based on the student's level of knowledge in the prerequisite and outcome concepts of that activity:

$$R = \frac{n_p P + n_o O}{n_p + n_o} \quad (1) \qquad P = \frac{\sum_i^{n_p} k_i w_i}{\sum_i^{n_p} w_i} \quad (2) \qquad O = \frac{\sum_i^{n_o} (1 - k_i)\, w_i}{\sum_i^{n_o} w_i} \quad (3)$$

where n_p and n_o are the number of prerequisite and outcome concepts in the activity, respectively; P represents the amount of known prerequisites and is calculated as the weighted average of student's knowledge in the prerequisite concepts of the activity; and O represents the amount of unknown outcomes and is calculated as the weighted average of knowledge that is not learned in each of the outcome concepts. These two variables can be calculated using (2) and (3), respectively.

In (2) and (3), k_i is the student's level of knowledge of the concept i, has the minimum value of 0 (no knowledge) and asymptotically reaches 1 (maximum knowledge). The term $1-k_i$ in (3) is the amount of knowledge that is not learned in the outcome concept i. The w_i is the smoothed weight of the concept obtained by performing a *log* function on TF-IDF values of the concepts. The rank R of an activity lies in the interval $[0, 1]$ with 1 representing the highest rank.

3 Study Design

To explore the effect of GS on student navigation and performance, we ran a classroom study in an undergraduate course of Object-Oriented Programming & Data Structures offered by the Computer Science Engineering program of the Arizona State University in fall 2014. The course focused on data structures with Java. In this course, the Mastery Grids interface extended with GS was used to

[1] `http://www.sis.pitt.edu/~paws/ont/java.owl`

access supplementary course materials. A total of 143 students were enrolled in the course. The instructor informed the students about the learning materials that could be accessed using the Mastery Grids interface. The instructor encouraged students to explore this content, but indicated that the use of this system was not mandatory.

To investigate how students navigated both with and without the presence of the sequencing, we split the course into two parts. Part 1, from Aug. 21 to Sep. 25, used the Mastery Grids system with no sequencing. In part 2, from Sep. 26 to Oct. 21, sequencing was enabled. At the beginning of the course, students took a pretest to evaluate their initial knowledge of Java programming concepts. To measure the students' knowledge gain, a post-test was administered on Oct. 21. The pretest and post-test had the same set of questions and the score ranged from 0 to 21. At the end of the semester, we collected questionnaires that asked students to report their opinions on the sequencing used in the Mastery Grids system.

The course's learning materials included parameterized questions on the semantics of Java, administered by the QuizJET system [9], and annotated code examples, administered by the Webex system. The parameterized nature of semantics questions allowed students to attempt to answer the same question several times, each time with a different parameter. As a result, the correct answer is different across attempts on the same question. An annotated code example is a complete program that has an expert's annotation (comments and explanations) for some lines of code. Students could interactively explore these annotations by clicking on the annotated lines. The learning materials were organized into topics defined by the course instructor. Overall, the course contained 111 questions and 103 examples spread over 19 topics.

4 Evaluation

We collected student logs for the analysis period between the pretest and the post-test. The data consisted of students' attempts at topics and activities, as well as information that showed whether attempted topics and activities (questions or examples) were recommended by the system or not. We removed all sessions with a duration of less than 30 seconds from the data. Then, we excluded students that were not sufficiently active in the system by discarding the data of those who had fewer than 30 attempts on questions, i.e. about $\frac{1}{4}th$ of the available questions. In total, 86 students used the system during the analysis period. Out of this number, there were 21 students with no attempt to solve questions and 12 students with less than 30 attempts on questions. After discarding data from these less active students, we had data from 53 students for our analysis.

4.1 Navigational Pattern Analysis

While the OSSM interface demonstrated a good ability to move the students along the common path through the topic sequence in a timely fashion, the goal of the GS algorithm was to help the students in breaking out from the common path when it is personally beneficial and to have students not spend too much time on topics that they have already sufficiently mastered, while also making sure that knowledge from previous topics is adequately mastered. To see to what extent the GS encouraged non-sequential navigation, we classified students' moves from current to next activity into four groups (patterns):

– Within-Topic: moving between activities in the same topic
– Next-Topic: moving from an activity in a topic to the activity in the next topic (according to the sequence of topics in the course)
– Jump-Forward: jumping to an activity in a topic that is two or more steps further away from the current topic
– Jump-Backward: jumping to an activity in an earlier topic

The Within-Topic and Next-Topic groups represent sequential navigation, and the Jump-Forward and Jump-Backward groups represent non-sequential navigation.

Table 1 shows the frequency of each pattern in the three contexts: part 1 (with no sequencing) and part 2 (with sequencing) separating student navigation to *Not-recommended* (part 2-N) and *Recommended* activities (part 2-R). The relative frequencies of the four patterns in each context are shown in Figure 3. The value in each cell is the probability (relative frequency) of the corresponding pattern in the corresponding context. A light blue color in the cell denotes a lower probability and a dark blue cell denotes a higher probability.

According to this table, when students make navigation decisions without sequencing (Part 1) or ignore it entirely (Part 2-N), they mostly follow a sequential pattern, working Within-Topic until they feel that their knowledge is sufficient and then moving to Next-Topic. This shows that students tend to attempt most of the activities in the topic before moving to the next topic, even if it is not the best strategy to improve their overall knowledge. The OSSM does hint to the students when it's time for them to move, but its guidance is quite conservative, since it is defined by the class as a whole. On the other hand, when students follow GS recommendations, their "groupthink" stay on the current topic shortens considerably, as students move to the next topic more quickly and expand their non-sequential navigation. There is good evidence in our study that the GS

Table 1. Frequencies of the four topic-based navigational patterns in part 1 (with no sequencing), and part 2 (with sequencing). Part 2-N and Part 2-R represent activities in part 2 that were Not recommended and Recommended, respectively.

Pattern	Part 1	Part 2-N	Part 2-R
Within-Topic	1801	4569	451
Next-Topic	431	689	189
Jump-Forward	216	287	162
Jump-Backward	219	328	161
Total	2667	5873	963

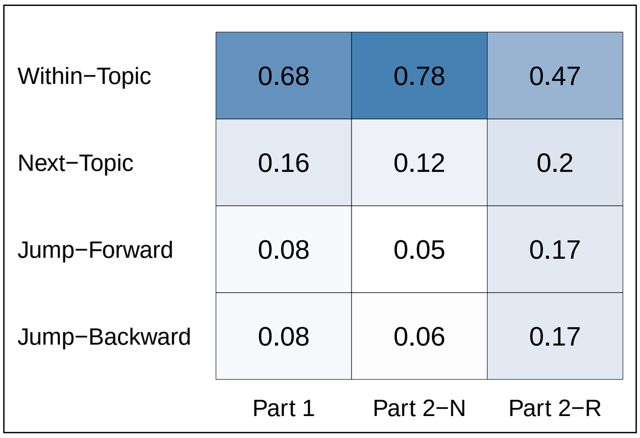

Fig. 3. Relative frequencies of four topic-based navigational patterns in part 1, not recommended items in part 2-N, and recommended items in part 2-R

promotes non-sequential navigation. However, we cannot conclude whether following the recommendations made by sequencing could benefit learning by more efficiently directing students to relevant activities. We examine this question in the next section.

4.2 The Value of GS: Amount of Learning and Speed

The mere presence of personalized guidance is not sufficient to provide an impact: what matters is whether the students choose to follow the guidance or to ignore it. We examined the added value of GS by comparing the *amount of learning* and *learning speed* of students who did not follow the guidance *(non-followers)* to the ones who did *(followers)*. To achieve this goal, we used normalized learning gain and learning speed as our evaluation measures. The normalized learning gain ($nGain$) is defined as the actual gain divided by the possible gain and is obtained using the score of the student on both the pretest and post-test. The speed of learning is defined as the ratio of normalized learning gain to the number of questions student attempted between the pretest and post-test ($nGain/n_q$). We multiplied this number by 100 to express it as a percentage ($\%speed$). To separate *non-followers* from *followers*, we calculated the *following ratio* per student that represents the fraction of activity accesses made when following recommendations. This ratio considers attempts on questions made in the second part of the study, when sequencing was available.

Figure 4 shows the distribution of the *following ratio*. As we can see from the skewed distribution, most of the students have a ratio of 0.2 or less; namely, they had followed a recommendation in less than $\frac{1}{5}th$ of their attempts. We selected $\frac{1}{5}$ as the cut-off for separating *non-followers* from *followers*. The *non-followers* group consists of 36 students with a following ratio of less than $\frac{1}{5}$

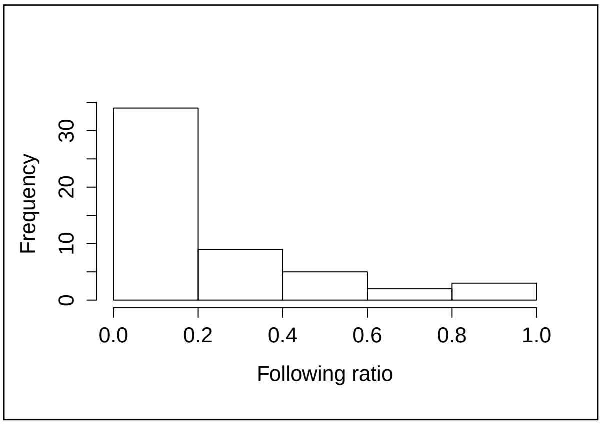

Fig. 4. Histogram of the following ratio of the students who participated in the study

and the *followers* group contained 17 students with a *following ratio* of greater than or equal to $\frac{1}{5}$. There were 8 students in the *non-followers* group and 6 in the *followers* group who either had a missing pretest or post-test, or seemed not motivated to work seriously on the post-test, as they got a lower score on the post-test than they did on the pretest. We filtered out those students and finally had 28 and 11 students left in the *non-followers* and *followers* group, respectively. We found that there were no significant differences between the groups, as far as the normalized learning gain. The speed of learning was higher among the *followers* ($M = 0.97\%, SD = 0.88\%$) than the *non-followers* ($M = 0.54\%, SD = 0.27\%$) but only reached borderline significance when compared to the *non-followers* group ($p = .083$ using a Welch t-test).

Since learning gain and learning speed might vary across students with different prior knowledge, we also separately compared *followers* and *non-followers* with low and high prior knowledge. If the pretest was less than the median of pretest scores, i.e. 11, a follower/non-follower was labeled as *low pretest*, otherwise it was labeled as *high pretest*. Table 2 provides a more detailed summary of these two parameters, as *followers* and *non-followers* within the low and high pretest score groups. The t-test was used in all of the comparisons, since parametric statistics assumptions were held.

We found that the mean of the normalized learning gain was not significantly different across *non-followers* and *followers* with low or high pretest scores, but that the speed was much higher for *followers* and reached a significant difference for students with high pretest scores. This implies that the GS may provide an efficient guidance that leads to a shorter learning path, at least for students with higher pretest scores. While this result seems promising, we have to account for other possible explanations given the design of our study. For example, since students were not randomly assigned to conditions, it is possible that the students who followed recommendations were more diligent students, so

Table 2. Mean±SD of evaluations measures for *non-followers* & *followers* separated by pretest group

	Low pretest (n=20)			High pretest (n=19)		
	Non-followers (n=14)	Followers (n=6)	p-value	Non-followers (n=14)	Followers (n=5)	p-value
ngain	0.51±0.28	0.42±0.19	.440	0.48±0.26	0.46±0.29	.870
%speed	0.55%±0.34%	0.97%±0.88%	.128	0.54%±0.27%	1.02%±0.70%	.039 *

Significance level * :< .05

that their improved performance was due to a selection effect known as 'selection bias.' For this reason, the above analysis needs another evaluation, and we hope to address this concern in a future study.

4.3 The impact of the GS on System and Class Performance

To see the effect of attempts suggested by sequencing on student performance, we fitted mixed models to predict the performance of the students in (1) attempts on self-assessment questions in the system (*in-system performance*), and (2) final exam taken at the end of the term (*out-of-system performance*). In all models, a random effect was included to account for unobserved variations between students. The models used the filtered data that had attempts from 53 active students (see Section 4).

To identify the influence of the GS on student *in-system performance*, we explored whether the student had a higher chance to answer the question correctly if it was suggested by GS. The variables of interest were (1) *correctness of attempt*, a binary variable that shows a correct or incorrect answer; and (2) *attempt type*, which shows whether an attempt was offered by sequencing or not. We fitted a logistic mixed effects model with *attempt type* as the fixed effect and *correctness of attempt* as the response variable. The collected data in part 1 (no sequencing) and part 2 (with sequencing) of the study consisted of a total of 5760 attempts on questions, from which 5275 were not offered by the GS and 485 were offered by the GS. The results indicated that the *attempt type* was a significant predictor of overall correctness ($\chi^2(1, 5760) = 14.17, p < .001$). The success was more frequent for questions recommended by the GS: the odds of having the correct answer when a question was offered by GS was 1.59 ($SE = 0.19$) times the odds of having the correct answer when a question was not offered by GS. This indicates that the GS guided students to questions at the proper difficulty.

To identify the influence of the GS on student *out-of-system performance*, we explored how the work in the system affected the score on the final exam, which ranges from 0 to 100. To address this question, we used the filtered data and separately counted the total number of attempts on activities both recommended by the GS and not recommended by the GS made by 40 students who had taken the final exam and used the system. We considered mixed models for predicting the score with a different set of predictors: (NQ), the total number of attempts on questions not recommended; (NQ_{GS}), the total number of attempts on questions

Table 3. Summary of the model fits for predicting student scores on the final exam

Model A		Model B	
	$\beta \pm SE$		$\beta \pm SE$
$Intercept$	68.50±6.24***	Intercept	68.84±5.37***
NQ	0.11±0.06	NA	0.06±0.03*
NQ_{GS}	0.69±0.30*	NA_{GS}	0.56±0.24*

Significance level * :< .05;** :< .01;*** :< .001

recommended by the GS; (NA), the total number of attempts on activities (questions or examples) not recommended; and (NA_{GS}), the total number of attempts on activities (questions or examples) recommended by the GS. Table 3 reports a summary of the estimated effects for the two fitted models: A and B. An interesting finding was that in Model A, the total number of questions accessed by the recommendations of the GS (NQ_{GS}) was significantly related to the student's final exam score. Attempting one question recommended by the GS was associated with an increase of 0.69 (or 0.69%) in the final exam score ($SE = 0.30, p = .019$). Model B also showed significant support for both the total number of attempts on activities that were not recommended (NA) and activities that were recommended by the GS (NA_{GS}): attempting one activity recommended by the GS was associated with an increase of 0.56 (or 0.56%) in the final grade ($SE = 0.24, p = .017$). At the same time, attempting one activity that was not offered was associated with a much lower increase of 0.06 in the final score ($SE = 0.03, p = .045$). In other words, working on both recommended and not recommended activities positively influenced the student's final exam score; however, the impact of activities that were recommended by GS was about 9 times greater than the activities that were not recommended.

5 Subjective Evaluation

At the end of the term, we administered a questionnaire that consisted of six questions about the recommendation features in the Mastery Grids system with answers in a 5-point Likert Scale (1:Strongly Disagree to 5:Strongly Agree). The questions are listed in Table 4 and the distribution of the answers is shown in Figure 5(a). Out of 95 students who participated, we kept only the answers of 51 students who used the system at least once.

As the data shows, students seemed to agree that they like to receive recommendations ($Q1 : M = 4.10, SE = 0.11$) and that the use of red stars to represent recommendations was clear ($Q2 : M = 3.86, SE = 0.14$). They also disagreed that recommendations were distracting ($Q5 : M = 2.41, SE = 0.15$). At the same time, it was less clear to them why some content was recommended ($Q4 : M = 3.82, SE = 0.15$), and they were interested to know the reasons why such recommendations were made ($Q6 : M = 4.20, SE = 0.11$). When we made a more detailed comparison between *followers* and *non-followers*, we noticed that *followers* ($M = 4.60, SE = 0.131, N = 15$) were even more curious

Table 4. Subjective evaluation questions

#	Question
1	In general, I would like the system to recommend me topics & content to focus on
2	It was clear to me that red stars were recommendations
3	Recommendations that I received this semester in Mastery Grids were useful for me
4	I could not understand why some topics and content areas were recommended to me
5	Recommendations distracted me from planning my work
6	It would be useful to see why some topics or content areas were recommended to me

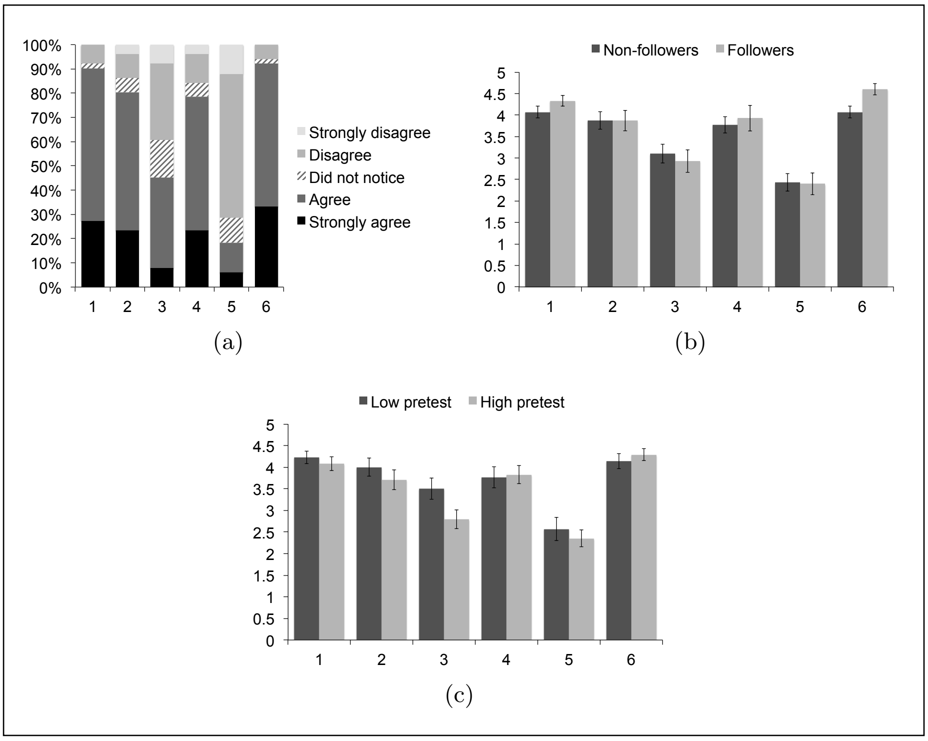

Fig. 5. (a) Distribution of answers by question, (b) Average score by question for followers and non-followers, (c) Average score by question for pretest groups

than *non-followers* ($M = 4.07, SE = 0.135, N = 30$) to know why some topics or contents were recommended (see Figure 5(b)). This difference was significant when using a Mann-Whitney test ($U = 133.5, p = .012$).

Furthermore, we found that while the average class opinion was somewhat neutral on the overall usefulness of the recommendations ($Q3 : M = 3.06, SE = 0.16$), students with low pretest scores gave significantly higher scores to the usefulness of the sequencing ($Q6 : M = 3.50, SE = 0.24, N = 22$) than students with high pretest scores ($Q6 : M = 2.79, SE = 0.22, N = 24$). This difference was also significant when using Mann-Whitney test ($U = 173.5, p = .037$) (see Figure 5(c)). This is an indication that the GS guidance helped students with lower scores as well.

6 Discussion and Future Work

This paper investigated the added value of knowledge-based guidance in the context of open social student modeling (OSSM). We presented a greedy sequencing (GS) approach that attempted to maximize student knowledge and demonstrated how it was implemented through the use of Mastery Grids, an OSSM interface for accessing learning materials. The evaluation of this combined approach provided several interesting findings.

The proposed approach encouraged non-sequential navigation patterns that guided weaker students to not-mastered materials in previous lectures and advanced stronger students to master materials in future lectures. As a result, it increased the learning speed of stronger students, which led these students to more optimal content navigation. In addition, we observed that the amount of work with materials selected by a proposed approach was associated with achieving considerably higher scores on the final exam. Although this does not mean that the proposed approach caused a higher grade on the final exam in every case, it still shows promising perspectives that could be further explored in future studies.

In future studies, we hope to address the limitations present in this study. First, it was focused on the domain of Java programming. Although the proposed GS approach can be adapted to other domains, more research is required before the findings of this study could be generalized. Second, the subjects in our study were undergraduate students who already knew the basics of Java programming. This could, in fact, explain the reason why relatively few students followed the guidance in our study. We need to plan a future study in an introductory Java course, where sequencing assistance is likely to be more critical. Finally, the survey report demonstrated that the interface needs to be modified in order to encourage students to follow recommendations. We would also like to increase the transparency of the proposed approach by increasing student awareness of the reasons to recommend specific learning content.

References

1. Brecht, B., McCalla, G., Greer, J., Jones, M.: Planning the content of instruction. In: Proceedings of 4-th International Conference on AI and Education, Amsterdam. pp. 24–26 (1989)
2. Brusilovsky, P.: Adaptive navigation support. In: The adaptive web, pp. 263–290. Springer (2007)
3. Brusilovsky, P.L.: A framework for intelligent knowledge sequencing and task sequencing. In: Intelligent tutoring systems. pp. 499–506. Springer (1992)
4. Bull, S., Britland, M.: Group interaction prompted by a simple assessed open learner model that can be optionally released to peers. In: Proceedings of Workshop on Personalisation in E-Learning Environments at Individual and Group Level (PING), User Modeling. vol. 2007 (2007)
5. Bull, S., Kay, J.: Student models that invite the learner in: The smili:() open learner modelling framework. International Journal of Artificial Intelligence in Education 17(2), 89–120 (2007)
6. Burke, R.: Hybrid web recommender systems. In: The adaptive web, pp. 377–408. Springer (2007)
7. Carbonell, J.R.: Ai in cai: An artificial-intelligence approach to computer-assisted instruction. Man-Machine Systems, IEEE Transactions on 11(4), 190–202 (1970)
8. Hosseini, R., Brusilovsky, P.: Javaparser: A fine-grain concept indexing tool for java problems. In: The First Workshop on AI-supported Education for Computer Science (AIEDCS 2013). pp. 60–63 (2013)
9. Hsiao, I.H., Sosnovsky, S., Brusilovsky, P.: 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)
10. Hsiao, I.H., Bakalov, F., Brusilovsky, P., König-Ries, B.: Progressor: social navigation support through open social student modeling. New Review of Hypermedia and Multimedia 19(2), 112–131 (2013)
11. Lindstaedt, S.N., Beham, G., Kump, B., Ley, T.: Getting to know your user–unobtrusive user model maintenance within work-integrated learning environments. In: Learning in the synergy of multiple disciplines, pp. 73–87. Springer (2009)
12. Loboda, T.D., Guerra, J., Hosseini, R., Brusilovsky, P.: Mastery grids: An open source social educational progress visualization. In: Open Learning and Teaching in Educational Communities, pp. 235–248. Springer (2014)
13. Mitrovic, A., Martin, B.: Evaluating the effect of open student models on self-assessment. International Journal of Artificial Intelligence in Education 17(2), 121–144 (2007)
14. Vassileva, J., Deters, R.: Dynamic courseware generation on the www. British Journal of Educational Technology 29(1), 5–14 (1998)