NAVIGATION SUPPORT AND SOCIAL VISUALIZATION FOR PERSONALIZED E-LEARNING

by

I-Han Hsiao

B.A. Management Information Systems, National Central University, 2002

M.S. Business Information Systems, Royal Holloway University of London, 2003

Submitted to the Graduate Faculty of

School of Information Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2012

UNIVERSITY OF PITTSBURGH

SCHOOL OF INFORMATION SCIENCES

This dissertation was presented

by

I-Han Hsiao

It was defended on

July 09, 2012

and approved by

Kevin Ashley, Professor, School of Law, Graduate Program in Intelligent Systems

Stephen Hirtle, Professor, School of Information Sciences

Michael Spring, Associate Professor, School of Information Sciences

Vladimir Zadorozhny, Associate Professor, School of Information Sciences

Dissertation Advisor: Brusilovsky, Professor, School of Information Sciences

NAVIGATION SUPPORT AND SOCIAL VISUALIZATION FOR

PERSONALIZED E-LEARNING

I-Han Hsiao, PhD

University of Pittsburgh, 2012

A large number of educational resources is now made available on the Web to support both regular classroom learning and online learning. However, the abundance of available content produced at least two problems: how to help students to find the most appropriate resources and how to engage them into using these resources and benefit from them. Personalized and social learning have been suggested as potential ways to address these problems.

This work attempts to combine the ideas of personalized and social learning by providing navigation support through an open social student modeling visualization. A series of classroom studies exploited the idea of the approach and revealed promising results, which demonstrated the personalized guidance and social visualization combined helped students to find the most relevant resources of parameterized self-assessment questions for Java programming. Thus, this dissertation extend the approach to a larger collection of learning objects for cross content navigation and verify its capability of supporting social visualization for personalized E-Learning.

The study results confirm that working with the non-mandatory system, students enhanced the learning quality in increasing their motivation and engagement. They successfully achieved better learning results. Meanwhile, incorporating a mixed collection of content in the open social student modeling visualizations effectively led the students to work at the right level of questions. Both strong and weak student worked with the appropriate

levels of questions for their readiness accordingly and yielded a consistent performance across all three levels of complexities. Additionally, providing a more realistic content collection on the navigation supported open social student modeling visualizations results in a uniform performance in the group. The classroom study revealed a clear pattern of social guidance, where the stronger students left the traces for weaker ones to follow. The subjective evaluation confirms the design of the interface in terms of the content organization. Students' positive responses also compliment the objective system usage data.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

1.1 THE PROBLEM AND MOTIVATION

A large number of educational resources are now made available on the Web to support both regular classroom learning and online learning. The abundance of available content produces at least two problems: how to help students to find the most appropriate resources and how to encourage them to use and benefit from them. Personalized and social technologies have been explored in several projects aimed at addressing these problems. Personalized learning was suggested as an approach to help every learner find the most relevant and useful content given a learner's current state of knowledge and interest (Kay., 2008). Social learning was explored as a potential solution to a range of problems including how to increase the motivations of students to learn (Barolli, Koyama, Durresi, & De Marco, 2006; Méndez, Lorenzo, Acosta, Torres, & González, 2006; Julita Vassileva, 2008; J. Vassileva & Sun, 2008). In our research group, these approaches have been investigated in two systems, QuizGuide (Brusilovsky, Sosnovsky, & Shcherbinina, 2004) and Knowledge Sea II (Brusilovsky, Chavan, & Farzan, 2004). QuizGuide provides topic-based & prerequisite-based adaptive navigation support for personalized guidance to programming problems. Knowledge Sea II uses social navigation support and map-based visualization to help students navigate weekly readings. Both works have explored personalized learning and social learning respectively and demonstrated their value and effectiveness in E-Learning. Today, similar projects that apply personalized learning or social learning independently are common. Very little literature addresses the crossroads of these two approaches. Therefore, the work presented in this dissertation attempts to integrate personalized learning and social learning to help students to find the most relevant resources within a large collection of educational content.

Within the area of personalized learning, this work is motivated by the success of *personalized guidance* and *open student modeling*. Personalized guidance is known to increase learning rate and quality (Brusilovsky, 2007; Kavcic, 2004), however, most of the research on personalized guidance focuses on an individual student representation and ignores the social aspects of learning. Open student modeling is another popular approach in the area of personalized learning that allows the students to observe and reflect on their progress. Open student modeling is important as a solution to address this issue of staying in control of one's own learning. In particular, visual approaches for open student modeling have been proven to provide students with an easy-to-grasp and holistic view of their progress (Bull, 2004; Mitrovic & Martin, 2007; Zapata-Rivera & Greer, 2000). Most of the open student modeling research focuses on a representation of an individual student ignoring the social aspect of learning.

In the area of social learning, this research is motivated by the success of *social visualization* and *social navigation support*. Several social visualization approaches explored in an e-learning context (Vassileva, 2008) focus mainly on student communication and collaboration rather than on their progress or the adaptive support for the rapid growing educational hypermedia content. While open student modeling is an excellent match to social visualization, there is almost no work on the crossroads of these approaches. Social navigation (Farzan & Brusilovsky, 2008) as well as other work on social learning techniques reviewed below indicated that the ability to see the work and the progress of student peers can help to increase both, the quality of guidance and the student motivation.

The goal of this work is to bringing these four past streams of research together: to integrate personalized learning with social learning by extending a traditional open student modeling interface with social visualizations and using social navigation to provide personalized guidance. The aim is to guide students to the relevant resources in a large collection of educational content and engage them in doing educational activities. Figure 1 depicts the contribution to the subareas in adaptive educational systems. Figure 1 depicts the contribution to the subareas in adaptive educational systems.

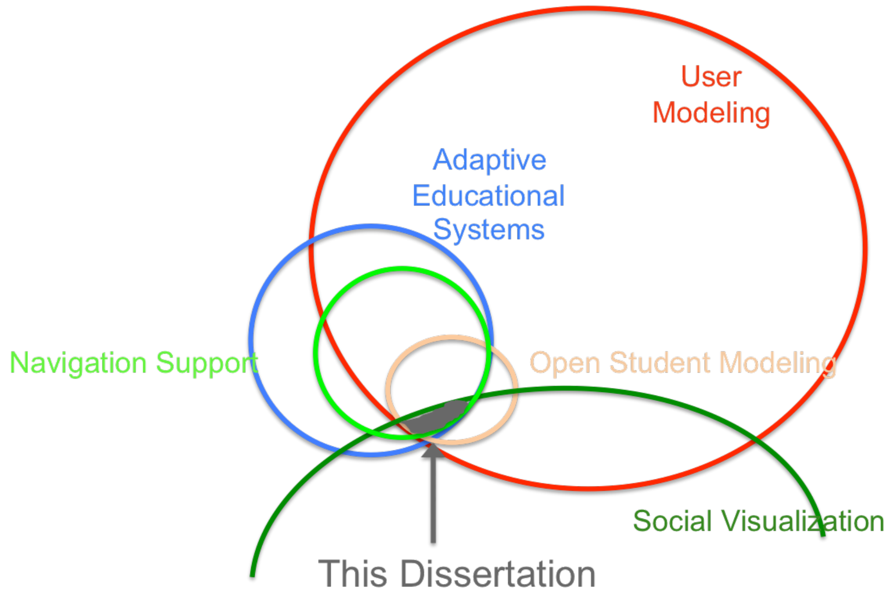


Figure 1. The contribution of this dissertation to the subareas in adaptive education systems

In this dissertation, I designed and carried out a classroom study to explore the approach to provide navigation support with social visualization using the *open social student modeling interface*. The study is designed to examine the feasibility of this approach for large collections of educational resources and the effects in three dimensions: students' motivation, engagement and learning. To understand deeper insight of social mechanism on providing personalized guidance on open social student modeling interface, I investigated the patterns of differences on students' problem solving success by problems' complexities and students' pre-knowledge.

In this work, we set out to answer the following questions.

Question 1: What are the design principles (key features) to implement personalized guidance using open social student modeling visualizations?

Question 2: Will the open social student modeling visualization provide successful personalized guidance within a rich collection of educational resources? I.e. Will this approach guide students to the right content at the right time?

Question 3: Will the open social student modeling visualization approach increase students' motivation & engagement to work with non-mandatory educational content?

Question 4: Will this approach improve students learning?

1.2 OPEN SOCIAL STUDENT MODELING: GUIDING AND ENGAGING STUDENTS THROUGH A SOCIAL VISUALIZATION INTERFACE

1.2.1 An overview of the proposed approach

This work proposes to provide navigation support and social visualization by using open student modeling interfaces in the context of E-Learning. We expect the combined ideas of navigation support and social visualization to elicit personalized and social guidance to enhance students' learning experiences. Based on this general assumption, we implement an open social student modeling interface with navigation support and social visualizations for interacting with a large collection of educational resources. The social visualization interface is a visual presentation of the student model, which allows students to directly access the content and, at the same time, maintain control of their academic performance.

This approach includes four parts: adaptive navigation support, open student modeling, social visualization, social navigation support and the integration of these four elements. Each approach will be illustrated in detail in the following subsections. In addition, Figure 2 provides an overview of the research model for the proposed approaches. With the combinations of navigation support and social visualization interventions, we expect the underlying behavioral, psychological and sociological theories are the mediating factors. The theories and methodologies will be defined and reviewed in detail in Chapter 2. Each technology is summarized in the following section. In Chapter 3, we present the project context, tools, a summary of series of formative studies and the comprehensive methodology with the experimental design. The results are presented in Chapter 4. We finally discuss this work and summarize the limitations and the contributions in Chapter 5.

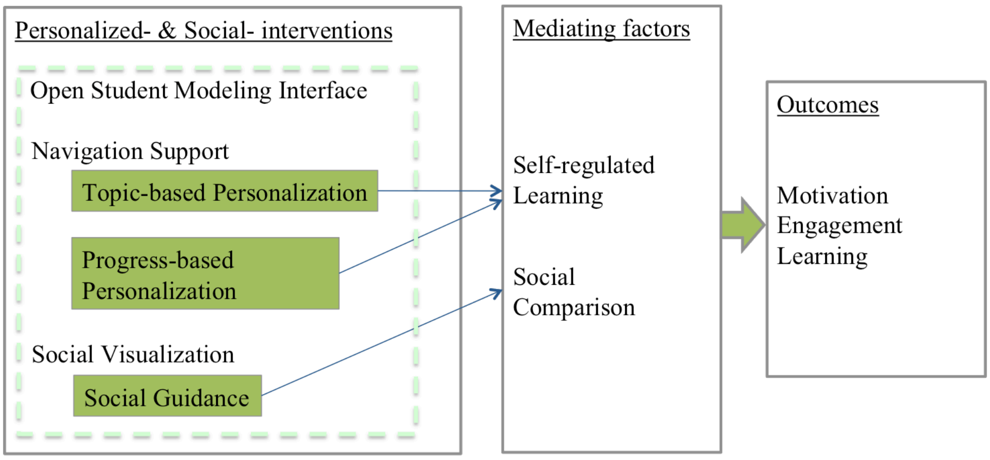


Figure 2. The research model for proposed approaches

1.2.2 The components of the proposed approach

1.2.2.1 Personalized learning approaches: open student modeling and personalized navigation support

Open student modeling is a popular technology in the area of personalized learning that provides the students with a visual interface to observe and reflect on their progress. SQL-tutor (Mitrovic & Martin, 2007) offers a simple example of an open student model (Figure 3). It adaptively displays the state of learner's knowledge using so-called *skill meter*. Such approach has been proven to promote students' awareness, increase motivation etc.

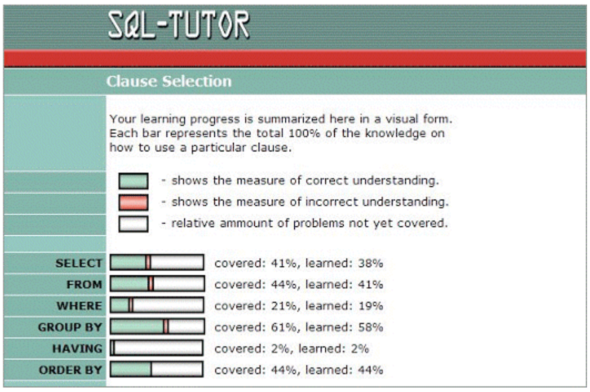


Figure 3. SQL-tutor - A classic example of open student modeling visualization with skills meter

QuizGuide (Figure 4) is one of the few systems that implemented the navigation support on open student modeling interfaces (Brusilovsky, Sosnovsky, et al., 2004; Sosnovsky & Brusilovsky, 2005). It utilizes open student modeling to increase students' motivation and uses navigation support to provide personalized guidance. We evaluated the same approach from QuizGuide in our target context for this work in (Hsiao, Sosnovsky, & Brusilovsky, 2009, 2010). In this dissertation work, we capitalize this approach by using navigation support on open student modeling interface to provide personal guidance.

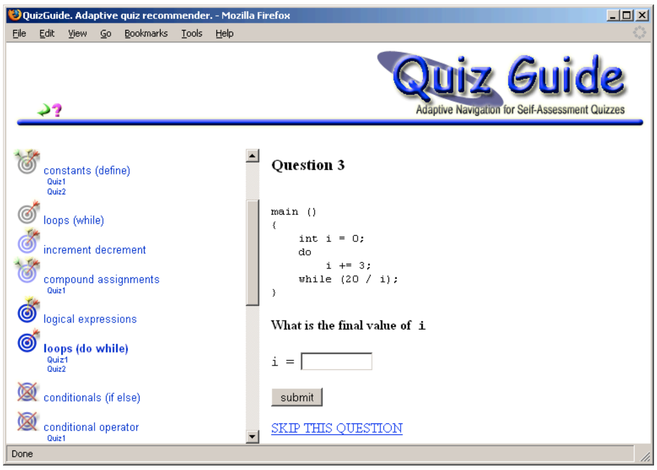


Figure 4. QuizGuide - An example of knowledge-based personalization system

In QuizGuide, a particular modeling technique has been used. Studies of concept-based content modeling have suggested this detailed model yields better accuracy of student knowledge inference and leads to more effective adaptations of the content for each individual student; however, such fine-grained modeling involves a high complexity of indexing and a lot of authoring effort to develop. Given these constraints, in QuizGuide, it adopts a similar, alternative modeling technique called topic-based modeling, which is based on the natural approach for a classroom teacher (or a textbook author) to organize the course in to separate units. It reduces the number of indices by splitting the learning materials into large units, which are the topics. Each topic represents a unique knowledge component. It provides indications of aggregated learning materials[1], instead of traditional indexing. In other words, by associating many learning objects (such as quizzes, questions, examples etc.) to topics, we will obtain a small-number of indices. Thus, we call it coarse-grained topic-based modeling.

[1] The term learning materials will be used interchangeably with learning objects throughout this dissertation.

1.2.2.2 Social learning approaches: social visualization and social navigation support

In the field of E-Learning, social visualization is usually applied to visualize some information about a group of students. Such group visualization enables students to compare and understand their own states. Comtella (Julita Vassileva, 2008) is an example of social visualization system (Figure 5), which captured various aspects of users interactions in a discussion forum and enhanced users motivation and participation. While social navigation support is intended to provide students the social guidance, which has been studied that such technique guides users to relevant information by showing the traces of past users' work (Dieberger, Dourish, Höök, Resnick, & Wexelblat, 2000).

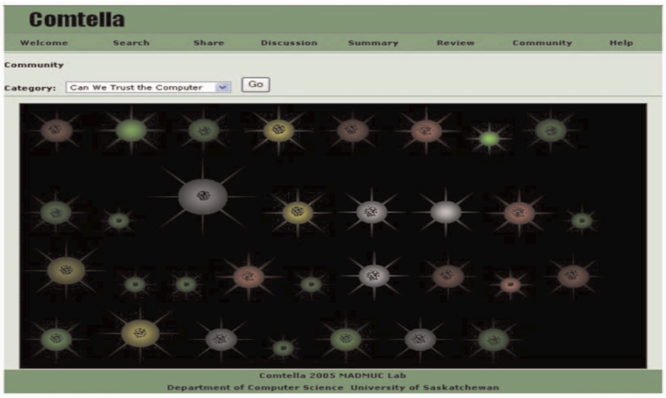


Figure 5. Comtella – A social visualization interface for online communities.

An example of the combined social visualization and social navigation supported system is demonstrated in Figure 6.

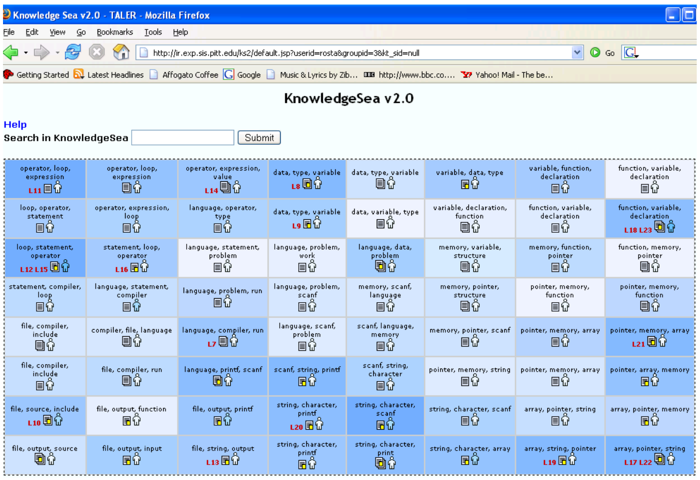


Figure 6. KnowledgeSea II – an example of social navigation based system

Traditionally, social navigation support mainly concentrates on displaying aggregated community performance versus individual performance for each educational resource or topic. It does not provide the outline of the complete student models to the students. In our approach, we focus on presenting an easy-to-grasp and holistic view of all student models as well as the content model. We intend to follow Shneiderman's Visual Information Seeking Mantra (Shneiderman, 1996): "overview first, zoom and filter, then details-on-demand". Students are initially presented a holistic view of their model in contrast to the models of their peers. Each student then has the option to start interacting with the system by zooming in to see the detail of the models or starting working on the content domain. In other words, the interface is no longer just the graphical user interface to access the adaptive content; it is also an interactive visualization which presents all students' models at one time.

The social visualization essentially exposes information to the group (eg. for classroom settings, the group represents the class). In the context of learning, students' performances are revealed openly for comparisons. Based on *Social Comparison Theory*, the social aspects of the system should give students the motivation to evaluate, to improve and to enhance their own performances. To date, among fifty years of social comparison theory literature, there are no quantitative measures for "social comparison" because most evidence supporting this theory is drawn from qualitative studies (i.e., interviews, questionnaires and observation). Therefore, in this work, we develop a set of quantitative measures for explicit and implicit social comparisons in the target context. The details of our implementation of such approach will be discussed in Chapter 3.

1.2.2.3 Bridging it together: Integration of Navigation support and Open Social Student Modeling using social visualization interface

The key technology innovation of this research is the integration of adaptive navigation support with social visualizations by using open student modeling interface, I call it *Open Social Student Modeling*. Traditionally, students interact with the adaptive education systems and receive adaptation effects from the user interfaces (Figure 7 left). Open student modeling provides students the opportunities to interact with their own models through the graphical interfaces (Figure 7 right), which is intended to help raise the students' awareness of their learning performances. Moreover, the combined approaches in open student modeling interfaces embed various sources of motivation, such as providing focused and relevant content to the students, allowing students to feel in control. Additionally, there is less work addressing the application of this modeling technique to support the working behavior (Verginis, Gouli, Gogoulou, & Grigoriadou, 2011) in the open student modeling literature. Therefore, in this work, we attempt to use navigation support and social visualizations to provide personalized and social guidance to students. Meanwhile, capitalizing the merits of open student modeling, social visualization and social navigation support to increase students' motivation in working on adaptive educational systems.

I hypothesize that:

- **Providing navigation support on open student modeling interface will guide students to the right content at the right time.**
- **Students will be motivated to do more work with social visualization.**

- **Students will improve their learning by using the integrated approach of open social student modeling visualization.**

In our approach, we try to capture the representations with personalized components and social visualizations to display the student models from succinct to detail and individual to social. Figure 8 is the implementation of our approaches of the combined personalized guidance and social visualizations by using open student modeling interface, which will be discussed more in detail in Chapter 3.

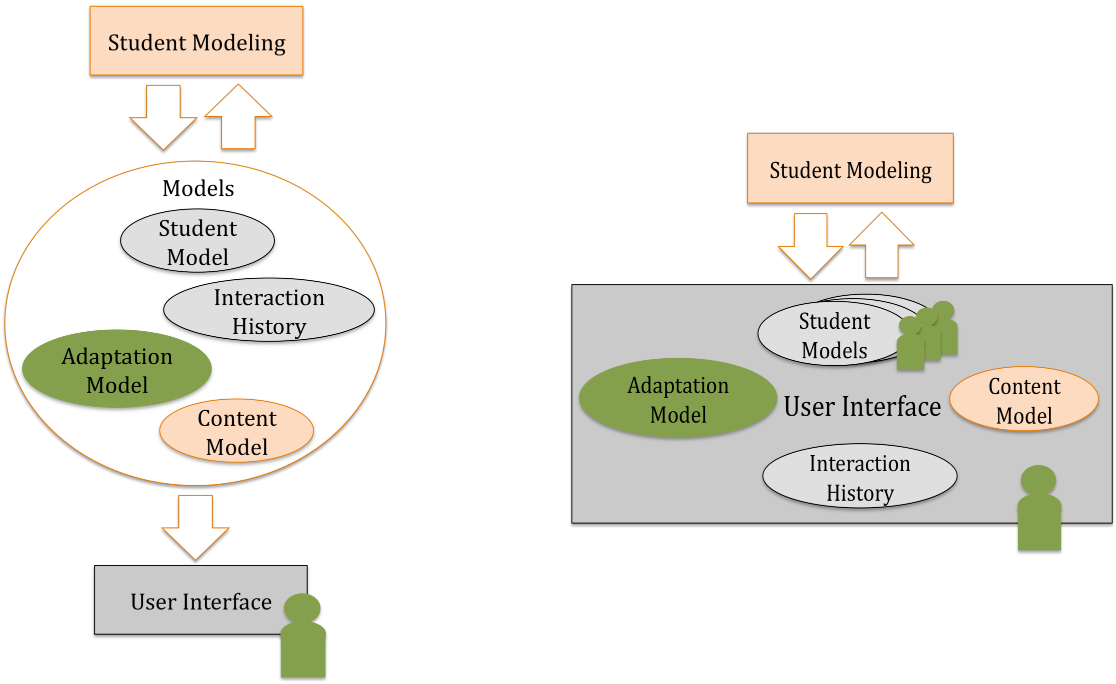


Figure 7. Traditional approach (left) vs. Integration of students' models into interface (right)

Figure 8. Progressor - combined approaches of open social student modeling visualization.

In order to answer the research questions of this work, the proposed approaches will be evaluated based on the research model. To address outcomes for each research question, we formulated the hypotheses and introduced the measurements for assessing them in detail in Chapter 3&4. In the pre-studies, we complied the proposed approaches and presented an investigation of the feasibilities in engineering the integration capability of adaptive navigation support with open social student modeling visualizations. Finally, we design a comprehensive study for researching a cross content navigation by using the proposed approaches. The major goal of this study is to demonstrate that the proposed approach is capable of supporting cross content navigation by open social student modeling visualizations in personalized E-Learning.

1.3 DISSERTATION OUTLINE

The rest of this dissertation is organized as follows: Chapter 2 provides a general overview of related work including adaptive educational systems and social technologies for E-learning. Chapter 3 presents the methodology with the context of the content collections of this dissertation work, a summary of the proposed paradigm with a series of formative studies and the experimental design of this dissertation study. We then discuss the results in Chapter 4. In Chapter 5, we summarize this dissertation work. Conclusions, discussions, contribution, potential future work and the limitations will be presented.

2.0 BACKGROUND AND RELATED WORK

This chapter covers the background literature and related work. First, I provide an overview of personalized E-Learning. I specifically focus on adaptive navigation support and open student modeling. Next, I review the social technologies for e-Learning, including social visualizations and social navigation support, followed by the corresponding supporting theories in the context of e-Learning. Lastly, I review exercised- and example-based learning approaches in e-Learning for the target context of this dissertation.

2.1 PERSONALIZED E-LEARNING

Personalized technologies have demonstrated some capabilities and successes in the field of E-Learning. A traditional knowledge-based adaptive educational system relies on several sources for the student's knowledge, the content being adapted and the ways it can be better tailored for the student. The content includes transaction logs, domain elements, and most importantly, associations between content items (problems, examples, etc.) and domain concepts. All of the sources combined allow the system to adaptively present the most suitable item to the students. However, maintaining the associations between content items and the domain concepts requires tremendous effort (i.e. domain experts must manually index the associated concepts with each item during the authoring phase). Hence, with the rapid growth in educational resources, new challenges for guiding students to the most appropriate resources among a greater extent amount of educational resources have suggested.

Traditional non-adaptive educational systems provide the same diagnosis in response to the same solution to a problem regardless of the student's past experience or preferences within the system. In other words, the same material can be unclear to novices and boring to advanced learners. Additionally, without navigational support to guide students through the hyperspaces, it is easy to get lost even in the very beginning or within a reasonably small hyperspace (Brusilovsky, 1996). Adaptive strategies are an alternative to the traditional "one-size-fits-all" approach. They build a model of the goals, preference, knowledge etc. of each individual student and use this model to interact with the student and adapt to his/her needs.

2.1.1 Adaptive Navigation Support

The most distinguishing benefits of adaptive educational systems can be found in two main technologies derived from Adaptive Hypermedia Systems, which include content level adaptation (also known as: Adaptive Presentation) and link level adaptation (also known as: Adaptive Navigation Support) (Brusilovsky, 2001a). Adaptive presentation technology aims to adapt the content according to a student model, based on student knowledge, goals, interests, background and individual traits. From the student model, personalized views can be provided to the students to protect them from getting lost in the complex hyperspaces. Successful examples include AHA! (Bra & Calvi, 1998), CooTutor (Wang, Li, & Chang, 2004), ActiveMath (Melis, Andres, Budenbender, Frischauf, Andrès, et al., 2001) etc. In this dissertation, I focus on the adaptive navigation support technique, which is aimed to help students orient themselves in the hyperspaces by changing the appearance of the links (Brusilovsky, 2003).

Adaptive navigation support is a group of techniques that aim to help individual users locate relevant information in the context of hypertext and hypermedia (Brusilovsky, 2001b). By adaptively altering the appearance of links on every browsed page, such methods as direct guidance, adaptive ordering, adaptive link hiding and removal, and adaptive link annotation support browsing-based personalized access to information. E-Learning, with its constant need to adapt to the level of student knowledge, is one of the most active application areas of adaptive navigation support. The educational power of adaptive navigation support has been recognized, and it has been turned into a number of interactive systems. For example, prerequisite-based adaptive navigation support is used in AHA! (De Bra & Calvi, 1998), ELM-ART (Weber & Brusilovsky, 2001) and KBS-Hyperbook (Henze & Nejdl, 2001), while progress based annotation is used in INSPIRE (Grigoriadou, Papanikolaou, Kornilakis, & Magoulas, 2001), InterBook (Brusilovsky, Eklund, & Schwarz, 1998) and NavEx (Brusilovsky, et al., 2009). QuizGuide (Brusilovsky, Sosnovsky, et al., 2004) uses both adaptive annotation approaches in parallel. There are various kinds of adaptive annotation techniques for adaptive navigation support, such as zone-based annotations in ISIS-Tutor (Brusilovsky & Pesin, 1994), which divides all resources into three groups or zones: 1) sufficiently known, 2) new and ready to be explored, and 3) not ready to be explored.

In the E-Learning context, these techniques demonstrated their ability to support faster achievement of the users' goals, reduce navigational overhead, and increase user satisfaction (Brusilovsky & Eklund, 1998; Davidovic, Warren, & Trichina, 2003; Kavcic, 2004; Olston & Chi, 2003). However, the majority of systems applying these techniques in E-Learning, as well as the majority of evaluation studies, focused only on guiding students to the right piece of text-based content – such as introduction of the concept or the explanation. In this context, neither the complexity of the content, nor the student learning success can be measured reliably. In our own experiences, QuizGuide & JavaGuide (Brusilovsky, Sosnovsky, et al., 2004; Hsiao, et al., 2010), we present two of the very few examples of applying adaptive navigation support to guide students to the most appropriate questions and problems. The results showed that adaptive navigation support helped to promote students' participation and significantly increased their success rate with online self-assessment quizzes. Students were more likely to answer a question correctly with adaptive navigation support than without it. We also found that adaptive navigation support effectively guided both strong and weak students to the appropriate quizzes and contributed to students learning. In addition, adaptive navigation support provides a stable effect promoting attempts and success rate across different complexity levels.

2.1.2 Open student modeling

A student model is a representation of the student's knowledge, difficulties, and misconceptions. Adaptive educational systems rely on student models to provide the adaptation effects to adapt to students' needs. Open student models are learner models that drive the personalization of adaptive educational systems as well as maintain the users' or his/her peers' information (Bull & Britland, 2007; Kay, 1997).

There are two main streams of work on open student models. One stream focuses on visualizing the model to support students' self-reflection and planning; the other encourages students to participate in the modeling process, such as engaging students through negotiation or collaboration on construction of the model (Mitrovic & Martin, 2007). Representations of the student model vary from displaying high-level summaries (such as skill meters) to complex concept maps or Bayesian networks. A range of benefits has been reported for opening the student models to the learners, such as increasing the learner's awareness of his or her developing knowledge, difficulties encountered in the learning process, and students' engagement, motivation, and knowledge reflection (Bull, 2004; Mitrovic & Martin, 2007;

Zapata-Rivera & Greer, 2000). Dimitrova et al. (Dimitrova, Self, & Brna, 2001) explore interactive open learner modeling by engaging learners to negotiate with the system during the modeling process. Chen et al. (Chen, Chou, Deng, & Chan, 2007) investigated active open learner models in order to motivate learners to improve their academic performance. Both individual and group open learner models were studied and demonstrated the increase of reflection and helpful interactions among teammates. Bull & Kay (Bull & Kay, 2007) described a framework to apply open user models in adaptive learning environments and provided many in-depth examples. Studies also show that students have a range of preferences for presentations on viewing their own knowledge in the open student modeling systems. Students highly value the options of having multiple views and being able to select the one they are most comfortable with. Such results are promising for potentially increasing the quality of students' reflection on their own knowledge (Mabbott & Bull, 2004). In our own work on QuizGuide system (Brusilovsky, Sosnovsky, et al., 2004) we embedded open learning models into adaptive link annotation and demonstrated that this arrangement can remarkably increase student motivation to work with non-mandatory educational content.

2.2 SOCIAL TECHNOLOGIES FOR E-LEARNING

Social technologies have been widely explored and are popular personalized E-Learning technologies. They rely on users' feedback to guide subsequent users. For instance, implicit user feedback is usually obtained through interactions with the systems, generating trails for other users to follow. Social technologies not only accelerate content creation but also provide opportunities to go beyond traditional methods of encouraging active participation in educational activities (Sigala, 2007). However, in the context of E-Learning, they create new problems for personalization. The massive amount of new content provided results in a massive amount of content item and domain concepts associations remaining unmanaged. Social technologies are primarily concerned about capturing general representations of the community and thus neglect the overlap with the domain model. With the increasing amount of educational resources, student performance can be more precisely established, but at the cost of increased complexity in modeling the domain and the content for better student knowledge indication. Therefore, the challenge for this work is to extend the benefits from personalized learning and social learning and bring these two streams of technologies together.

According to Vygotsky's Social Development Theory (Vygotsky, 1978), social interaction will affect the process of cognitive development. The Zone of Proximal Development (ZPD) is the distance between a student's ability to perform a task under adult guidance and/or with peer collaboration and the student's ability to solve the problem independently. This is where the learning occurs. Research on social learning has confirmed that it enhances learning outcomes across a wide spectrum including better performance, better motivation, higher test scores and level of achievement, development of high level thinking skills, higher student satisfaction, self-esteem, attitude, and retention in academic programs (Cecez-Kecmanovic & Webb, 2000; Johnson, Johnson, & Smith, 1998; Koedinger & Corbett, 2006).

Nowadays, a range of approaches is used to support social learning. This subsection focuses on two of these approaches that are directly contributing to this dissertation - social navigation support and social visualization.

2.2.1 Social visualization

The term Information Visualization was defined in 1999 as “the use of computer-supported interactive visual representations of abstract data to amplify cognition” (Card, Mackinlay, & Shneiderman, 1999), a definition commonly accepted in the Computer Graphics Information Visualization community (Ostergren, Hemsley, Belarde-Lewis, & Walker, 2011). Other definitions, like the one proposed by Gershon and Page, extend the scope of information visualization beyond computer-supported methods to “a process that transforms data, information and knowledge into a form that relies on the human visual system to perceive its embedded information” (Gershon & Page, 2001). While the latter definition is more extensive, both definitions are wide enough to portray the practice of information visualization as a technique that can be applied to support several fields, as has been shown in geospatial analysis (Maceachren, Wachowicz, Edsall, Haug, & Masters, 1999), software engineering (Ellis, Wahid, Danis, & Kellogg, 2007), and social networks (Heer & Boyd, 2005), among many others.

The existing approaches of visualizing interaction of users inside a system go beyond representation of “footprints”. For instance, the representation of users and their implicit or explicit interactions as a network is a popular approach and several studies have surveyed different visualization techniques (Freeman, 2000; Skold, 2008). While Skold focused on presenting several network layouts and software tools to represent networks, Freeman surveyed social network visualization to show examples of different ways to use position, color, size and shape to encode network information. One frequently cited system that makes use of a node-link network layout is Viszter, a “visualization system for playful end-user exploration and navigation of large-scale online social networks” (Heer & Boyd, 2005). In Viszter, the users are represented as nodes and the edges between the nodes represent friendship links inside the Friendster system. Viszter not only allows examination of users and their friends, but also of community structures. Another way to represent a social network or the social interaction of users in a system is by Hyperbolic Trees (Lamping, Rao, & Pirolli, 1995). They are intended to visualize large hierarchy-based networks and they have been used in several domains, such as visualizing interpersonal relationships in social networks from a user-centric point of view (Ho, Chang, Chen, & Yang, 2010), or visualizing Tweets in Technology Enhanced Education (TEL) for trend detection (Kraker, Wagner, Jeanquartier, & Lindstaedt, 2011). Other studies have used matrix representation to simplify the visualization of the typical node-link structure under the formation of hairballs in large social networks, such as MatrixExplorer (Henry & Fekete, 2006). The aforementioned methods, excepting the one on Tweets in TEL, represent static networks. On the other side, dynamic networks, defined as networks that change over time, rely on extensions of the node-link visualization approaches. A good example of four methods designed to facilitate the perception of networks over time is presented in (Windhager, Zenk, & Federico, 2011) including: animation, layer comparison, layer merging and 2.5 layout. (Viegas & Donath, 2004) discussed and presented evidence from a user study that relied solely on the graph-based model for social network, which has limitations in that you may end up with rather illegible visualizations. They suggest the use of zooming and multiple viewing modes to better capture a social visualization.

In the field of E-Learning, social visualization is usually applied to visualize some information about a group of students. Such group visualization enables students to compare and understand their own states. Group models have been used to support the collaboration between learners within the same group, and to foster competition in groups of learners (Vassileva & Sun, 2007). Vassileva and Sun(Vassileva & Sun, 2007) investigated community visualization in online communities. They summarized that social visualization allows peer-recognition and provides students with the opportunity to build trust in others and in the group. Bull & Britland (Bull & Britland, 2007) used OLMlets to research the problem of facilitating group collaboration and competition. The results showed that providing the option of displaying the models to their peers increases the discussion among students and encourages them to start working sooner. CourseVis (Mazza & Dimitrova, 2007) is one of the few systems providing graphical visualization to teachers and learners for multiple groups of users. It helps instructors to identify problems early on, and to prevent some of the common problems in distance learning.

2.2.2 Social navigation

Social navigation is one of the most popular personalization technologies that relies on users' activities for modeling. It captures human's natural behavior by following the crowd. Social navigation also represents a set of methods for organizing users' explicit and implicit feedback for supporting information navigation (Dieberger, et al., 2000). This technique attempts to support a known social phenomenon where people tend to follow the "footprints" of other people (Brusilovsky, Chavan, et al., 2004; Dieberger, 1997; Wexelblat & Maes, 1999).

The educational value has been confirmed in several studies (Brusilovsky, et al., 2009; Farzan & Brusilovsky, 2008; Kurhila, Miettinen, Nokelainen, & Tirri, 2006). One of the big advantages of the social navigation approach is that it requires no prior users' knowledge about the content. Therefore, it doesn't require the creation of any additional models, which greatly facilitates the content indexing for a large volume of content with knowledge-based personalization work alone. Knowledge Sea II (Brusilovsky, Chavan, et al., 2004) is an example of a social navigation system for e-Learning. It provides students with navigational support for the reading material from several online sources, such as e-books, tutorials, etc.

Tutorial pages are automatically clustered into the cells of a dynamic map according to their term vectors. Knowledge Sea II paints the map cells with different degrees of blue color based on the amount of traffic students have generated for the resources assigned to the cell. Knowledge Sea II also represents the individual traffic of the student by a little human icon colored with the same palette. It allows students to compare their own progress with the progress of the rest of the class and focus on the resources where they lag behind. In addition to traffic information, Knowledge Sea II allows students to annotate the tutorial pages and uses this information as an additional social navigation cue. There are some other systems that use social navigation technique, such as Dogear (Millen, Feinberg, & Kerr, 2006), KALAS (Svensson, Höök, Laaksolahti, & Waern, 2001) etc.

2.3 SUPPORTING THEORIES

2.3.1 Self-regulated learning

Self-regulated learning includes students' meta-cognitive strategies for planning, monitoring, and modifying their cognition (Pintrich & de Groot, 1990; Zimmerman, 1990). Self-regulated students select and use self-regulated learning strategies to achieve desired academic outcomes on the basis of feedback about learning effectiveness and skill. It involves self-monitoring to interpret feedback from their academic learning (Zimmerman, 1990). In (Azevedo, Guthrie, & Seibert, 2004), authors investigated how self-regulated learning helped students acquire conceptual understanding. The results showed that students who gained higher conceptual understandings (high jumpers) tended to be good at regulating their learning by using effective strategies, planning their learning by creating sub-goals and activating prior knowledge, monitoring their emerging understanding, and planning their time and effort. On the other hand, students who gained lower conceptual understandings (low jumpers) tended to handle task difficulties and demands by engaging mainly in help-seeking behavior, and did not spend much time monitoring their learning. However, knowledge of cognitive and meta-cognitive strategies is usually not enough to promote student achievement; students also must be motivated to use the strategies, and regulate their cognition and effort (Pintrich, 1999). Studies have shown that assisting students to recognize their task performance builds self-efficacy and positive beliefs, which in turn helps sustain motivation for continued strategy use (Butler, 1998). In this dissertation, we used personalized guidance and social visualizations to increase students' awareness and motivation for promoting their self-regulated learning.

2.3.2 Social comparison

According to social comparison theory (Festinger, 1954), people tend to compare their achievements and performance with people who they think are similar to them in some way. There are three motives that drive one to compare him/herself to others, namely, self-evaluation, self-enhancement, and self-improvement. The occurrence of these three motives depends on the comparison targets, namely, lateral comparison, downward comparison and upward comparison. Earlier social comparison studies (Veroff, 1969) demonstrated that students were inclined to select challenging tasks among easy, challenging, and hard tasks by being exposed to social comparison conditions. Feldman and Ruble (1977) argued that age differences resulted in different competencies and skills in terms of social comparison. As young children grow older, they become more assured of their general competence through social comparing skills (Feldman & Ruble, 1977). Later studies showed that inducing social comparison with a graphical feedback tool decreases social loafing and increases productivity(Shepherd, Briggs, Reinig, Yen, & Jay F. Nunamaker, 1995). A synthesis review of years social comparison studies' summarized that upward comparisons in the classroom often lead to better performances (Dijkstra, Kuyper, Werf, Buunk, & Zee, 2008). Among fifty years of social comparison theory literature, most of the work was done with qualitative studies by interviews, questionnaires and observation. In this dissertation, the open social student modeling interface allows students to manipulate the content and the students' models among their peers, which enables perform explicit social comparisons. The results of this approach are expected to contribute to quantitative analysis for applying social comparison theory.

2.4 EDUCATIONAL TOOLS FOR PROGRAMMING LANGUAGE LEARNING

2.4.1 Exercise-based learning

Online quizzes have become a popular tool for the self-assessment of student knowledge in the context of modern education (Brusilovsky & MIller, 2001). Several studies have demonstrated the effectiveness of self-assessment quizzes for medical training (Henson, Dews, Lotto, Tetzlaff, & Dannefer, 2005), physics education (Titus, Martin, & Beichner, 1998), and the learning of programming language (Brusilovsky & Sosnovsky, 2005b; Williams, Bialac, & Liu, 2006). Traditionally, the quizzes have consisted of a range of static questions that have been composed by the teacher. However, it is time-consuming for teachers to author questions and to maintain the large pool of questions supporting the necessary level and breadth of knowledge assessment needed for their classes. These high authoring costs result in an insufficient number of questions, which creates problems when using online quizzes for assessment. For example, students can more easily memorize a small number of answers and thus answer by rote memory instead of understanding or cheat on the assessment. As a result, lack of questions may hinder students from assessing their true knowledge.

One of the most effective solutions is to generate dynamic questions. Creating parameterized questions and exercises is one way to implement this solution. Essentially, parameterized questions are templates created by the author. When presented, the templates will be instantiated with randomly generated parameters. As a result, every question template is able to produce many different questions. A number of systems have explored the use of parameterized questions in a range of topics and demonstrated the benefits of this approach, including CAPA (Kashy, Thoennessen, Tsai, Davis, & Wolfe, 1997), WebAssign (Titus, et al., 1998), EEAP282 (Merat & Dukki, 1997), Mallard (Graham & Trick, 1997), QuizPACK (Brusilovsky & Sosnovsky, 2005b) and QuizJET (Hsiao, Brusilovsky, & Sosnovsky, 2008).

2.4.2 Learning from worked examples

Nowadays, multiple code examples, ranging from small code snippets to complete programs, can be found in any programming textbook, and are also frequently provided on an attached CD or a web site supporting the textbook. The educational power of examples has been recognized, and from this has come a number of interactive systems that have attempted to increase the value of examples as tools for learning (Brandt, Dontcheva, Weskamp, & Klemmer, 2010; Brna, 1998; Brusilovsky, 2001c; Brusilovsky & Yudelson, 2008; Burow & Weber, 1996; Davidovic, et al., 2003; Gómez Albarrán, 2005; Linn, 1992; Pirolli & Anderson, 1985; Weber, 1996). Using worked examples has been exhaustively studied and perceived as an effective instructional strategy to teach complex problem-solving skills (van Merriënboer & Sweller, 2005). According to *Cognitive Load Theory*, human beings have limited capacity in their working memory (Sweller, 1988). When novices are learning the necessary schemas to solve new types of problems, they are imposing an extraneous cognitive load, which denies limited working memory resources to cognition germane to learning (Sweller, van Merrienboer, & Paas, 1998). Studies have compared learning only by problem solving to only by studying worked examples and found that pure worked example study was better for novices (Atkinson, Derry, Renkl, & Wortham, 2000; Renkl, 1997). Meanwhile, a number of lab experiments and studies have also shown that students learn more efficiently from problem solving activities with worked examples mixed in (Pashler, et al., 2007).

Our original approach of using examples to support online learning in the context of programming courses was suggested by the WebEx system (Web Examples), developed by our research group several years ago (Brusilovsky, 2001c). It has been exhaustively studied and used in real classrooms for interactive access to examples enhanced with line-by-line comments. Such technology has been disseminated across several programming classes in C, Java, and SQL that were taught in several institutions, ranging from large research universities to community colleges (Brusilovsky, Grant, Hsiao, Moore, & Sosnovsky, 2007).

2.4.3 Support for teaching programming

In programming language learning, the language semantics is considered one of the cornerstones of programming expertise. These kinds of semantics questions are critical for any programming course and are included in many any assessments and exams. While semantics-oriented questions do not directly assess student pragmatics knowledge (i.e., the ability to write programs that can solve a specific problem), computer science educators argue that semantics provides a foundation for pragmatics. Semantics knowledge and program tracing skills have been a focus of several studies (Lister, 2004 #153) and the target of many tools for teaching programming (Kumar, 2009 #154) (Hristova, 2003 #155).

3.0 PROJECT CONTEXT

The objective of this dissertation is to validate the framework (navigation support and social visualization by using open student modeling interfaces) for scalable educational content collections. This kind of study requires a considerable preparation. Sufficient volume of learning content should be developed, a set of preliminary studies to determine the critical features of the target framework should be performed, and finally, on the basis of these studies, the framework itself should be developed. This chapter covers all of these issues and provides a complete picture of the project context and the experimental design of the study.

3.1 LEARNING CONTENT

In programming language learning, it is essential to teach students the program syntax, pragmatics, and the semantics. For our project context, we developed two collections: self-assessment questions and annotated examples. Altogether, these collections provide a reasonably large and diverse set of educational content to support a semester-long study of open social student modeling and navigation support. Each collection covers a full range of topics for an introductory Java class, from basics to advanced topics such as objects, classes, polymorphism, inheritance, and exceptions. For each collection of content we had to prepare both an authoring tool and a delivery technology to be presented in the adaptive educational system. The tools and technologies for each collection are listed in the Table 1. This section briefly examines most of the essential aspects of the prepared content. More details on both the authoring and delivery sides of self-assessment questions and annotated examples is provided in APPENDIX A~E.

Table 1. Content collections with associated authoring, delivery and presentation systems.

	Authoring	Delivery	Presentation
Self-assessment quizzes	QuizJET	QuizJET	JavaGuide
Annotated examples	AnnotEx	WebEx	NavEx

3.1.1 Self-assessment questions on the semantics of Java language

A collection of self-assessment questions developed for our study focuses on assessing student knowledge of Java semantics. Each question asks students to mentally execute a fragment of Java program code and predict the outcome of this execution. For instance, “What will be the final value of an indicated variable?” or “What will be printed by the program to the standard output?” Since language semantics is considered one of the cornerstones of programming expertise, these kinds of semantics questions are critical for any programming course and are included in many assessments and exams. While semantics-oriented questions do not directly assess student pragmatics knowledge (i.e., the ability to write programs that can solve a specific problem), computer science educators argue that semantics provides a foundation for pragmatics. Semantics knowledge and program tracing skills have been a focus of several studies (Lister, 2004 #153) and the target for many tools for teaching programming (Kumar, 2009 #154). The semantics questions developed for our studies are *parameterized*. I.e., the question, one (or several) numeric value in the text of the program is randomly parameterized when the question is delivered to a student. As a result, students essentially access the same question multiple times with different values for the parameter and different correct answers. The parameterized mechanism prevents students from cheating by simply memorizing the answers. Meanwhile, repeatedly examining and evaluating the program code provides opportunities for students to learn the program syntax and pragmatics. In addition, the quizzes cover different complexities, which are defined by the number of concepts involved in the question. The varieties of the questions' complexities allow the personalized guidance technology to guide students to the proper level of questions and avoid them selecting problems that are either too simple or too complicated problems and ending either bored or discouraged. An example of a QuizJET question is presented in Figure 9 and APPENDIX A~C.

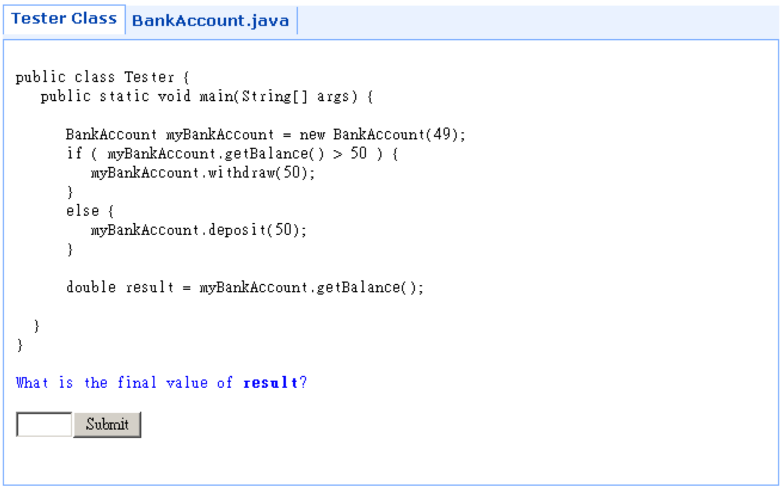


Figure 9. An example of QuizJET question: classes are organized by tab pages. One or more of the parameters in the program codes will be randomly generated when the user attempts the question.

3.1.2 Interactive annotated Java program examples

In another pool of content, annotated examples, students are able to read the line-by-line program code explanations by clicking on the code lines. The simple interaction helps in programming language learning in three ways. First, the annotations of each program line explain the meaning of the code. The explanations include the programming language construct to programming skills, which is meant to bridge the gap between program syntax and semantics. Second, it provides a direct and focused way to study the program code and explanations, where the traditional textbooks usually spread the explanations over the codes by using some special fonts or colors in scattered places in the text. Third, reading from an adaptive interactive system creates more proactive activities than reading large text from a traditional textbook. It provides a rich environment for students to build their understandings in this context. Figure 10 presents an example of the interactive annotated program example. The details of the program codes and annotations authoring, delivery, and presentation is described in APPENDIX D~F.

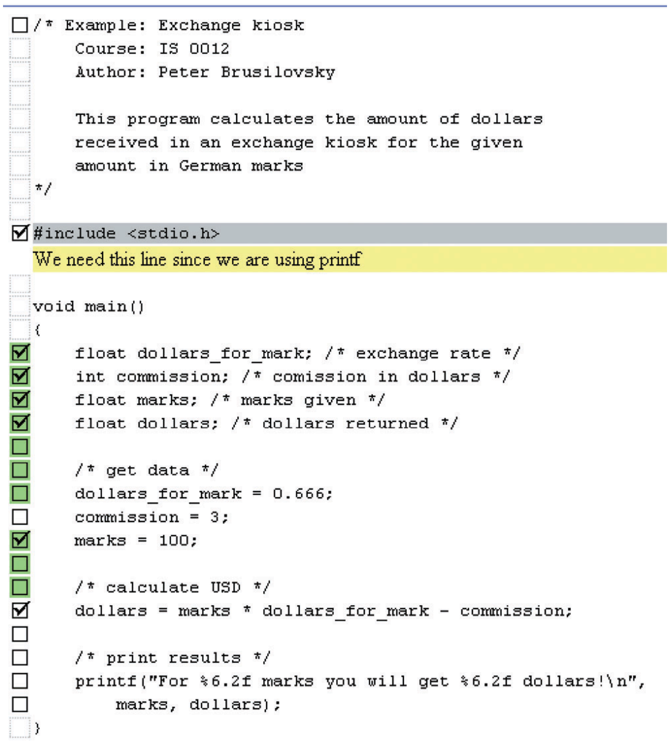


Figure 10. An example of the interactive annotated program example.

3.2 PRE-STUDIES: EXTENDING THE SCOPE OF VISUALIZING STUDENT MODELS WITH ADAPTIVE NAVIGATION SUPPORT

We have conducted three formative studies to explore the viability of the approach explored in this dissertation: providing personalized guidance in open social student modeling visualizations. Among three pre-studies, we investigated the approach in the context of single educational content collection, parameterized self-assessment questions. The three pre-studies are summarized with the lessons learned contributing to the dissertation's final study design. Figure 11 presents a conceptual diagram on the progression of our approach in the three pre-studies.

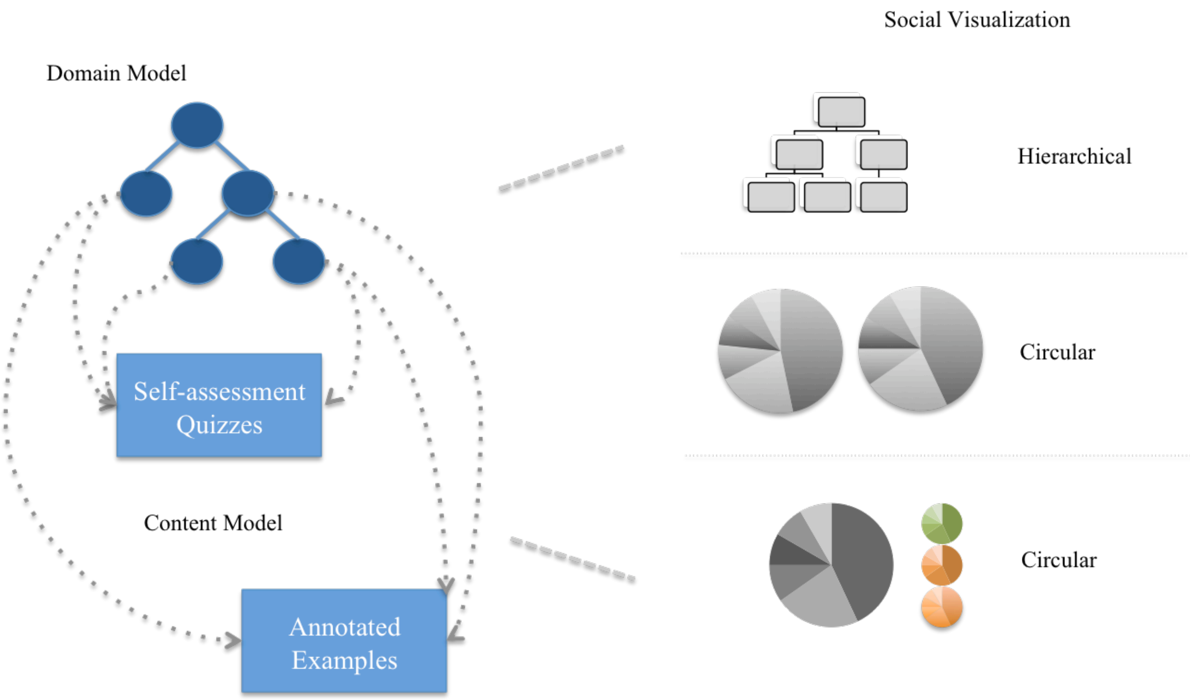


Figure 11. Pre-studies of the approach progression

3.2.1 QuizMAP: adaptive navigation support of parameterized questions with TreeMap

In the first study, we explored a richer integration of open student modeling and adaptive social navigation support. We enhanced the original topic-based navigation support with social navigation (Dieberger, et al., 2000) by using an expressive tile-based TreeMap visualization, called QuizMap, to present many individual tiles on a single map. Students could have their individual detailed performance view as well as holistic view of the group's performance. We learned that students liked the open student modeling implementation for two reasons. They liked being able to see themselves in contrast to the class and they liked being able to interact with the content directly. Open social student modeling visualizations allowed students to follow the social navigation tendency and explore what their peers had been working on. However, as the users increased or the activities increases, some tiles grew bigger and some shrank smaller. The map became dense and complicated. It ended up providing no clear guidance. A snapshot of QuizMap is presented in Figure 12. The details of QuizMap's design and evaluation are summarized in APPENDIX F.

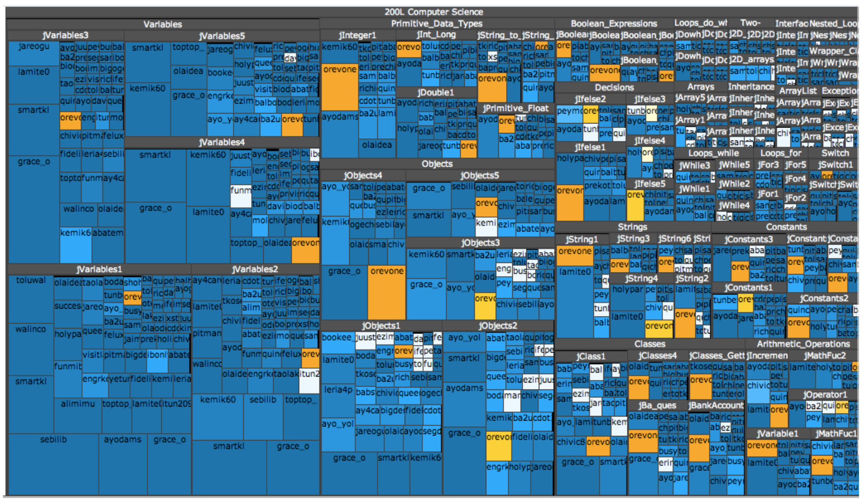


Figure 12. QuizMAP interface

3.2.2 Parallel IntrospectiveViews: visualizing student models through open social student modeling interface

In the second study, our objective was to stress providing more guidance in the open social student modeling interface. We capitalized on our past success in QuizGuide - topic-based and prerequisite-based guidance for personalized e-learning (Brusilovsky, Sosnovsky, et al., 2004). We investigated the prospects of open student modeling with topic- & progress-based personalized guidance with an extension of IntrospectiveViews (Bakalov, et al., 2010) interface. We called it Parallel IntrospectiveViews. Student followed the guidance and progressed throughout the course. They were attracted to the system and interacted with the content remarkably. We succeeded in providing personalized guidance to open social modeling interface. In addition, we discovered that the students were motivated to interact with the content through the comparative interfaces. Therefore, it inspired us to increase the peer awareness in the next study. Figure 13 presents a snapshot of the Parallel IntrospectiveViews interace. The details of the design of the system and its evaluation are summarized in APPENDIX G.

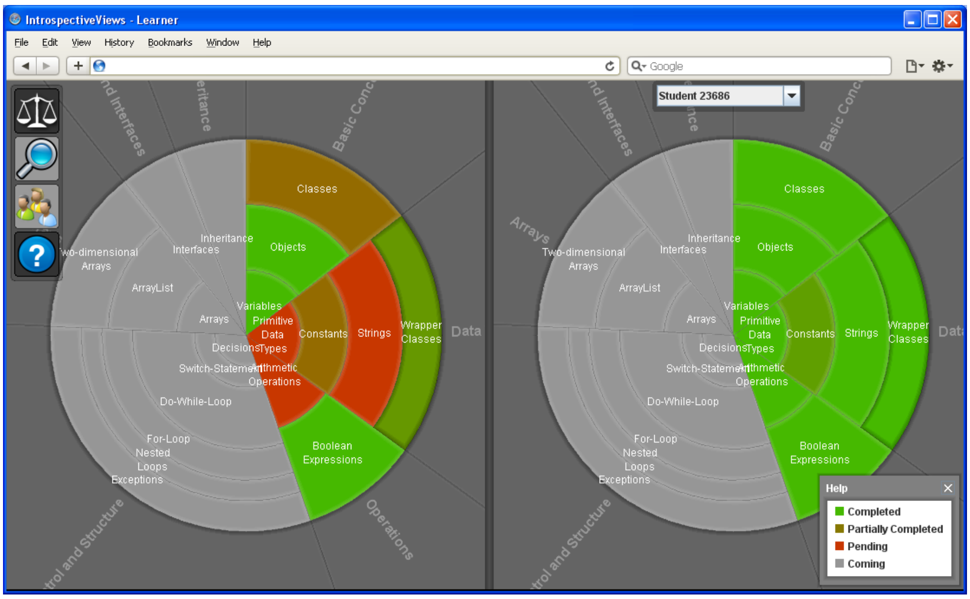


Figure 13. Parallel IntrospectiveViews interface

3.2.3 Progressor: personalized access to programming problems through open social student modeling interface

In the third study, we aimed to finalize the combined approaches of providing personalized guidance in social visualizations by using an open student modeling interface. We implemented Progressor. Progressor was designed to combine all the merits of the previous two studies. It featured direct access to the learning content, topic- & progress-based personalized guidance, open access to the peers' models and the comparative interfaces. The color schemes were improved and aligned to the percentage of the students' progress. In addition, to increase students' peer awareness, we designed peer models thumbnails preview to increase the model's transparency. The study results confirmed the value of our approach and demonstrated positive effects. Figure 14 presents the Progressor interface. The detail design and evaluation are summarized in APPENDIX H.

Figure 14. Progressor interface.

3.2.4 Lessons learned from three pre-studies

In the three pre-studies summarized above, we explored various aspects of the proposed open social student modeling approach. The results of these studies allowed us to identify several features that we believe are important for the full-scale implementation of this approach. Below we summarize and discuss a set of critical features for providing successful personalized guidance using open social student modeling.

- **Sequence:** topic-based personalization provides the sequence of the topics and gives direction for the students to progress through the course. They also provide the flexibility to explore further topics or redo already covered topics. In the QuizMap study, the topic nodes in the tree hierarchy were flat and non-sequential. It was challenging for students to identify the course structure. However, once we improved the design by providing a clear sequence for progressing through the topics in the Parallel IntrospectiveViews and Progressor studies, students benefited from the general guideline of the course structure and explored more the diverse topics that were appropriate for them at the moment. Then we learned that the topic-based personalization in open social student modeling visualization worked more effectively when a *sequence* feature was implemented. In addition, we have also found that strong students tended to explore ahead of the class and weak students tended to follow them, even for the topics that were outside the current scope.

- **Identity:** identity captures all the information belonging to the student. It is a representation of the student's unique model as well as one of the main entrances to interact with the domain content. From the QuizMap study, we learned that the representation of coloring the student's own model to contrast with the rest of the student models is not enough. This addressed the differences between the student herself and the rest of the class, but did not carve out a clear model unit that belonged to the student. All the models coexisted in one detailed view (Brusilovsky, Hsiao, & Folajimi, 2011). Later on, in Parallel IntrospectiveViews study, we utilized the concept of *unity* which proposed that the perception of identity is higher if the model represents unity. This concept makes the students identify themselves with the model and allows them to easily compare themselves each other (Bull & Kay, 2007; Chen, et al., 2007).

- **Interactivity:** interactivity in the visualization of the user model can be implemented in several forms. Based on our pre-studies experiences, we learned that students benefited a lot from accessing content by directly clicking on the student's own model. The idea is simple but effective, as the visualization of the user model is not a secondary widget but the main entrance allowing the students to access content directly. Moreover, students are also enabled to interact with content through their peers' models, or interact with their peers by comparing and sorting their performances. Such interactions provide students with direct access to the learning content, at the same time allowing them to visualize different level of details of the aggregated information. It allows students to deal with the complexity and the manipulation allows them to feel in control over their models (Kay, 1997).

- **Comparison:** letting students compare themselves with each other is key for encouraging more work and better performance (Dijkstra, et al., 2008). In the formative studies, we found evidence that students made interactions through their peers' models. Moreover, this dissertation is mainly driven by the underlying supporting theory of *Social Comparison*. We believe that socially exposing models implicitly forces the students to perform cognitive comparisons. From the Parallel IntrospectiveViews study to the Progressor study, we learned that lowering the cognitive loads for making comparisons could result in encouraging more interactions.

- **Transparency:** through the pre-studies, we gradually implemented transparency for the student models. In QuizMap, students models were spread and scattered into different sizes and colors of cell in the TreeMap. It was easy to have a holistic view, but it was difficult to compare one student to another. In Parallel IntrospectiveViews, students were able to select from a list of their peers' names to access the student models one by one. Each one was represented in the pie-shape. The unity characteristic had improved the level of transparency for easier recognition of student models. However, this accessibility did not increase the peer awareness. Therefore, in the Progressor study, the peer models' representation was greatly enhanced by providing the preview with the unity. In turn, this increased peer awareness and resulted in better student performance.

3.3 APPLICATION: PROGRESSOR$^+$

To research the framework of cross content navigation in the open social student modeling visualization, we implemented a tabular interface - Progressor$^+$, named after Progressor. In this section, we describe the design rationale and the interface of Progressor$^+$.

3.3.1 The design rationale

Based on our pre-studies in earlier section, from QuizMap (Brusilovsky, et al., 2011) , Paralell IntrospectiveViews (Hsiao, Bakalov,Brusilovsky, & König-Ries, 2011) to Progressor (Bakalov, Hsiao, Brusilovsky, & Konig-Ries, 2011), we have examined the feasibility of fusing navigation support in open social student modeling visualizations. However, the goal of this dissertation work is not only to validate of the approach but also to bring this approach into actual practice. To achieve this goal, we have to verify this approach in a close-to-reality scenario.

To do so, we first acknowledge that the approach works for a single collection of educational content with navigation support in the open social student modeling visualization. The collection of educational content is a set of self-assessment questions. Given that the questions covered all ranges of topics, the materials focus mainly on the students' problem-solving skills, which may not be representative enough for a realistic online learning environment. Therefore, we chose to incorporate another well-established educational content area - annotated examples, to increase the diversity of online learning objects. Our challenge then became to blend the mix of diverse content collections on to the social visualization interface. *Do the pie-shapes of Progressor fit in this context?* In Progressor, the capability to accommodate cross content display was barely implemented. If we would have sliced the pie to represent multiple content area into smaller sectors, navigating and comparing segments of pie graphs in a huge dataset may have become perceptually and cognitively too complicated in terms of taking longer time for comprehension (Gillan & Callahan, 2000). Thus, we were motivated to implement a new open social student modeling visualization for cross content navigation as a scalable and sustainable framework.

In our investigation of visualizing a large dataset, we chose to use the tabular interface for cross content navigation in open social student modeling visualization. The design rationale is inspired by the success of interacting with and visualizing large data in Table Lens (Rao & Card, 1994). We attempt to utilize the most salient feature of a table by providing a coherent set of information in rows and columns to represent the mix of collections of different content. Based on the small multiples principle (Tufte, 1990), providing the visual constancy will allow focus on the changes. Progressor has succeeded in achieving this task by using the same pie-shapes of thumbnail peer models. We believe coherence in the rows and columns of a table could accomplish the same task. We hypothesize that regularity of the content allows students to more easily perform compound comparisons among the mixed collections of differing content. At the same time, the system still maintains a quick grasp overview. Meanwhile, such an interface potentially provides better scalability for cross content navigation through simply adding new rows.

3.3.2 The interface of the system

Progressor$^+$ is designed to visualize the student models and progress in a social manner. The interfaces are presented in Figure 15. Each student model is represented as a table with two rows as a student model unit. Each row represents one collection of educational content. The class average is represented in the same fashion as a group model. The rest of the students in the class also consist of the same forms of the student models. Essentially, all the rows are joined together and are presented in a single large table. In other words, all the student models are combined in the same big table. Each cell is colored coded, highlighting the student's progress through the topics of the collection. There are ten color shades to represent percentiles of progress (Figure 16).

Figure 15. Progressor^{+}: the tabular open social student modeling visualization interfaces

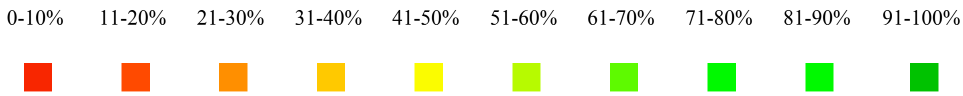


Figure 16. 10 color shades and corresponding percentiles.

There are several other table layout options available for students, including a collapse view, an expansion view and a filtered view. The collapse and expansion views are used to focus on the target student model or the specific type of content. Students are able to manipulate the views for model comparisons or detail inspections. The filtered view requires a criterion selection to refine the exploration view. The filtering criteria include sorting the progress by content types and sorting by success rate. The default setting of Progressor^{+} is configured as fully expanded table rows of the whole community and sorted by average progress in descending order.

In this dissertation, we incorporate two sizable pools of educational content: QuizJET quizzes and WebEx examples. QuizJET provides online self-assessment exercises in the domain of Java programming. A typical exercise requires students to analyze a simple Java program and answer a question about the final value of one of the variables or the console output produced by the program. QuizJET is described in more detail in Section 3.1. WebEx provides annotated programming examples. An annotated programming example can consist of fragments of codes or a complete program with line-by-line annotations. WebEx is described in more detail in APPENDIX E. Note that Progressor$^+$ itself does not serve the learning content nor support the main learning activity; it is developed as a value-added service. To access the content, students interact directly with the Progressor$^+$ table cells by clicking on the intersection of the topic and the content type (Figure 15 bottom right). Once this is selected, a panel of the lists of content will be presented along with usage details for each content item. For instance, how many attempts have there been on the question? How many times has the question been successfully solved? How many lines of annotations have been studied?

3.4 EXPERIMENTAL DESIGN

To achieve the objectives of this dissertation we designed a semester-long classroom study by providing the system as one of the supplemental course tools for the class. Semester-long classroom use will allow us to obtain a realistic longer-term use of the technology compared to the regular 2 hours lab study. It will also capture the real scenario of the curriculum on all ranges of course topics. More importantly, it will allow us to measure the long-term student engagement. To validate our hypotheses, the study will be compared to three other classroom studies. All three other classroom studies featured the same classes, same kinds of students, same course materials (including textbooks, slides, assignments, exams), same course schedule, same pre-/post- tests, and same set of self-assessment questions and annotated examples.

The classroom studies were carried out in the undergraduate course of "*Fundamentals of Object-Oriented Programming"* offered at School of Information Sciences, University of Pittsburgh. This is a required course for Information Science majors. The students registered for this course were commonly a mixture of students in Information Sciences major and students undeclared majors from the *School of Art and Sciences*. Only a few students from other sciences or engineering related degree program registered for this course. QuizJET was introduced in a 2008 Spring semester; JavaGuide was introduced in a 2008 Fall semester; Progressor was introduced in a 2011 Spring semester and Progressor$^+$ was introduced in a 2012 Spring semester. The Progressor conditioned semester is considered as the primarily baseline group, the instructor was the same as in the Progressor$^+$ course, whereas the QuizJET and JavaGuide semesters were taught by a different teacher. Therefore, they are considered as the secondary baselines. It is essential to point out that the systems were used as non-mandatory tools for the course. In this dissertation work, we consider the groups of students who used the systems as the sample of volunteered subjects. Table 2 shows the composition of the conditions and the participants of all the classroom studies, including the number of students, male and female composition, weak and strong distribution, average scores in the pre-tests.

Table 2. Study conditions & participants

	Conditions			
	Secondary Baselines		Primary Baseline	Experiment
Semester	2008 Spring	2008 Fall	2011 Spring	2012 Spring
Systems	QuizJET	JavaGuide	Progressor	Progressor+
Content	Quizzes[2]	Quizzes[5]	Quizzes[5]	Quizzes, Examples
Number of the students				
Overall	31	38	51	56
Working with the system	16 (52%)	22 (58%)	30 (59%)	38 (68%)
Male/Female student distribution				
Overall	25 / 6	27 / 11	36 / 15	44 / 12
Working with the system	13 / 3	16 / 6	23 / 7	32 / 9
Weak / Strong student distribution				
Overall	16 / 15	30 / 8	41 / 10	49 / 7
Working with the system	6 / 9	14 / 5[3]	26 / 4	34 / 4
Average scores in pre-test				
Overall	10.18	4.97	3.53	3.20
Working with the system	10.20	2.68	3.67	3.05
IS majored / others (undeclared, mechanical engineering, biomedical informatics)				
Overall	25 / 6	21 / 17	23 / 28	23 /33
Working with the system	12 / 4	10 / 12	8 / 22	17 / 21

[2] Examples were also available to the class through a traditional course management portal instead of having the navigational support through the social visualization interface

[3] Three students working with the system in the Fall 2008 semester did not take the pre-test.

All four classes were given the same pre-test during the first week to collect their pre-knowledge of the course. The systems were introduced to the classes in the third week of each semester and were available for the students from then on, for an overall fifteen week time period. During the fifteen weeks, students voluntarily logged on to the systems and worked on the QuizJET exercises or/and the WebEx examples. Students were instructed in how to use the systems and advised to use them, but such use was not mandatory for the course work. The post-tests were administrated at the 16th week of classes to measure the students' learning. A questionnaire survey was given shortly after the post-tests. There were four exams including the final exam across each semester; they were the important evaluation time marks and scheduled at the 5th, 9th, 15th and 17th week of the semester accordingly. The experiment schedule and course time line were sketched in Figure 17.

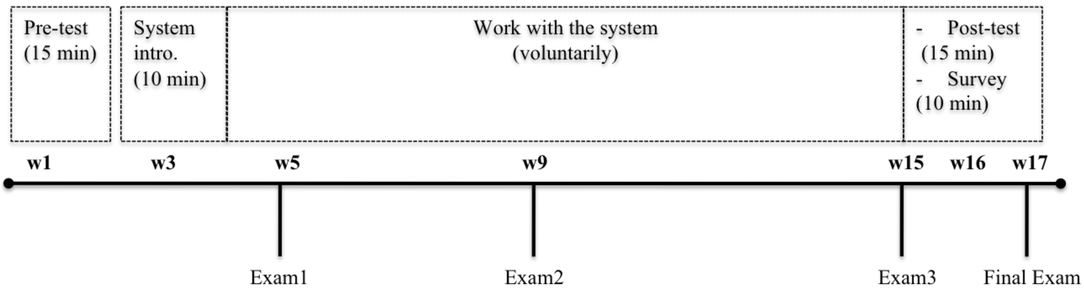


Figure 17. Experiment and course schedule

To ensure that the student cohorts were comparable, we first examined the students' pre-test scores. A one-way between-subjects analysis of variance was performed on the pre-test scores as a function of 4 different interfaces (QuizJET, JavaGuide, Progressor, and Progressor^{+}). We found that the students who used the QuizJET (*M*=10.20, *SE*=0.048) system had significant higher pre-knowledge than the average of the other three systems (*M*=3.13, *SE*=0.048), $F(3, 99)= 3.258$, $p= 0.025$[4]. The assumption of homogeneity of variance was met, Brown-Forsythe $F(3, 99)= 2.750$, $p= .052$. The assumption of normality was only met for the

[4] That the students who used QuizJET had significant higher pre-tests scores could be attributed to two reasons. 1) there were stronger students used the system that term. 2) there were more Information Sciences majored students using the system. 3) there were more repeaters from previous semester, which had already been given the pre-tests once.

QuizJET group (Table 3).

Table 3. Test of normality of the pre-test scores for each system

System	Shapiro-Wilk *W*	*df*	*p*
QuizJET	.923	16	.186
JavaGuide	.816	22	.002
Progressor	.838	30	.000
Progressor+	.897	38	.002

3.5 DATA COLLECTION

The main objective of this dissertation work is to investigate students' motivation, engagement and learning results of using Progressor$^+$ to access multiple collections of educational resources. For that purpose, we collected three kinds of data from the classroom studies:

1. The results of pre- and post-tests.

Each pre- and post-test consisted of ten questions from all the topics covered in the course, ranging from objects, classes, decisions, iterations, data structures, interfaces to inheritance etc. The questions were designed to assess students' understandings by asking students to evaluate a piece of given code in the target context. The complete pre- and post-tests can be found in APPENDIX I.

2. The transactional log data of students' interactions with the systems. The log data consists of two parts:

- Content interaction: includes the content interactions of QuizJET quizzes and WebEx examples. We record every student click on any content provided to him or her. A

QuizJET quiz will be recorded by the *student-id, group-id, session-id, question-id, answer-correctness and time*. A WebEx example will be recorded by the *students-id, group-id, session-id, example-line-id and time.*

- Social visualization interaction, including:
    - o Content selection: show QuizJET quizzes only, show WebEx example only or show both; the default shows both. The *student-id, group-id, session-id, content-type <quiz, example or mix> and time* are recorded.
    - o Progress sorting (sort by quizzes, sort by examples or sort the average of both; default sorts the average of both collections). The *student-id, group-id, session-id, content-progress <quiz, example or mix> and time* are recorded.
    - o Knowledge sorting (sort by the quizzes success). The *student-id, group-id, session-id, content-knowledge <quiz> and time* are recorded.
    - o Social comparisons (compare to a specific student or compare to the class on average). The *student-id, group-id, session-id, comparisons-type <student-id or class-average >, content-type <quiz, example or mix> and time* are recorded.

3. The subjective evaluation of the systems based on the questionnaires.

Due to the systems being non-mandatory for the course, not all students filled out the questionnaires regarding to the system use. The survey questions were designed to explore the users' subjective opinions. There were five aspects, including *Usefulness*, *Ease of Use*, *Ease of Learning*, *Satisfaction* and *Privacy and Data Sharing*. In addition, other comments on the systems were also collected in free text formats. Questionnaires are attached in APPENDIX J.

3.6 OUTCOME VARIABLES

In order to investigate the effects on students' motivation, engagement, and learning of using Progressor$^+$ to access multiple collections of educational materials, we needed to find a set of outcome variable to measure the impact of the proposed technology on these factors. These variables will allow us to formulate the research questions as a set of hypotheses that can be assessed in the study. Most of the variables for the investigation can be extracted from the student logs. Here we follow our prior work on evaluating the navigation support and interfaces and use the same set of parameters (students' participation, course coverage and feature usage) as consistent measurements. The detail definition for each parameter is summarized in Table 4.

Table 4. Definitions for parameters used

Parameter	**Definition**
Questions	Number of questions that a student attempts to solve
Success rate	Number of questions correctly answered divided by all attempts
Examples	Number of examples that a student explores
Lines	Number of lines that a student explores
Exploration rate	Number of lines explored divided by all explored example lines
Topic coverage	Distinct number of topics viewed
Question coverage	Distinct number of questions attempted
Example coverage	Distinct number of examples explored
Line coverage	Distinct number of lines explored

Using these variables and other data we have developed several ways to measure the expected outcome. The outcome measurements are discussed below:

- **Motivation & Engagement**:

In investigating students' motivation and engagement, we hypothesize that *students are motivated and engaged in using Progressor*$^+$ and produce more quantities of interactions and higher coverage. Specifically, we expect the *Attempts*, *Time* and the diversity of the content explored will increase.

First of all, we summarize the systems' usage to gauge the students' motivation and engagement. The independent variables include the question *Attempts*, the explored examples, the explored example lines, the course coverage (distinct topics, distinct questions and distinct examples) and the time spent on interacting with the systems.

Secondly, following the topic-based personalization guidance, students are expected to focus on the "current" topics (Zone A – lecture stream zone in Figure 18) (Brusilovsky, et al., 2009). In Figure 18, the shaded areas in Zone C & D are the regions of the off-"current" course topic activities, which are the self-motivated activities performed by the students themselves. Thus, we measure the ratio of students' activity performed outside the current course focus to the topic coverage that a student roams and works with in the system. The computational notation is presented in Equation 1, where *m* is denoted as motivation and *i* stands for each student. We called this indicator the M-ratio. To better understand deeper of the intensity of students' motivation, such a ratio can be further divided into two statistics, *forward* r_m and *backward* r_m, where *forward* r_m represents the ratio of moving ahead of current course focus and *backward* r_m represents revisiting past topics. Both statistics explain the students' self-motivation to work on the content through the systems. The canonical formula is presented in Equation 2. For the M-ratio, we used the number of actions in the Zone C & D divided by the total number of actions. To calculate the measure of *forward* r_m, we used Zone A & D, where we used Zone A & C to calculate the measure of *backward* r_m.

Equation 1: M-ratio

$$r_m = \frac{\#outsideScopeTopic_i}{\#attemptedTopic_i}$$

Equation 2: the canonical M-ratio

$$r_m = forward_{r_m} + backward_{r_m}$$

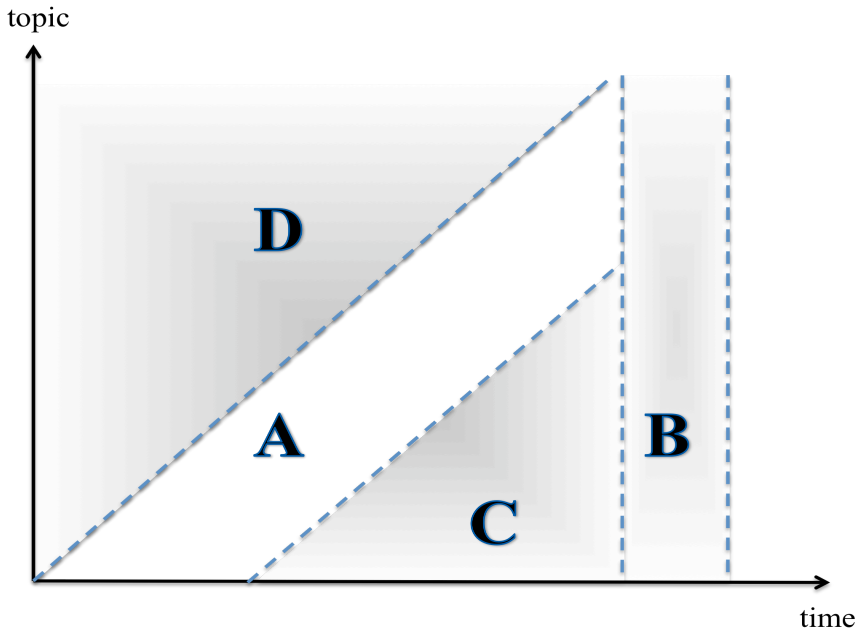


Figure 18. Projected self-motivated activities

- **Learning**:

In investigating students' learning results, we hypothesize that *students will benefit from Progressor*$^+$ and result in higher absolute knowledge gain. Meanwhile, we expect multiple collections of content will result in the highest normalized knowledge gain.

Therefore, we use pre-test and post-test scores to measure the students' knowledge gain. The canonical formula of the student's *Absolute Knowledge Gain* is denoted as the differences between pre-test and post-test scores (Equation 2). The normalized knowledge gain, also be computed based on Equation 3.

Equation 2: Absolute knowledge gain

$$\textbf{\textit{KnowledgeGain}} = \textbf{\textit{Score}}_{\textit{post-test}} - \textbf{\textit{Score}}_{\textit{pre-test}}$$

Equation 3: Normalized knowledge gain

$$\textbf{\textit{NormalizedKnowledgeGain}} = \frac{\textbf{\textit{PosttestScore - PretestScore}}}{\textbf{\textit{1-PretestScore}}}$$

- **Navigation quality**:

In examining the navigation quality in Progressor$^+$, we hypothesize that *providing navigation support in Progressor$^+$ will guide students to the right content at the right time.* Specifically, we expect students will be guided to the right levels of questions and as a result, achieve a high *Success Rate* in answering the questions.

Therefore, to assess navigation quality, we measure the success of the students' answers to the self-assessment questions. Success Rate is the percentage of correctly answered questions calculated as the total number of questions attempted divided by total score (number of correctly answered question). Note that each self-assessment question is parameterized, which means a question may be attempted several times. Each of these attempts (correct or incorrect) is counted in the activity and the success parameters. Typically, students work with the same question until the very first successful attempt, however, a number of students keep working with the questions even after the first success. Success Rate allows us to understand the average statistics of the student's problem solving skill.

3.7 SUMMARY OF THE STUDY HYPOTHESES

Based on the defined measurements, we formulate our research questions into hypotheses according to the outcome variables.

- **Motivation & Engagement**:

Hypothesis 1.1:

There will be no significant difference between Progressor and Progressor$^+$ in the question Attempts.

Hypothesis 1.2:

Students will have significantly more question attempts in both Progressor and Progressor$^+$ than in QuizJET.

Hypothesis 1.3:

Students will explore more examples & lines in Progressor$^+$ than in Progressor.

Hypothesis 1.4:

Students will explore significantly more examples & lines in Progressor$^+$ than in QuizJET.

Hypothesis 1.5:

Students will attempt significantly more distinct questions in Progressor$^+$ than in QuizJET.

Hypothesis 1.6:

Students will explore significantly more distinct examples and lines in Progressor$^+$ than in QuizJET.

Hypothesis 1.7:

There will be no significant differences in time spent in working on quizzes *between Progressor$^+$ and Progressor.*

<u>Hypothesis 1.8:</u>

Students will spend significantly more time on working with the quizzes in Progressor$^+$ than in Progressor or QuizJET.

<u>Hypothesis 1.9:</u>

Students will spend significantly more time in studying examples in Progressor$^+$ than Progressor or QuizJET.

<u>Hypothesis 1.10:</u>

There will be no significant difference in the M-ratios between Progressor$^+$ and Progressor.

<u>Hypothesis 1.11:</u>

Both Progressor$^+$ and Progressor will have significantly higher M-ratios than QuizJET.

- **Learning:**

<u>Hypothesis 2.1:</u>

The post-tests scores will be significantly greater than the pre-tests scores after using Progressor$^+$.

<u>Hypothesis 2.2:</u>

Students will achieve significantly higher normalized knowledge gain after using Progressor$^+$ than JavaGuide.

<u>Hypothesis 2.3:</u>

Students will achieve significantly higher normalized knowledge gain after using Progressor$^+$ than QuizJET.

- **Navigation quality:**

Hypothesis 3.1:

Students will achieve significantly higher Success Rate in Progressor$^+$ than in QuizJET.

Hypothesis 3.2:

There will be no significant differences of Success Rate between Progressor$^+$ and Progressor.

4.0 RESULTS

We summarize the differences between the conditions and the main direction of the effects of this work that we anticipated discovering for both collections of content (Figure 19). In Table 5 and Table 6, we present all the parameters' average statistics for both content collections in all the conditions. The table will be broken down and dissected in detail in the following subsections: 1) The impact on motivation and engagement; 2) The impact on students' learning 3) The navigation quality; 4) The social mechanism; 5) The subjective evaluation.

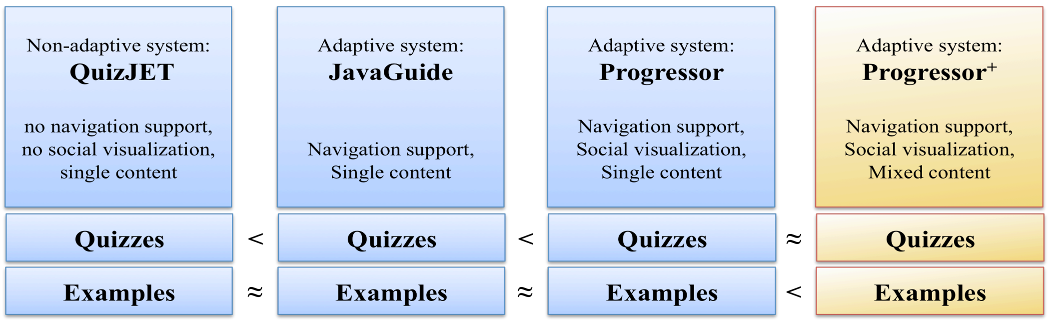


Figure 19. Expected effects of the conditions

Table 5. Summary of all parameter statistics of self-assessment quizzes collection

Quiz					
	Parameters	QuizJET	JavaGuide	Progressor	Progressor$^+$
	Active users	16	22	30	38
Quantity	Attempt	80.81±22.06	125.5±25.66	205.73±40.46	190.42±21.20
	Success	42.63%±1.99%	58.31%±2.74%	68.39%±4.32%	71.20%±4.49%
	Session	3.75±0.53	4.14±0.65	8.4±1.39	5.18±0.55
Coverage	Distinct topics	7.81±1.64	11.77±1.07	11.47±1.34	12.92±0.90
	Distinct questions	33.37±6.50	46.18±6.11	52.70±6.92	61.84±4.49

Table 6. Summary of all parameter statistics of annotated examples collection

	Parameters	QuizJET	JavaGuide	Progressor	Progressor$^+$
	Active users	21	20	7	35
Quantity	Example	10.86	19.75	28.71	27.37
	Line	104.24	116.6	219.71	184.18
	Session	4.42	5.35	5.50	4.94
Coverage	Distinct topics	8.48	9.15	12.28	12.20
	Distinct examples	10.86	17.3	25.125	27.37
	Distinct lines	80.33	67.1	115.22	141.5

4.1 IMPACT ON MOTIVATION & ENGAGEMENT

One of the main hypotheses of this dissertation is that providing navigation support and social visualization by using open student modeling interfaces will increase students' motivation and engagement within a mixed collection of educational content. To validate students' motivation and engagement, we itemize several sub-hypotheses as listed below. All the sub-hypotheses are composed by the quantity measures, which were introduced in the previous chapter.

4.1.1 Question attempts

Hypothesis 1.1:

There will be no significant difference between Progressor and Progressor$^+$ in the question Attempts.

Hypothesis 1.2:

Students will have significantly more question attempts in both Progressor and Progressor$^+$ than in QuizJET.

In this dissertation, we are evaluating the navigation support and social visualization combined approach with multiple content collections in Progressor$^+$ as if in a more realistic learning scenario. Therefore, we can prove that this approach will work in a scalable content framework only if we can demonstrate there is no significant difference in the amount of work done between Progressor and Progressor$^+$. Meanwhile, the amount work done with those two systems is significantly higher than with the non-adaptive system, QuizJET. We performed one-way between-subjects analysis of variance on the quantity of the work done as a function of system conditions. Table 7 summarizes the test results of the two collections of work for the three conditions. As we anticipated, we did not find significant differences in the amount of work done between Progressor and Progressor$^+$. This demonstrates that incorporating annotated examples in Progressor$^+$ did not sacrifice the self-assessment questions usage, which confirmed Hypothesis 1.1.

Table 7. The statistics for comparing the amount of work done among systems

		F-stats	*p*-value
questions	QuizJET (*M*=80.81, *SE*=27.13) vs. Progressor (*M*=205.73, *SE*=27.13)	*F*(1, 44)=24.20	<0.001
	QuizJET (*M*=80.81, *SE*=27.13) vs. Progressor$^+$ (*M*=190.42, *SE*=27.13)	*F*(1, 52)=23.72	<0.001
examples	QuizJET (*M*=10.86, *SE*=4.22) vs. Progressor (*M*=28.71, *SE*=4.22)	*F*(1, 26)=12.13	<0.001
	QuizJET (*M*=10.86, *SE*=4.22) vs. Progressor$^+$ (*M*=27.37, *SE*=4.22)	*F*(1, 54)=11.89	<0.001
lines	QuizJET (*M*=104.24, *SE*=21.32) vs. Progressor (*M*=219.71, *SE*=21.32)	*F*(1, 26)=9.55	<0.001
	QuizJET (*M*=104.24, *SE*=21.32) vs. Progressor$^+$ (*M*=184.18, *SE*=21.32)	*F*(1, 54)=7.11	0.007

To prove that the adaptive navigation support combined with the social visualization approach will work in a mixed collection of educational content, we have to show that this approach shows an increase in educational activities performed with the non-adaptive system. The statistical analysis showed that indeed, both Progressor's and Progressor+'s users completed significantly higher amount of question attempts than QuizJET. The results confirmed Hypothesis 1.2, which verifies that our approach motivated the students to put more effort into working with the systems.

4.1.2 Amount of work with examples

Hypothesis 1.3:

Students will explore more examples & lines in Progressor$^+$ than in Progressor.

Hypothesis 1.4:

Students will explore significantly more examples & lines in Progressor$^+$ than in QuizJET.

First of all, we did not find that significantly more examples were explored in Progressor$^+$ than in Progressor. Unfortunately, the Hypothesis 1.3 is rejected. However, when we looked into the reasons why there were no significant differences of the example average usage between two systems, we found that Progressor$^+$ involved a substantial amount of the student participation rate (62.5%) while Progressor only engaged a handful of students (13.7%). This demonstrated that Progressor$^+$ encouraged the participation for using annotated examples. On the other hand, without using the proposed approach for annotated examples in Progressor, only the active students used the content. Therefore, comparing the average statistics does not represent the power of the annotated example collection. Thus, Hypothesis 1.3 should be refined as *Students who actively use the system will explore significant more examples & lines in Progressor$^+$ than in Progressor*. Thus, to verify the adjusted hypothesis, we consider the average statistics of the active students and we found that students indeed explored more examples in Progressor$^+$ (M=50.29, SE=2.07) than in Progressor (M=25.13, SE=2.07), F(1, 12)= 10.17, p<.01. In addition, students explored more lines in Progressor^{+} (M=223.11, SE=5.20) than in Progressor (M=115.22, SE=5.20), F(1, 12)= 5.07, p<.05. When we compared the amount of work with examples with non-adaptive system – QuizJET, and we found that students explored significantly more examples in Progressor^{+} (M=27.37, SE=4.22) than in QuizJET (M=10.86, SE=4.22), F(1, 54)=11.89, p<.01. These results demonstrated that the dissertation approach successfully motivated students to do more work for multiple collections of content in Progressor^{+}. Meanwhile, it confirmed Hypothesis 1.4.

4.1.3 Course coverage

<u>Hypothesis 1.5:</u>

Students will attempt significantly more distinct questions in Progressor^{+} than in QuizJET.

<u>Hypothesis 1.6:</u>

Students will explore significantly more distinct examples and lines in Progressor^{+} than in QuizJET.

To examine whether the dissertation approach is effective in guiding students navigating through multiple collections of content, we compare course coverage (the amount of distinct statistics) between Progressor^{+} and the non-adaptive system, QuizJET. We found that students attempted significantly more distinct questions in Progressor^{+} (M=61.84, SE=5.13) than in QuizJET (M=33.37, SE=5.13), F(1, 54)=18.19, p<.01. Students also explored significantly more distinct examples in Progressor^{+} (M=27.37, SE=4.22) than in QuizJET (M=10.86, SE=4.22), F(1, 54)=11.89, p<.01. These significance allow us to confirm the Hypothesis 1.5 & 1.6.

Moreover, the Pearson correlation coefficient indicated that the more diverse questions the students tried, the higher the success rate they obtained (r=0.707, p<.01) and the more diverse the examples the students studied, the higher the success rate they obtained (r=0.538, p<.01). We also looked at how frequently the students repeated questions, examples and lines. We found that the more often the students repeated the same questions and the more often the students repeated studying the same lines the higher success rate they obtained (r=0.654, p<.01; r=0.528, p<.01). The analysis of motivational effects presented in this section has suggested that the combined approach can effectively enhance students' motivation in the targeted learning context.

4.1.4 Time spent working with the content

In this sub-section, we analyze the effects of such an approach on student engagement. We hypothesize that the dissertation approach will engage students with Progressor$^+$ and spend more time in both collections of content through the system in other conditions. All the sub-hypotheses are listed below.

Hypothesis 1.7:

> *There will be no significant differences in time spent in working on quizzes between Progressor$^+$ and Progressor.*

Hypothesis 1.8:

> *Students will spend significantly more time on working with the quizzes in Progressor$^+$ than in Progressor or QuizJET.*

Hypothesis 1.9:

> *Students will spend significantly more time in studying examples in Progressor$^+$ than Progressor or QuizJET.*

In our pre-studies (section 3.2), we found that students doubled the time spent (in terms of sessions) in Progressor than QuizJET. However, we did not find this pattern in the same parameter when comparing Progressor$^+$ and QuizJET. Nevertheless, the intensity of students' work per session is actually higher in Progressor$^+$. This number hinted that our assumption that students might spend more time in Progressor$^+$ than in QuizJET could still be correct, if that time was being divided over fewer sessions. Therefore, we computed the actual average time spent for each content collection (Table 8). The results showed that students spent fewer sessions in Progressor$^+$ in quizzes, however, they did work longer per session. On average, they spent 3.72 and 4.94 times more minutes in Progressor and Progressor$^+$ than in QuizJET. The results confirmed Hypothesis 1.8. The significance is reported in the Table 8. Meanwhile, there were several other interesting findings:

1) Comparing quiz usage between JavaGuide and Progressor, we did not find a significant difference in the amount of time spent per session or the average attempts per session. We did find marginally significantly more total time spent in Progressor than in JavaGuide. These results indicated that providing personalized guidance in open social student modeling interface (Progressor) was as efficient as the non-social open student modeling interface (JavaGuide). Progressor, on the other hand, showed longer engagement by spending more time in total.

2) Comparing the usage between Progressor and Progressor$^+$, we found significantly more time spent per session in Progressor$^+$ than in Progressor. There was no significant difference between Progressor and Progressor$^+$ in total time spent on the quizzes and average attempts per session. The results confirmed the Hypothesis 1.7. Meanwhile, these results combined demonstrated that introducing annotated examples to the open social student modeling visualization did not sacrifice the usage of self-assessment quizzes. In addition, it increased the engagement per session.

As Hypotheses 1.7 and 1.8 are sustained, we confirmed the capability of our approach in the new interface, which successfully engaged students to work on the self-assessment quizzes. However, to generalize this effect from our approach for mixed collections of educational content, we have to verify Hypothesis 1.9. From the example collection, we found that students spent 4.13 and 3.23 times more minutes average per session in studying the annotated examples in Progressor$^+$ than in QuizJET and Progressor. These were both significant differences. With the adaptive navigation support and social visualizations combined, students studied more. These results showed us that our approach successfully engaged students to study more on the annotated examples without diminishing the value of working with quizzes.

Table 8. The intensity measures of students' work for all conditions

Intensity		QuizJET	JavaGuide	Progressor	Progressor$^+$
Quiz	Time/session (minutes)	16.01	36.28	26.75	57.32**
	Total time (minutes)	60.04	150.19**	224.7**	296.9**
	Attempt/session	21.55	30.31	24.49	36.73
Example	Time/session (minutes)	15.73	22.66	20.12	65.00**
	Total time (minutes)	69.52	121.23	110.66	321.1**
	Example/session	2.45	3.69	4.56	5.54
	Lines/session	23.54	21.79	34.95	38.69

Overall, each student on average spent nearly 5 hours working on the quizzes in Progressor$^+$ and 5 hours and 20 minutes studying the annotated examples. These numbers alone demonstrated that our approach successfully engaged students to work on the non-mandatory systems. In addition, we found that the more time the students spent in one type of the content in Progressor$^+$, the more likely they were to spend more time in another type of content (r=0.81, p<.01). Yet, does longer engagement lead to better learning? We will discuss the effects on students' learning in the next section.

4.1.5 The diversity of work and M-ratio

<u>Hypothesis 1.10:</u>

There will be no significant difference in the M-ratios between Progressor$^+$ and Progressor.

<u>Hypothesis 1.11:</u>

Both Progressor$^+$ and Progressor will have significantly higher M-ratios than QuizJET.

To evaluate the influence of Progressor$^+$ in motivating activity, we also calculated the average M-ratios for both content collections (Table 9). M-ratio is calculated to discover the proportion of students' self-motivation work that was performed outside the current learning scopes, referred back to Section 3.6. We found that there was no significant M-ratio difference between Progressor and Progressor$^+$ for both collections. In addition, both Progressor and Progressor$^+$ had significant higher M-ratios than QuizJET for both collections, $F(1, 44)= 2.63$, $p< .05$, $F(1, 52)= 9.88$, $p< .01$. Hence, Hypothesis 1.10 and Hypothesis 1.11 are both confirmed.

To understand the students' motivation, we broke down the M-ratio and looked at two other measures, the *forward* r_m and the *backward* r_m. We found that among those students who motivated themselves to do work outside current learning scopes, there was no significant differences in reviewing past topics. However, there was a distinct effect on previewing future topics for adaptive navigation support and social visualization combined as compared to than without this combination. The students who were self-motivated were actually driven to explore ahead of current learning focus. This effect was consistent for both collections. Such a pattern can be attributed to the social phenomenon, which will be further discussed in the subsection 4.4.

Table 9. The M-ratio characterizes the students' self-motivated activities, which estimates the percentage of students' activity performed outside the current course focus. Forward and backward M-ratios indicate the students' motivation to preview and review the topics outside the current course focus.

		QuizJET	Progressor	Progressor$^+$
Quizzes	M-ratio(r_m)	0.20±0.17	0.33±0.05	0.30±0.02
	forward r_m	0.008±0.004	0.217±0.099	0.145±0.081
	backward r_m	0.196±0.157	0.117±0.048	0.155±0.149
Examples	M-ratio(r_m)	0.161±0.098	0.375±0.074	0.380±0.152
	forward r_m	0.041±0.004	0.227±0.027	0.179±0.121
	backward r_m	0.119±0.009	0.146±0.006	0.195±0.114

4.2 IMPACT ON LEARNING: KNOWLEDGE GAIN

This analysis of an educational innovation is not complete without the analysis of its impact on students' learning. Our approach has been demonstrated to produce an impressive motivational and engagement effect on students. However, we are reminded that students were able to learn the subject in many ways: labs, lectures, assignments etc. To determine the effectiveness of our approach, we need to prove that students' activities with the systems were transformed into real students' learning. Therefore, in this subsection, we particularly associate students' interactions with Progressor$^+$ to their learning results. We consider the results of pre- and post- tests scores as the general learning gains, and hypothesize a significant growth after using Progressor$^+$. Meanwhile, we also hypothesize that the dissertation approach will result in the highest normalized knowledge gain among all other conditions.

Hypothesis 2.1:

The post-tests scores will be significantly greater than the pre-tests scores after using Progressor$^+$.

Hypothesis 2.2:

Students will achieve significantly higher normalized knowledge gain after using Progressor$^+$ than JavaGuide.

Hypothesis 2.3:

Students will achieve significantly higher normalized knowledge gain after using Progressor$^+$ than QuizJET.

It is essential for an educational innovation to demonstrate that it positively impacts students' learning. We performed paired sample t-test to evaluate the significance of the students' *Absolute Knowledge Gain* (Section 3.6). We found the students who used Progressor$^+$ indeed achieved significant higher post-test scores (M=15.0, SD=0.6) than their pre-test scores (M=3.2, SD=0.5), t(37)= 17.276, p<.01. The Hypothesis 2.1 is sustained.

In addition, we expect providing navigation support and social visualization combined will result in more learning. We performed one-way between-subjects analysis of variance on the *Normalized Knowledge Gain* as a function of 4 different systems (QuizJET, JavaGuide, Progressor and Progressor$^+$). We found that student obtained significantly greater *Normalized Knowledge Gain* by working on the self-assessment questions through Progressor$^+$ (M= 0.581, SE= 0.050) than on QuizJET (M= 0.361, SE= 0.050), $F(1, 52)$= 4.223, p<.05, η^2=.025. Therefore Hypothesis 2.2 is rejected and Hypothesis 2.3 is sustained.

Following previous motivational and engagement analyses, we also found that the more the students studied (more lines), the higher level of knowledge they gained (r=0.492,

$p<.01$). The more time the students spent on the content (quizzes and examples), the higher the level of knowledge gain they obtained ($r=0.563$, $p<.01$; $r=0.448$, $p<.01$).

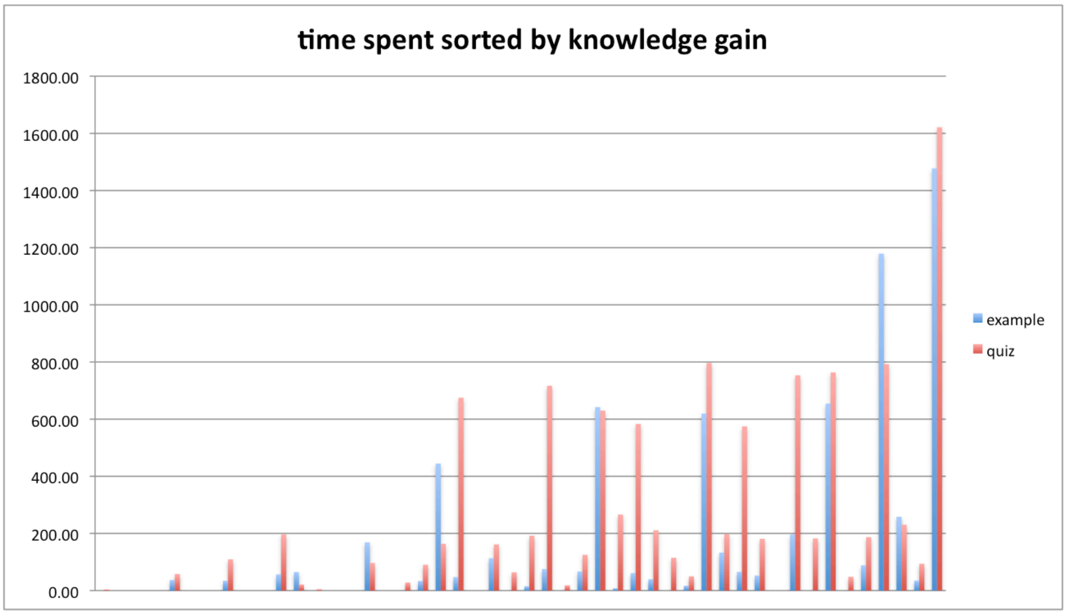


Figure 20. Students' time spent on both examples and quizzes in Progressor+ sorted by the knowledge gain

4.3 QUALITY OF NAVIGATION SUPPORT

Problem solving is an important skill acquired by learning. It has been studied so that it can enhance the transfer of concepts to new problems, yield better learning results, make acquired knowledge is more readily available and applicable especially in new contexts etc. (Dolmans, De Grave, Wolfhagen, & Van Der Vleuten, 2005; Melis, Andres, Budenbender, Frischauf, Goduadze, et al., 2001). Succinctly put, it is not independent from knowledge acquisition. In our target context, self-assessment quizzes provide the students with opportunities to practice problem solving. To evaluate the system's impact on students' problem solving success, we used the parameter – *Success Rate*, which calculates the percentage of correctly answered questions calculated as the total number of questions attempted divided by total score (number of correctly answered question).

To validate one of our main hypotheses that providing navigation support will guide students to the right content at the right time, we specify the following sub-hypotheses with the parameter – *Success Rate*. In the following subsections, we diagnose the impact of our approach by analyzing students' problem solving success in several aspects: 1) general impact on problem solving success, 2) problem solving success by content complexity, 3) problem solving success and the students' pre-knowledge. We finally summarize the findings in the subsection 4.4.4.

4.3.1 General impact on problem solving success

Hypothesis 3.1:

Students will achieve significantly higher Success Rate in Progressor$^+$ than in QuizJET.

Hypothesis 3.2:

There will be no significant differences of Success Rate between Progressor$^+$ and Progressor.

Note that each self-assessment question is parameterized, which means a question may be attempted several times. Each of these attempts (correct or incorrect) is counted in the activity and the success parameters. This not only challenges the students' ability to work on the problems but also challenges their confidence of self-assessing themselves on the problems. Typically, students work with the same question until the very first successful attempt, however, a number of students keep working with a question even after the first success. Essentially, a perfect *Success Rate* (100%) is rarely happened, which is not necessarily a bad thing. We expect the students to make mistakes and to repeatedly work on the same problem until they obtain success. It tells us that they are sure that they really understand the problem. In other words, we encourage students engage in trial-and-error and challenge them to be confident in the answer they provide to the question. Therefore, to gauge students' understanding on the problems, we compare the problem solving success across different systems.

The system usage data showed that the students achieved significantly higher *Success Rate* in Progressor (*M=0.*684*, SE=0.071*) and in Progressor$^+$ (*M=0.*712*, SE=0. 071*) than in QuizJET (*M=0.*426*, SE=0.071*), $F(1, 44)= 2.622$, $p<.05$, $\eta^2=.021$; $F(1, 52)= 11.027$, $p<.01$, $\eta^2=.017$. We also found that the students achieved a higher *Success Rate* than in JavaGuide. However, it was not significant. The results demonstrated that the navigation support in open social student modeling visualization successfully and significantly increase the students' problem solving success, where non-social navigation supported system did help, but did not make a significant impact. This proved that the navigation support and social visualization combination did indeed bring added value to the system, where the navigation support alone did not.

4.3.2 Problem-solving success and content complexity

In the past, we found that adaptive navigation support had an impact in the guidance on students' problem-solving success by different content complexities (Hsiao, et al., 2010). Students were found to be better prepared for *Easy* and *Moderate* levels of questions, without venturing too far in the *Complex* area of questions, resulting in success in all complexity levels. However, do we find the same pattern with navigation supported open social student modeling visualizations? Does the content complexity impact students' problem solving success? Do social visualizations provide positive value on top of adaptive navigation support or vice versa? In this subsection, we aim to evaluate the combined approach of adaptive navigation support and social visualization in the same context.

The results showed that the combined approach increased *Attempts* in all three levels of complexities, instead of just the easier levels. Meanwhile, the *Success Rate* significantly increased across all three levels of complexities. This indicates that the combined approach not only encouraged students to do more work earlier in the course, when the questions are relatively easy, but students were also challenged to work on harder questions. The combined approach systems produced dramatically increased use for all three levels of complexities. The increase of use in the *Complex* level was particularly interesting. Due to the nature of the complex questions, it requires a more comprehensive understanding and persistent effort to actually complete a single question. The results showed that with adaptive navigation support and social visualization combined, students managed to achieve high performance in all three levels, including the *Complex* one (Figure 21).

Based on the *Attempt per question* statistics, students were found to repeat the same level of questions more frequently. Such an outcome allowed us to conclude that the students had a solid preparation in easier levels of questions and consistently practiced the same levels of questions resulted in achieving a remarkably high success rate for all three levels of complexities (Figure 22). In addition, we found that the students using Progressor$^+$ did not repeat the Complex questions as frequently as the students did in Progressor, but they both achieved the same high level of *Success Rate*. Students worked on the appropriate quizzes and explored the annotated examples at the right time according to the progress and the wisdom of the crowd. This piece of evidence revealed the cross content navigation in the open social student modeling visualizations effectively led the students to work at the right level of questions for their readiness. The number of the average *Attempts*, *Attempt per question* and *Success Rate* by complexity levels for all conditions were summarized in table 10.

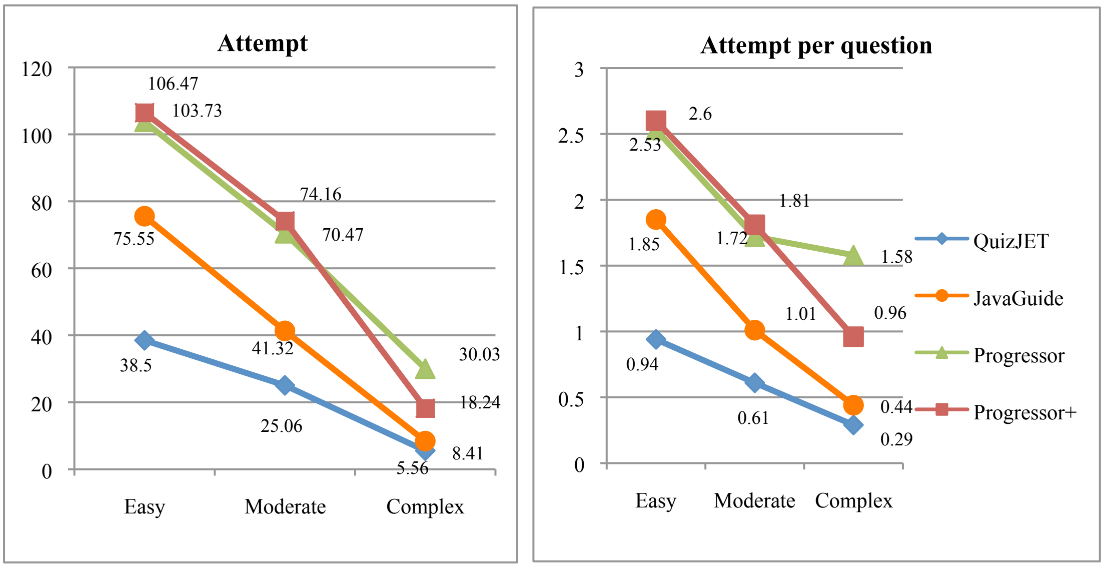


Figure 21. The average *Attempt* & *Attempt per question* of four systems on different complexity levels

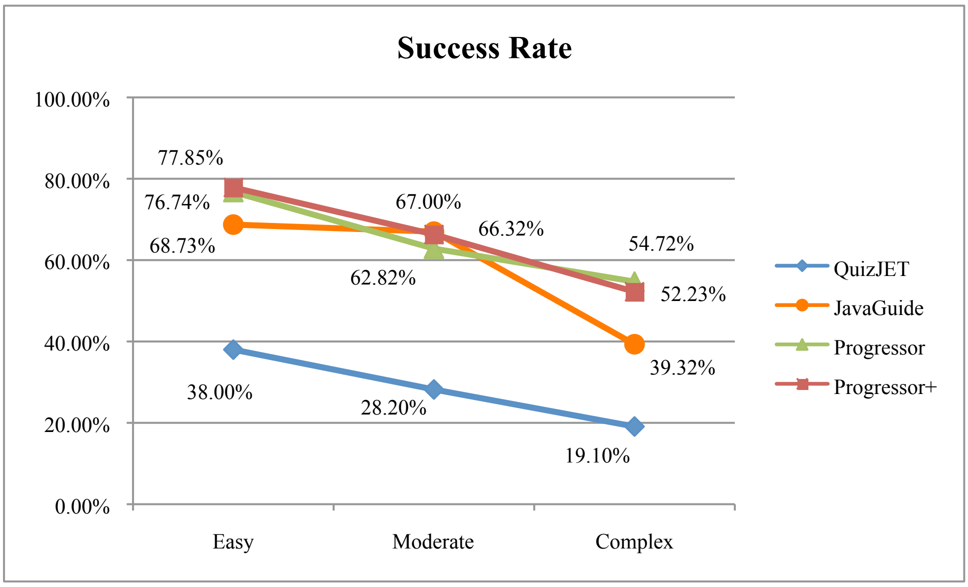


Figure 22. The *Success Rate* of three systems on different complexity levels

Table 10. The summary of the average Attempts, Attempt per question and Success Rate by complexity levels for all conditions

	QuizJET	JavaGuide	Progressor	Progressor^{+}
Attempts				
Easy	38.5	75.77	103.73	106.47
Moderate	25.06	41.32	70.47	74.16
Complex	5.56	8.41	30.03	18.24
Attempt per question				
Easy	0.94	1.85	2.53	2.6
Moderate	0.61	1.01	1.72	1.81
Complex	0.29	0.44	1.58	0.96
Success Rate				
Easy	38.00%	68.73%	76.74%	77.85%
Moderate	28.20%	67.00%	62.82%	66.32%
Complex	19.10%	39.32%	54.72%	52.23%

4.3.3 Problem-solving success and students' pre-knowledge

Since we found the assumption of normality tests of the students' pre-test scores was not met for the most of the conditions, we examined the data spread and found that there was a clear split among the data. Based on the positivlye skewed pre-test scores, we split the students into two groups – weaker and stronger groups. With their pre-test scores (ranging from a minimum 0 to a maximum 20, with the threshold at score 7), strong students scored 7 points or higher (7~13) and weak students scored less than 7 (0~6). In addition, in Figure 23, we observed a trend when we associated the time spent with the amount of work sorted by the students' pre-knowledge. It appeared that the low pre-knowledge students tended to spend more time on the content. This motivated us to check the nitty-gritty of students' work by their pre-knowledge. More importantly, the pre-knowledge differences allow us to examine the hypothesis that whether navigation support indeed guides students with different prior knowledge to the right content at the right time.

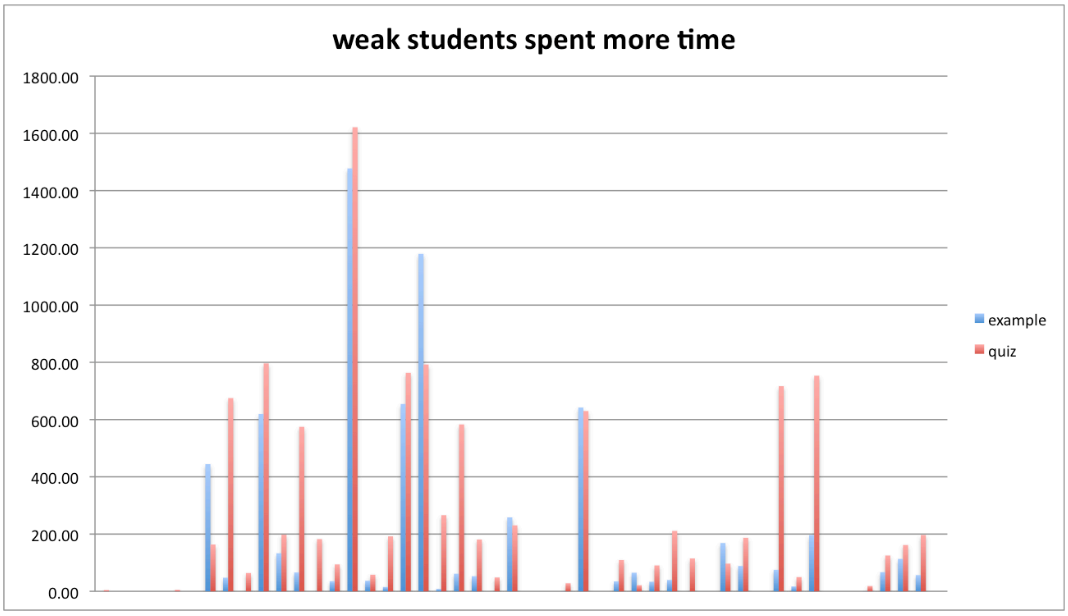


Figure 23. The time spent for each collection for Progressor$^+$ users sorted by students' pre-knowledge from low to high

In Figure 24a & 24b, we depicted the *Attempt per question* and *Success Rate* by students' pre-knowledge. Again, students' pre-knowledge was split by their pre-tests scores (ranging from a minimum 0 to a maximum 20, with the threshold at score 7), strong students scored 7 points or higher (7~13) and weak students scored less than 7 (0~6). Strong students were painted in darker colors and weak students were painted in paler colors. QuizJET, JavaGuide, Progressor and Progressor$^+$ were represented in blue, orange, green and red accordingly. Both strong and weak students' activities in all conditions across all levels of complexities statistics were summarized in Table 11. Here are several interesting findings:

1) The QuizJET users (blue lines) showed no specific pattern in the Attempt per question by the students' pre-knowledge. However, we found that stronger students had

significant higher success rate in easier levels of question complexities than weaker students. There was a gap between strong and weak students.

2) With Progressor users (green lines), it was found that both strong and weak students had the same high statistics on attempt per question for all three levels of complexities and achieved higher success rates than with QuizJET. However, the strong and weak students had erratic patterns on their *Success Rate.* The possible explanation was the weaker students followed the traces left by the stronger ones and did the same amount of work as the stronger did. However, they may not have put enough effort as they were supposed to. The differences got noticeable when the questions became more complex. Therefore the gaps became significant.

3) the red lines represent the Progressor$^+$ users. We found that the weaker students on average worked significantly more than the stronger ones, and they manage to produce the same high *Success Rate*. It is understandable that the weaker students required more attempts to digest the content while the stronger ones did not. Students voluntarily studied the annotated examples or simply repeated working on the self-assessment questions to get themselves ready. This is good news when both strong and weak student worked with the appropriate levels of questions and managed a consistent performance across all three levels of complexities. Meanwhile, this also showed that the both strong and weak students resulted in a uniform cohort in terms of the problem-solving success.

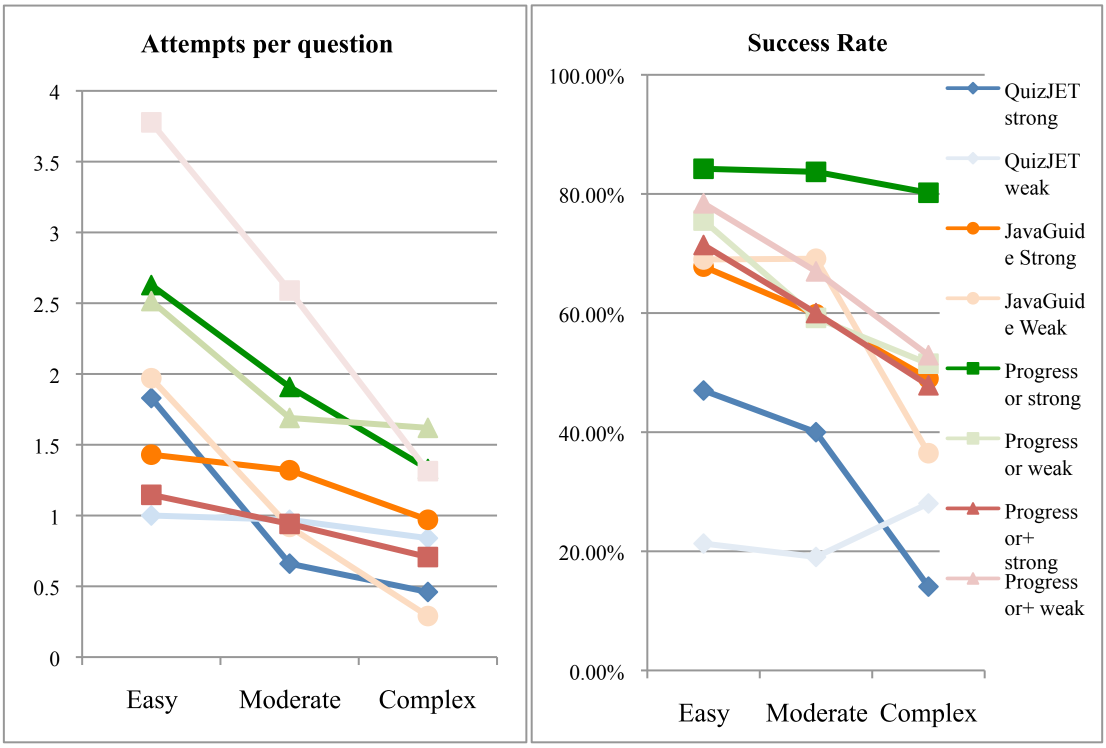


Figure 24-a & 24-b. The pattern of differences of *Attempt per question* and *Success Rate* for three systems on a variety of students' pre-knowledge and question complexities

Table 11. The summary of the total *Attempt* per question and *Success* Rate on a variety of students' pre-knowledge and complexity levels for all systems

Attempt per question		Easy	Moderate	Complex
QuizJET	Strong	1.83	0.66	0.46
	Weak	1.00	0.97	0.84
JavaGuide	Strong	1.43	1.32	0.97
	Weak	1.97	0.92	0.29
Progressor	Strong	2.63	1.91	1.33
	Weak	2.52	1.69	1.62
Progressor$^+$	Strong	1.15	0.94	0.71
	Weak	3.78	2.59	1.31
Success Rate		**Easy**	**Moderate**	**Complex**
QuizJET	Strong	47.00%	40.00%	14.07%

	Weak	21.30%	19.05%	28.03%
JavaGuide	Strong	67.80%	59.80%	49.00%
	Weak	69.00%	69.12%	36.47%
Progressor	Strong	84.22%	83.71%	80.20%
	Weak	75.53%	59.19%	51.50%
Progressor^{+}	Strong	71.43%	60.00%	47.87%
	Weak	78.42%	66.99%	52.92%

4.3.4 Summary of the findings

The analysis of the impact on students' problem solving success tells us that adding navigation support on open social student modeling visualizations helped students to achieve a significantly higher *Success Rate*. In addition, incorporating a mixed collection of content in the open social student modeling visualizations effectively led the students to work at the right level of questions. Both strong and weak students worked with the appropriate levels of questions for their readiness, yielding consistent performance across all three levels of complexities.

The results show that combining adaptive navigation support in the open social student modeling visualization effectively guides students to the right content at the right time. Additionally, providing a more realistic content collection on the navigation supported open social student modeling visualizations results in uniform performance for the group.

4.4 THE MECHANISM OF SOCIAL GUIDANCE

Our study demonstrated that social guidance can match or even surpass traditional knowledge-based guidance in its ability to guide students to the right content in the right time. But what is the mechanism of social guidance? Why is the progress data collected from the class and presented in visual form able to provide this remarkable quality of guidance, matching guidance based on expert knowledge?

In the previous studies, we found that strong students tended to explore the content ahead of weak ones. That was an important mechanism of our approach to provide social guidance where stronger students would leave traces for weaker ones to follow. However, that pattern was only found within the context of self-assessment quizzes. Do we find the same pattern among multiple collections of educational resources? Are the stronger students still capable of pioneering a good route for the class? Are there any other social mechanisms and effects derived from Progressor^{+}? In this subsection, we summarize the findings of social visualizations and plotted the system interactions for pattern discovery.

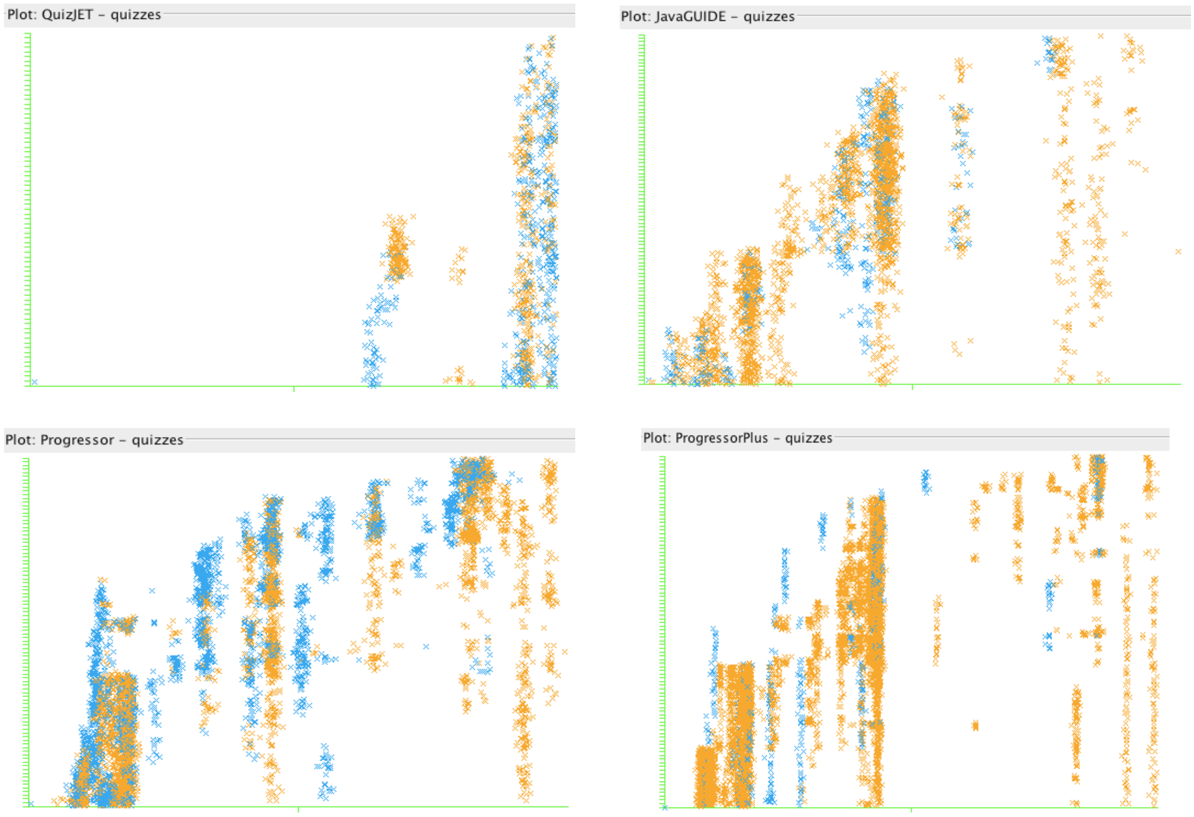


Figure 25. All quizzes attempts distribution by time and question complexity performed by the students in four systems. top-left: QuizJET(a); top-right: JavaGuide(b); bottom-left: Progressor(c); bottom-right: Progressor^{+}(d)

In Figure 25, we plotted all the students' activities on system QuizJET, JavaGuide, Progressor and Progressor^{+} by time and question complexity. The time of the interaction is marked on the X-axis and the question complexity goes from easy to complex on the Y-axis. Each data point represents an attempt at a question. The blue dots belong to the stronger students and the oranges ones belong to the weaker ones. By visualizing all the interactions performed on the systems, we observed several interesting findings.

- There is a general pattern for all conditions, which is that the students were found actively working with the systems during exams preparation periods. They tended to work on the topics from past to current. During the final exam period, students tended to review all ranges of the topics. Due to the fact that the subject is inherently cumulative in nature, we expect to find this pattern as a stable effect.

- With topic-based personalization (b&c&d), there were noticeable trends that students progressed, which resulted in more work done according to the lecture stream. This is an important message that students were focusing and were able to benefit from the topic-based guidance without jumping too far beyond the current scope. Without such personalization (a), students were only found to work on the systems for exam preparation, yielding a very skewed *Attempts* distribution.
- Differences in the amount of work (Attempts) were noticeable from the two figures (a&b) on the top row to the bottom two figures (c&d). The bottom two figures (c&d) represent the systems with the influence of social visualization, which resulted in higher intensity of the attempts. It not only demonstrated that the students were voluntarily engaging with the systems, but also showed the consistency of the motivational effect over time.
- The timing for beginning work in the system was also discovered by the differences of pre-knowledge levels with the social visualization mediation, where the pre-knowledge levels were determined by the pre-test scores (ranging from a minimum 0 to a maximum 20, with the threshold at score 7), strong students scored 7 points or higher (7~13) and weak students scored less than 7 (0~6). The strong students tended to explore the questions ahead of the weaker ones (the blue dots go before the orange dots) in social visualization systems (c&d). In Table 12, we calculated the average time that the strong students attempted the question before the weak students did across all ranges of question complexities. On average, strong students worked on the questions 38.04 and 37.70 hours in advanced compared to the weak students. The effect was much more noticeable in the Complex questions. This shows good implicit social guidance in that good students left the traces for bad students to follow. Without the social guidance, there were no clear patterns found (a&b). Strong and

weak students' actions were mixed. Strong ones may be under challenged, while the weak ones may suffer from venturing too fast for advanced questions.

- A model exposure difference was found between two social visualization systems (c&d). Both Progressor and Progressor$^+$ users were exposed to the entire model, from each individual's to the class. However, the pie shape model in Progressor took a relatively bigger portion of the space on the screen compared to the table model in Progressor$^+$. The model thumbnails preview was limited by the screen sizes and resulted in presenting only the top students from the class at a first glance in Progressor. Students had to scroll down the sorted model list to see the rest of the models. In Progressor$^+$, on the other hand, there was less scrolling required to view the complete model list. In other words, the top students' models seemed to stand out as highlighted models in Progressor. This may have given extra incentive for the top students, which resulted in encouraging competitiveness and hard work. Therefore, the model exposure differences explained why the stronger students in Progressor tended to work more than in Progressor$^+$.

Table 12. Strong students attempted the questions averagely ahead of weak students in hours by content complexities

(hours)	Easy	Moderate	Complex	Avg
Progressor	17.15	13.39	83.59	38.04
Progressor$^+$	9.17	19.63	84.30	37.70

4.5 SUBJECTIVE EVALUATION

In addition to the log analysis, we distributed the questionnaires to collect students' opinions on the Progressor$^+$ at the end of the classroom study. There were 24 students who filled out the survey[5], 17 male and 7 female. In the survey, there were 23 questions, including the usability of GUI elements to users' satisfaction of the interface in general. Users were asked to evaluate the questions on a 5-points Likert scale, 1 – *Strongly Disagree*; 2 – *Disagree*; 3 – *No Strong Opinion*; 4 – *Agree*; 5 – St*rongly Agree*. They were also advised to provide free-text comments as they wish. We further break down the 23 questions into 5 categories, including *Usefulness*, *Ease of Use, Ease of Learning, Satisfaction and Privacy & Data Sharing*. The itemized results are going to be discussed following. A summary of the survey is charted in Figure 26. The complete questionnaire can be found in APPENDIX J.

In the *Usefulness* category, we intended to investigate general feelings towards the system interface, key features, and content collections. For *Question A1*: we found that there was no single point of disagreement regarding the interface and the class content organization. In fact, 78% and above of the students participants agree or strongly agree that the interface helped them to understand the class content organization. *Question A2*: asks whether the interface helped the users to identify their weak points. 69% of the participants considered it did and the rest of them had no strong opinions. The results of *Questions A1 & A2* allowed us to see that the combination of topic-based and progress-based personalization successfully worked to help students mentally organize the class content, and drew their attention to the topics where they needed to focus the most. In *Question A3*, the majority of the student (63%) did not have opinions on whether the interface helped them to plan their class work. A quarter of the students actually thought negatively of it, which is understandable in a sense that the system was provided as one of the supplemental tools for the class. It was supposed to serve as complimentary resources for the class, but not as the core main class work. Therefore, it was not meant to be aligned with the class work among the assignments, exams, lab exercises, and exams. For the system functionalities' usefulness,

[5] Due to the bomb threats at the end of the semester, there were several students left campus. Therefore we did not collect all the active Progressor$^+$ users' opinions.

we asked whether the interface had helped the students to access the content in *Question A4* & whether the color indication of the progress was clear to them in *Question A5*. Students responded 96% and 88% positive accordingly. In *Question A6- A8*, we asked about the functionalities specifically. In *Question A6*, *viewing classmates' progress motivates me to progress on mine,* students had agreement slightly toward positive (21% extremely positive, 21% positive, 25% neutral, 25% negative and 8% extremely negative, with a 27% standard deviation). In *Question A7, sorting the progress helps me to find who can help on difficult topics*, students also felt positively toward agreement (17% extremely agree, 25% agree), with 21% disagree and 4% extremely disagree. In *Question A8, sorting the success helps me to find who can help on difficult topics*, 17% of the students felt extremely positive, 13% positive, 41% neutral and 25% negatively, 4% extremely negative). These results indicated that the students generally valued the open social student modeling visualization differently. However various their thoughts on the system, it was quantitatively proven to engage students. In the last section of the subjective evaluation on *Usefulness*, students were asked whether the content collections were helpful. The results were very uniform. 67% of the students felt the annotated examples were extremely helpful or helpful, 29% had no strong opinions and only 4% of them disagreed. 63% of the students considered the self-assessment quizzes extremely helpful or helpful and 37% of them had no strong opinions. There were no negative opinions found.

In the second category of the survey, we investigated the *Ease of Use* of the system. Students were asked to evaluate whether the interface was easy to use, the interface was user friendly, and the interface required the fewest steps to accomplish what they wanted to do. Students had very consistent positive results.

In the third category of the survey, students were asked to evaluate the *Ease of Learning* on the system. 79% of the students agreed or strongly agreed that they learned how to use the system quickly. 87% of them agreed or strongly agreed that they easily remembered how to use the system. 79% of them admitted that it was easy to learn how to use the system.

In the fourth category of the survey, we asked questions to gauge the students' satisfaction on the system. We found that 71% of the students were satisfied with the interface, with no complaints about the functions, interfaces or the content. When we asked whether the interface was fun to use and whether the interface was pleasant to use, students had a similar distribution of opinions, slightly favoring positive. 12% strongly agree, 38% agree, 38% neutral and 12 % disagree on *Question D2*. 8% strongly agree, 50% agree, 34% neutral and 8% disagree on *Question D3*. In Question D4, we asked about whether they would recommend the interface to their classmates. 58% of the students strongly agreed or agreed. No negative opinions were found.

In the last category of the survey, *Privacy & Data Sharing* was investigated. Most of the respondents were inclined to be open in sharing the data. In *Question E1*, we asked whether the students like the idea of comparing their own progress to others. 67% of them strongly agree or strongly agree to it. In *Question E2, I feel comfortable sharing my progress with others*, 79% of the students strongly agreed or agreed. In *Question E3*, we found that 79% of the students did not mind sharing their average progress anonymously. Overall, we found no extremely negative opinions among all three questions.

Finally, in the free text comments, we found that some of the students expressed their appreciation. Some of them wished the tools were offered for other courses. Some even suggested the alignment of the content for exams or usage for participation or credits.

The survey collected the students' opinions and experiences on Progressor^{+}. They generally felt positively on all aspects, particularly the appreciation on *Ease of Use, Ease of Learning and Privacy & Data Sharing* categories. Additionally, students found that the content provided was valuable, given its being non-mandatory for the class. Despite the fact that there were various opinions on interfaces features, such as sorting and comparing, the overall attitude toward the system *Usefulness* was positive. This survey result confirms the design of the interface in terms of the content organization. Students' positive responses also compliment the objective system usage data.

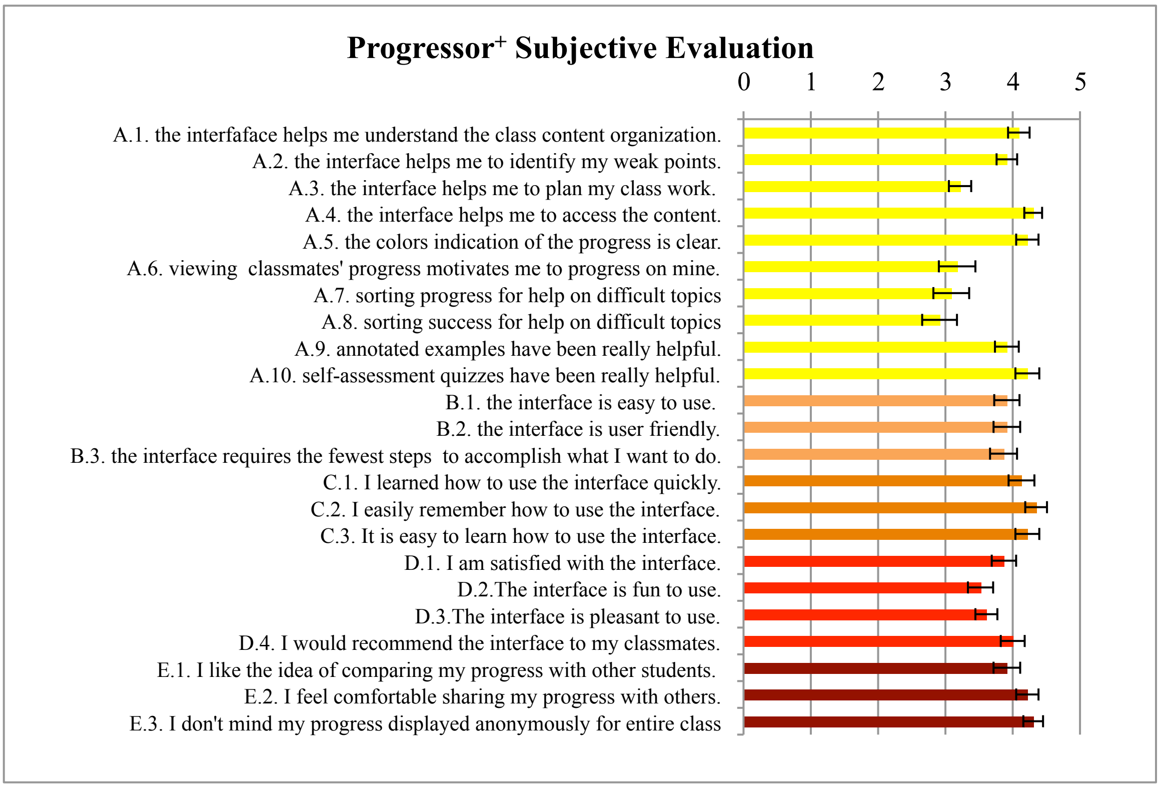


Figure 26. Summary of the subjective evaluation for each itemized survey question. Yellow color represents category A questions (Usefulness); Light orange represents category B (Ease of Use); Orange represents category C (Ease of Learning); Red represents category D (Satisfaction); Ruby represents category E (Privacy & Data Sharing).

5.0 CONCLUSION AND DISCUSSION

This dissertation explored the approach of open social student modeling visualization to help students with cross content navigation. We designed Progressor$^+$ and evaluated it in a real classroom. The results of the study helped us to obtain more insight on the influence of open social student modeling on students' learning. We first summarize the results in the following section and then discuss the limitation and contribution of the approach, and potential directions for future work.

5.1 SUMMARY OF THE RESULTS

Our main goal for this dissertation was to find a content navigation approach that merges the benefits of personalized learning and social learning and results in better learning for students. We proposed to integrate navigation support and social visualization approaches using an open social student modeling interface. We designed the framework for navigating rich content collections using this approach. To investigate students' motivation, learning and the navigation quality, we conducted a thorough evaluation in a classroom study.

In this subsection, we first present the result summary of this dissertation. We also revisit the research questions formulated in the first chapter. We then outline the contributions of this work.

5.1.1 Results summary

The classroom evaluation of our approach demonstrated that we achieved our main goal – helping students to navigate a rich collection of learning resources. Providing navigation support through open social student modeling visualizations helped students to locate the most relevant content and achieve a significantly higher *Success Rate*. In addition, incorporating a mixed collection of content in the open social student modeling visualizations effectively led the students to work at the right level of questions. Both strong and weak student worked with the appropriate levels of questions for their readiness, which yielded a consistent performance across all three levels of complexities. Additionally, providing more realistic content collection on the navigation supported open social student modeling visualizations results in uniform performance for the group. The classroom study revealed a clear pattern of social guidance, where the stronger students left traces for weaker ones to follow. This effect was much more noticeable, especially for the Complex questions.

The analysis of our approach confirms that students spent more time on the system, attempted more self-assessment quizzes, and explored more annotated examples. They achieved a higher diversity in attempting the self-assessment questions and exploring the annotated examples. Students were motivated to do more work. They were engaged with the system; they spent about 5 hours for each collection. Nevertheless, they successfully achieved better learning results. Students obtained significant higher knowledge gain comparing to no such support condition.

The subjective evaluation results showed that they generally felt positively about all aspects of the tool, particularly the appreciation on *Ease of Use*, *Ease of Learning* and *Privacy & Data Sharing* categories. Additionally, students found the content provided was valuable, given its being non-mandatory for the class. Despite the fact that there were various opinions on interfaces features, such as sorting and comparing, the overall attitude toward the system *Usefulness* was positive. These survey results confirm the design of the interface in terms of the content organization. The students' positive responses also complement the objective system usage data.

5.1.2 Revisiting the research questions

Question 1: What are the design principles (key features) to implement personalized guidance using open social student modeling visualizations?

In section 3.3, we concluded a set of features for personalized guidance using open social student modeling visualizations. They respectively are *Sequence, Identity, Interactivity, Comparison and Transparency*. Based on these design principles, we designed Progressor^{+} and evaluated it objectively and subjectively in a classroom study.

Question 2: Will the open social student modeling visualization provide successful personalized guidance within a rich collection of educational resources? I.e. Will this approach guide students to the right content at the right time?

Yes.

The approach of navigation support in open social student modeling visualization was illustrated and discussed in several projects during the PhD study. In Chapter 4, we evaluated the scalability of this approach for cross content navigation across a mixed collection of educational resources. The results showed that Progressor^{+} successfully guided students to the right level of questions based on their pre-knowledge. The personalized guidance helped students to work on the right level of problem at the right time. The social guidance helped students to explore more diverse problems. Moreover, introducing annotated examples to the open social student modeling visualization did not sacrifice the usage of self-assessment quizzes.

Question 3: Will the open social student modeling visualization approach increase students' motivation & engagement to work with non-mandatory educational content?

Yes.

A comprehensive answer to this question was given in Section 4.1. It provided a detailed evaluation of several parameters associated with motivation and engagement. The quantity of work, course coverage, and time spent were discussed. The amount of work, including the self-assessment quizzes and annotated examples, was dramatically increased. Students were observed to work on the topics ahead of the course schedule. They also revisited past topics voluntarily and heavily. Each student on average spent nearly 5 hours working on the quizzes and 5 hours and 20 minutes studying the annotated examples. This longer engagement paid off in the learning results and problem solving success.

Question 4: Will this approach improve students learning?

Yes.

In Section 4.3, we reported that the students' post-test scores were significantly higher than their pre-test scores. The approach of personalized guidance using open social student modeling visualization achieved the significantly higher normalized knowledge gain compared non-adaptive supported approach. In addition, we found a significant correlation between the time spent on the collections and the students' knowledge gain.

5.1.3 Contribution to the education field

The first contribution of this project is combining the ideas of adaptive navigational support and social visualization by using open social student modeling interface. The combined approach lowers the modeling complexity for knowledge-based personalization and increases the precision of social navigation support among the increasingly large and diverse number educational resources. This approach decreases the threshold for semantic-enriched online education. It also brings online education closer to the modern classroom. In addition, the approach has been proven to effectively guide students to the right content at the right time. It could be one of the pioneer works in open social student modeling realm.

Second, this dissertation summarized the design principles for personalized guidance using open social student modeling visualization based on a series of pre-studies.

Third, this dissertation established a scalable framework based on the design principles. The implementation, Progressor$^+$, was evaluated in the dissertation study. This framework allows extending the content collections to simulate a more realistic online learning environment. In addition to e-Learning, the classroom study also demonstrated that the tool can also be used as a complementary tool for real classrooms.

Forth, the underlying theories for adaptive navigational support and social visualization actually complement each other when brought together. According to learners' choices and beliefs about self-testing studies, students are generally overconfident about their memories and underestimate the amount they will learn by studying (Kornell & Son, 2009). The overconfidence of understanding is more severe among less advanced learners (Falchikov & Boud, 1989), who need most to be improved (Falchikov & Goldfinch, 2000). Therefore, this dissertation unveiled the social comparison mechanism by providing comparative interfaces and demonstrating the strong and weak students' performances in quantitative analyses with this approach.

5.2 DISCUSSION

This section concludes the dissertation with a discussion of some limitations of the proposed approach and an outline of possible directions for future research.

5.2.1 Limitations

There are some limitations of this project, discussed as below:

1. All of the systems were provided as supplemental tools for the same course. While Progressor$^+$ attempts to provide as realistic a scenario as possible by incorporating diverse learning objects for the learning environment, within the non-controlled classroom context students are still able to learn from the subject many different ways. (i.e. having hands-on experiences in coding plays a very important role in the programming language learning context. In our curriculum, students claim to benefit the most from the laboratory sessions.) The system used is just one of the factors that contributed to the learning. The content collections used in this dissertation work did not cover all the knowledge taught in the programming course. However, we took into account the semantics questions when measuring the students' learning.
2. Despite the face that the same curriculum was given across all semesters for the classroom studies, there were two instructors within four semesters. However, one of our main goals is to capture the longer-term engagement of our approach. Therefore,

we can only recognize the potential differences and design the study with primary and secondary baselines.

3. The first open social student modeling interface was introduced in the spring semester of 2010, where the latest system, Progressor^{+}, was introduced in the spring semester of 2012. Because social technology is rapidly evolving, students could potentially have been exposed to mass social media within these two years and gradually become more comfortable with using social tools. Our study is not able to capture this phenomenon.

5.2.2 Future work

Future research is planned in the two main directions:

1. Exploring the value of this approach in other domains and contexts;
2. Further improvement of the current implementation;

The conducted evaluation demonstrated a number of positive impacts of our approach – adaptive navigation support through open social student modeling. Moreover, based on students' subjective evaluation, they appreciated this non-mandatory tool, and would enjoy seeing it made available for other courses. Will this approach work in other subject domains? Will the same design support cross-domain navigation? Will other content collections work in current setting? Besides, the tool seems especially successful in serving as a complimentary tool for real classrooms. How does it work to include it in the curriculum? Does it work for online courses? All of these questions are motivating and can be answered by conducting more studies to expand the horizon of this approach. I strongly believe in continuing work in this direction for the sake of education and look forward to generalizing this approach for large scale personalized e-Learning.

Although the system was generally appreciated in the subjective evaluation, there were some questions on which students clearly indicated their concerns about the system.

For instance, they thought the system was pleasant to use but they held a reserved attitude towards the question “the system was fun to use”. We also acknowledge that there are several prominent aspects for improvement. There are two main important components that can be improved: the personalized guidance component and the interface component. On the personalized guidance level, an interesting approach to try is to include more knowledge-based components (i.e. learning analytics can be applied to indicate students’ knowledge). On the interface level, the social comparison could be aggregated and used to rank a list of recommended topics, hide unpopular topics from navigation, or add another layer of annotation based on the social feedback.

APPENDIX A

QUIZJET: JAVA EVALUATION TOOLKIT

QuizJET supports authoring, delivery, and evaluation of parameterized quizzes and questions for Java programming language. QuizJET can work in both assessment and self-assessment modes and covers a broad range of Java topics from Java language basics to advanced topics such as objects, classes, polymorphism, inheritance, and exceptions.

QuizJET Student Interface

A typical QuizJET question consists of a small Java program. One (or several) numeric value in the text of the program is instantiated with a random parameter when the question is delivered to a student. As a result, student can access the same question multiple times with different values of the parameter and different correct answers. To answer a question, students need to examine the program code and solve a follow-up task. The task can take one of two forms: “What will be the final value of an indicated variable?” or “What will be printed by the program to the standard output?”

A tabbed interface design has been implemented to allow questions consist of several classes. The driver class, containing the main function, is always presented on the first tab. It is the entry point to the question. The first tab also includes the question task and the field for student’s input. The system’s feedback is also presented in the first tab after a student’s answer has been evaluated. A QuizJET question example is presented in Figure 27. By clicking on different tabs students can switch between the classes to access the full code of the program.

Once a student enters an answer and clicks the “Submit” button, QuizJET reports the evaluation results and the correct answer (Figure 28). Whether the result were correct or not, the student can click the “Try Again” button to assess the same question with a different value of the generated parameters. This option provides students with an opportunity to master a particular topic.

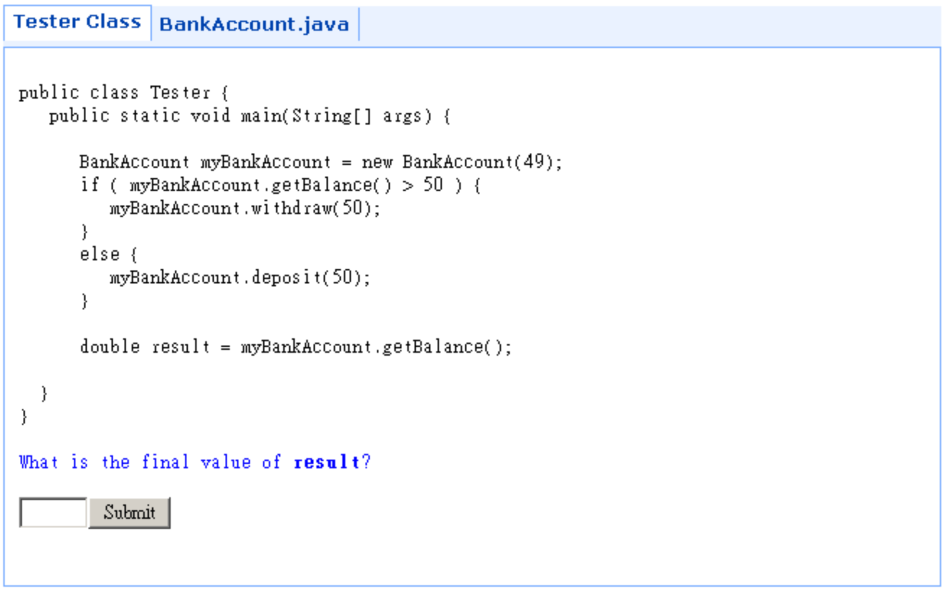


Figure 27. The presentation of a QuizJET question

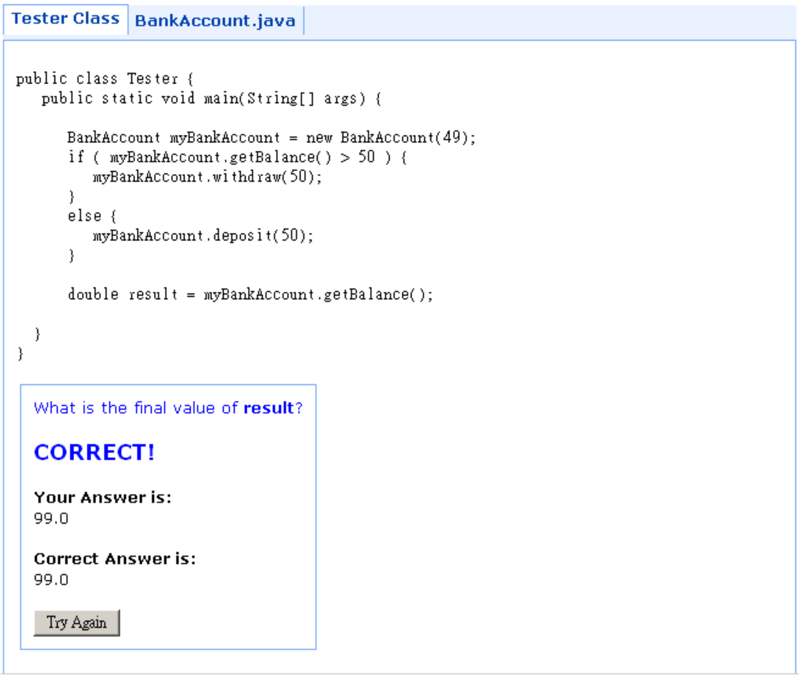


Figure 28. The evaluation results of a QuizJET question

QuizJET Architecture

QuizJET has been developed as a component of ADAPT2 architecture for distributed adaptation and user modeling[6]. It complies with the ADAPT2 protocols for user authentication, reporting user interaction, and adaptation. URLs of QuizJET questions can be augmented with ADAPT2 HTTP parameters to notify the system about the current user, group, and session. Upon verifying student answers QuizJET also generates a learning event transaction, which contains information about the user, the question, the result of the interaction, etc. The transaction is sent to the user modeling server CUMULATE that computes student knowledge and reports it to the interested systems (Brusilovsky, Sosnovsky, & Shcherbinina, 2005). This architecture enables easy integration of QuizJET with value-added adaptation services.

[6] Description of ADAPT^2 can be found at: http://adapt2.sis.pitt.edu/wiki/ADAPT2

Each QuizJET question is accessible by a unique URL. Once a question is launched, QuizJET server generates a question and delivers it to a student's browser. When the student submits a solution, QuizJET executes the question code to produce the right answer, compares it to the user's input and presents a feedback.

QuizJET Question Authoring

QuizJET offers a form-based online authoring interface for developing new quizzes and questions. Figure 10 demonstrates the process of QuizJET question authoring. The question template form requires an author to specify several question parameters. An author has to provide the Title for the question template and specify which Quiz it belongs to. The rdfID is a unique attribute to reference the question template. A short comment about the question template can be given under the Description field. The Assessment Type dropdown box is the attribute, which specifies the task of the question. Currently, there are two forms of the task available: evaluation of the final value of a variable and prediction of what will be printed to the standard output. The body of the question template should be provided in the Code field. In the code, the _Param variable indicates where the randomized parameter will be substituted. Maximum and Minimum specify the interval for the parameter generation. Answer Type dropdown box provides a list of data types for the final value. Privacy indicates the availability of the question to QuizJET users. Currently QuizJET includes 101 question templates grouped into 21 quizzes. Authors are allowed to upload supplemental classes to include in their questions. Every supplemental class is reusable and is listed on the right hand side of the authoring interface (Figure 29).

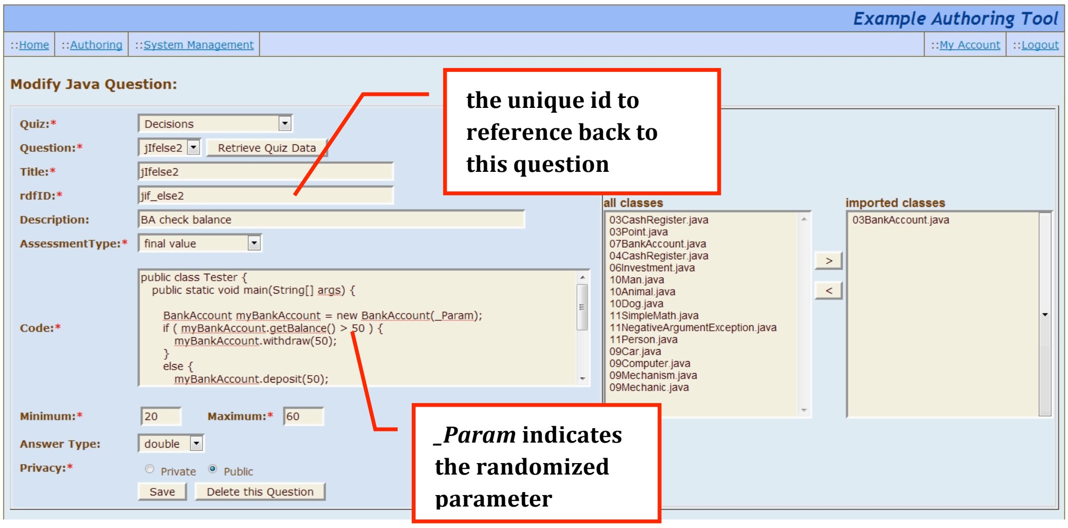


Figure 29. A fully authored QuizJET parameterized question

APPENDIX B

JAVAGUIDE: ADAPTIVE NAVIGATION SUPPORT FOR QUIZJET QUESTIONS

In order to motivate students to work with the questions, we attempted to use an adaptive technology approach, in order to promote user participation and guide users to appropriate quizzes and questions. JavaGuide was implemented to provide adaptive navigation support for QuizJET questions. It inherits its infrastructure from QuizGuide (Brusilovsky, Sosnovsky, S., and Shcherbinina, O., 2004), which has been successfully used in a number of C-programming courses (Brusilovsky, Sosnovsky, S., and Shcherbinina, O., 2004; Brusilovsky, Sosnovsky, S., and Yudelson, M., 2004).

JavaGuide: adaptive navigation support for QuizJET questions

The development of QuizJET along with its authoring system, allowed us to create a sufficient volume of questions, which was vital for further experiments with personalized guidance. Our next step was to develop JavaGuide, the system that provides students with personalized guidance to QuizJET questions. The questions in JavaGuide are combined under large topics (from three to six questions per topic) that organize the course material into instructionally complete chunks. Students can browse the material by clicking on topic and question links (Figure 30). A click on a topic link folds/unfolds questions available for the topic. This allows students to organize their learning space more flexibly. A click on a question link loads the corresponding question in the question frame of the system's interface. On both levels – topics and questions – the system offers personalized guidance using adaptive link annotation, one of the most popular adaptive navigation support techniques.

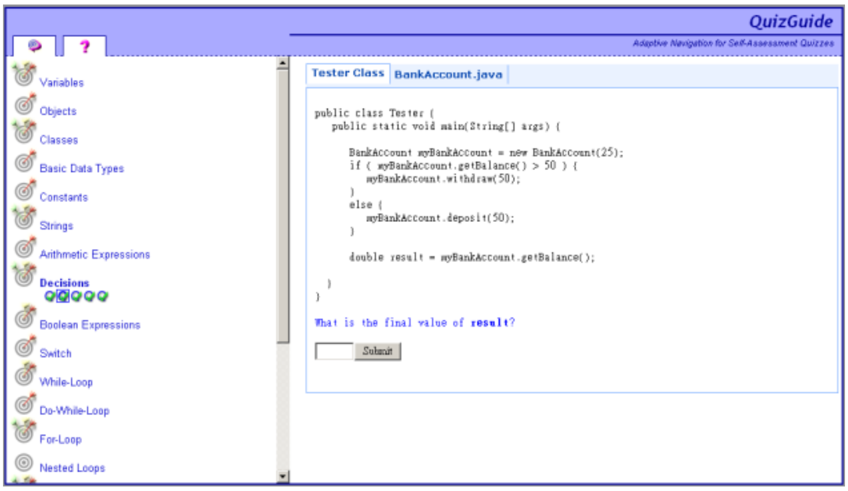


Figure 30. JavaGuide Interface

On the topic level, JavaGuide uses a specific form of adaptive link annotation inspired by the ideas of open learner modeling: it presents to a student the content of her/his user model in the form of navigational cues. Every topic link annotation represents the current state of a student's knowledge for the topic. As a result, a student is constantly aware of his/her performance and is able to focus on those parts of the course, in which he/she has not demonstrated enough progress.

Topic-level adaptive annotations are visible to students as "target-arrow" icons (Figure. 31). The icons deliver two kinds of information to the student: the individual performance of the student with the topic's content and the relevance of the topic to the current learning goal of the entire course. The number of arrows (from 0 to 3) in the target reflects the progress demonstrated for the topic. Once the student has solved enough questions correctly, the topic will be annotated with the "3-arrows target", which indicates the highest level of mastery and tells the student that he/she should focus on different topics. If no or very little progress has been made on the topic, the target icon for this topic will be empty, which invites the student to concentrate on this topic more.

The color of the topic icon designates the relevance of the topic to the current learning goal (Figure. 31). As new topics are introduced by the teacher of the course, JavaGuide annotates them with bright-blue icons representing the current learning goal of the students. Topics that have been introduced earlier in the course are no longer relevant to the current goal. JavaGuide indicates so by annotating them with grey icons. If a student has problems with any of the past topics that need to be mastered in order to understand the current learning goal, he/she most probably will have problems with the current topics as well. To support students in resolving such problems, JavaGuide annotates topics that are prerequisites for any of the current learning goals, with pale-blue target icons. Finally, all the topics that have not been introduced in the course yet, are annotated with crossed-out target icons; this means the student is not ready for them yet.

Figure 31. Upper row: the level of relevance to the current learning goal (current goal, prerequisite for the current goal, passed goal, future goal); lower row: levels of knowledge for the topic.

Thus, the topic annotations in JavaGuide combine two kinds of adaptation: individual progress-based adaptation and group-wise time-based adaptation. JavaGuide does not restrict the access to the learning content in any way. The students can access any topics, even those that have not been introduced yet. JavaGuide merely informs the students about the individual and group-wise importance of the topics and tries to direct students to the best learning content at any particular moment of time.

To help the student understand the meaning of all elements of the interface, JavaGuide dynamically generates mouse-over hints for the icons. A detailed help explaining all interface elements is available as well.

To further assist students in navigating through the corpus of available learning content, JavaGuide also supports adaptive annotation for individual questions. Question icons of JavaGuide report to students the completion status of questions. The completion status of a question is a binary entity. It reflects whether the specific question has been solved correctly at least once. As soon as a student submits his/her first correct answer to a question, the corresponding icon receives a checkmark. This can help students to choose between similar questions characterized within a topic. If one of the questions has a checkmark, and another does not, a student who is still interested in testing her/his knowledge of this topic will be guided to the unsolved question.

Content Integration Technology

In order to bridge the content between QuizJET and JavaGuide, each QuizJET question had to be associated to a group of concepts and be assigned to the topics in JavaGuide. The approach we adopted here was to build a Java-ontology-based Parser to generate representative concepts. The Java ontology implemented in this parser was developed in the TALER Lab at the School of Information Sciences, University of Pittsburgh. It can be accessed at http://www.sis.pitt.edu/~paws/ont/java.owl. It was designed in Protégé 3.3.1 as an OWL-Full ontology (Figure 32). There are more than 300 classes connected via three relations: standard rfs: subClassOf, partOf-hasPart and relatedTo.

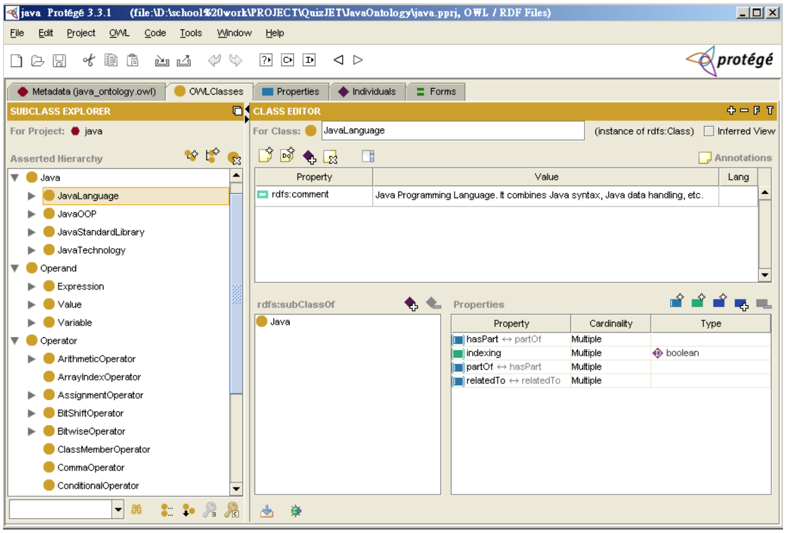


Figure 32. Java Ontology

The Java Parser was implemented with JFlex (LEXical analyzer generator for Java) (Hudson) and CUP (Constructor of Useful Parsers) (Hudson, 1999). The Java Parser can not only analyze the program syntax but also the program complexity. There are 42 associated rules and 127 distinct concepts that can be recognized, which contributes to a 73.41% parsing rate. Among the current collection of QuizJET quizzes, there are 41 easy ones, 41 moderate ones and 19 hard ones.

APPENDIX C

ANNOTEX: COMMUNITY-BASED PRODUCTION OF ANNOTATED EXAMPLES WITH PEER-REVIEW PROCESS

AnnotEx, Example Annotator System, was developed to support community-based authoring of annotated programming examples. It allows a community of students (for example, a class) to author annotations to examples, as well as to provide comments and ratings on the annotations produced by their peers. Each member from the community has three tasks to complete in the example annotating process. The first task is to author the annotation of the example. The second task is to provide ratings/comments about the example annotations. The third task is to re-annotate, ie, to edit and expand the original annotations. AnnotEx is a Web-based system which can be accessed anywhere with a web browser and an Internet connection.

The AnnotEx interface (Figure 33) divides the screen into two sections. The upper section represents student tasks; the lower section illustrates the example pool of the community. The tasks are sequentially arranged from left to right, based on the process flow, annotating, rating/commenting and re-annotating, respectively. Upon the completion of each task, she/he can continue on to the next task. The example pool of the community is available at all times, regardless of which task s/he is doing. AnnotEx is enhanced by an evaluation prototype. A five-star rating mechanism has been adopted to indicate the quality of the evaluation. Ratings are collected from the second task. The average ratings of the example from the community will be shown on the main page.

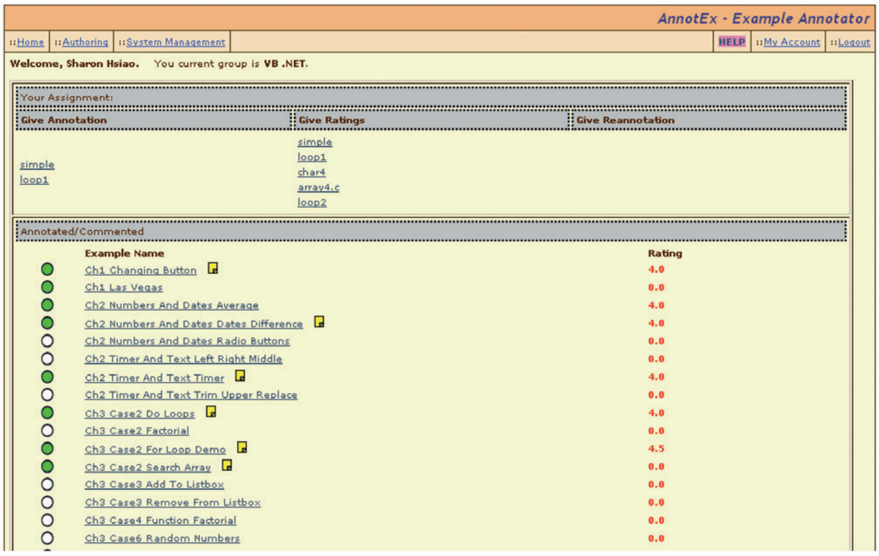


Figure 33. The main page of a community on AnnotEx

In Figure 30, the main page of a community on AnnotEx, the green circles mark which examples are annotated, while white ones are not annotated. Yellow post-it icons show comments on the annotations. Ratings are shown at the right. Figure 34 (left) presents the first task, an annotation task. The interface is divided into left and right. The left side displays the example code, line-by-line. The right side is the place for students to write their own corresponding annotations, line-by-line. Students can also click on the button at the top to copy the program code. Figure 34 (right) is the interface of the second task, rating and commenting. The top of the screen is the area to provide ratings. The main body consists of three parts: (1) on the left, the example code again appears in black; (Lindstaedt, et al.) blue letters in the middle are annotations, corresponding line-by-line to the example code; and (3) on the far right, students provide comments, line-by-line. The third task, re-annotating, has the same interface as the first task. A detail peer review process is explained in (Hsiao & Brusilovsky, 2011).

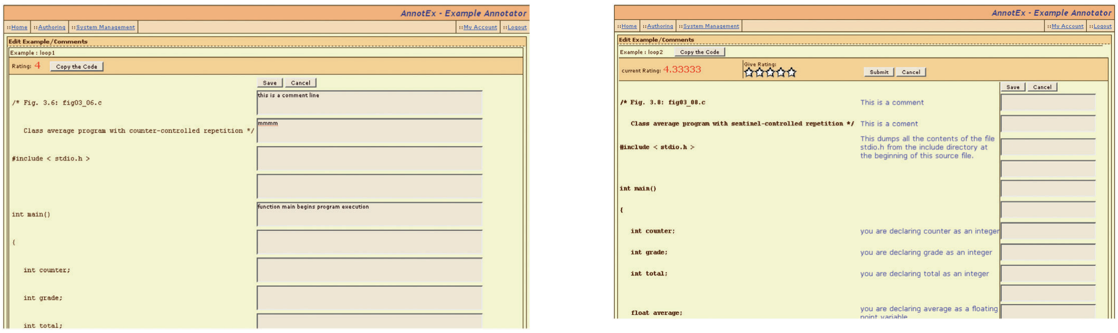


Figure 34. The interfaces for authoring and peer reviewing example annotations on AnnotEx.

AnnotEx was evaluated in several studies and reported encouraging results (Hsiao & Brusilovsky, 2008, 2011). The authoring system supports the mass production of example annotations and enhances student learning. The study results demonstrated that community was capable of authoring and peer reviewing which resulted in positive impacts on both the quality and the quantity of produced content. Moreover, the authoring process was highly appreciated by the students; at least 80% of them stated that both reading peer annotations and authoring their own annotations helped them to understand the subject they were working on.

APPENDIX D

WEBEX SYSTEM

WebEx provided Web-based interactive access to examples enhanced with line-by-line comments, allowing students to browse the comments at their own pace and chosen sequence (Brusilovsky, 2006) (Figure 35).

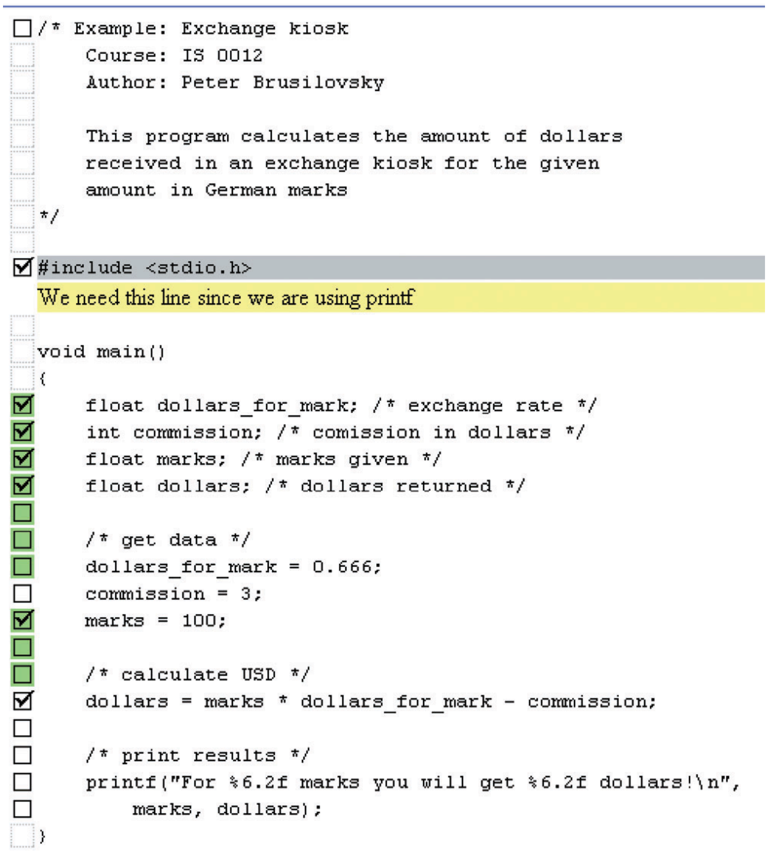


Figure 35. WebEx system.

APPENDIX E

NAVEX: ADAPTIVE NAVIGATION SUPPORT FOR ANNOTATED EXAMPLES

NavEx combines zone-based and progress-based of adaptive annotation techniques and provides adaptive navigation support for annotated examples (Yudelson & Brusilovsky, 2005). The interface of NavEx is shown in Figure 36. Once students logged on the system, they could see the list of all examples annotated with adaptive visual cues. Students use this list (shown in the left frame) to select examples, which are immediately loaded in the main frame for exploration. The main (right) frame has a name of currently viewed example and the code example itself.

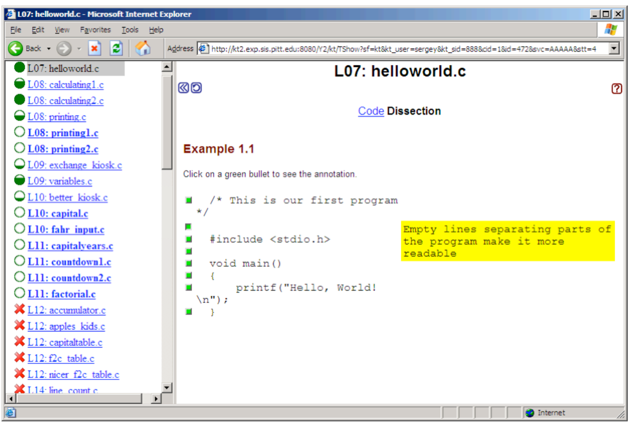


Figure 36. Interface of NavEx

NavEx provides the adaptive guidance by showing personalized annotations next to example links. Depending on student’s progress, NavEx shows fillable circles with green colors to indicate whether the examples are “sufficiently known” or “ready to be learned”.

The progress measures range from 0% to 100% with 25% increments (Figure 34). A red X icon (Figure 37) means that the example is not ready to be learned, but the student is free to explore it (Yudelson & Brusilovsky, 2005). Adaptive navigation cues provided by NavEx increased students' work in general. Students were exploring material more often and were covering it in a greater width, returning to examples from previous lectures regularly. In the subjective evaluation, students showed great appreciation of the value of navigation support.

Not ready to be explored	Ready to be explored

Figure 37. Annotation cues for examples in NavEx

APPENDIX F

QUIZMAP: ADAPTIVE NAVIGATION SUPPORT OF PARAMETERIZED QUESTIONS WITH TREEMAP

A TreeMap is a space-filling visualization method for representing hierarchical information (Shneiderman, 2004). By dividing the display area into a nested sequence of rectangles whose areas are associated to attributes of the data set, it effectively illustrates the structural information with slices and dices. TreeMaps have been applied to a wide variety of domains ranging from financial analysis (Wattenberg, 1999), petroleum engineering (C. Plaisant, 2003) to network security analysis (Mansmann, Fischer, Keim, & North, 2009). Some studies have focused on specialized techniques to visualize large number items on a TreeMap without aggregation (Fekete & Plaisant, 2002). The innovative idea to use TreeMaps to visualize a model of individual learner knowledge was first suggested in (Lindstaedt, et al., 2009).

QuizMap is a TreeMap representing the work of a user group (such as a class) with self-assessment questions. We customized the TreeMap by using the size and color of the rectangles to display the performance of the student. To adapt the TreeMap approach to the context of self-assessment questions, we structured system's TreeMap into 4 levels. Each level of the TreeMap clusters different level of information in detail. The top level consists of 1 root node, which represents the summary information of the entire class, including the overall attempts, successful rate and average statistics. The second level consists of 21 nodes corresponding to topics covered within the class. Under each topic node, next level is formed by the parameterized self-assessment questions belonged to the topic. The bottom level of the TreeMap shows performance of each individual student in a group for each question. The QuizMap structure is presented in Figure 38.

The sizes of the rectangle for each node represent the amount of work done. The color indicates the amount of knowledge gained (credited with each successful answer). The student's own performance is colored in orange and to contrast with the rest of the class, colored in blue. The darker the color, the higher success it presents and vice versa. Both reddish yellow and bluish color tints can be decomposed into 10 different "shades" (Figure 39). All the absolute values of the performance are displayed when user hover over the rectangle. These two different color schemes are meant to make it easier for the student to compare his or her performance with the performance of individual peers and the whole class.

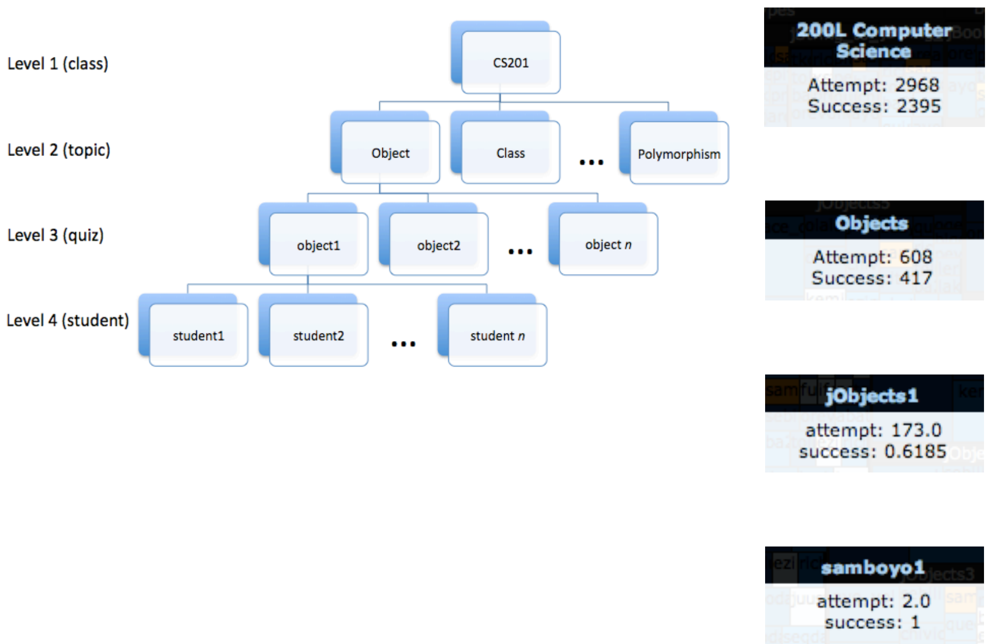


Figure 38. QuizMap structure.

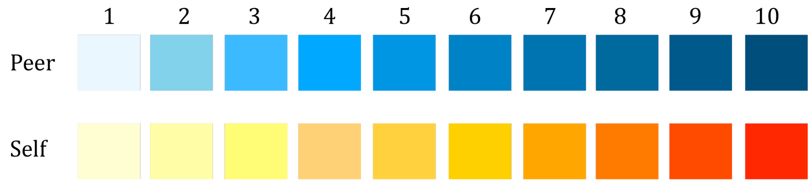
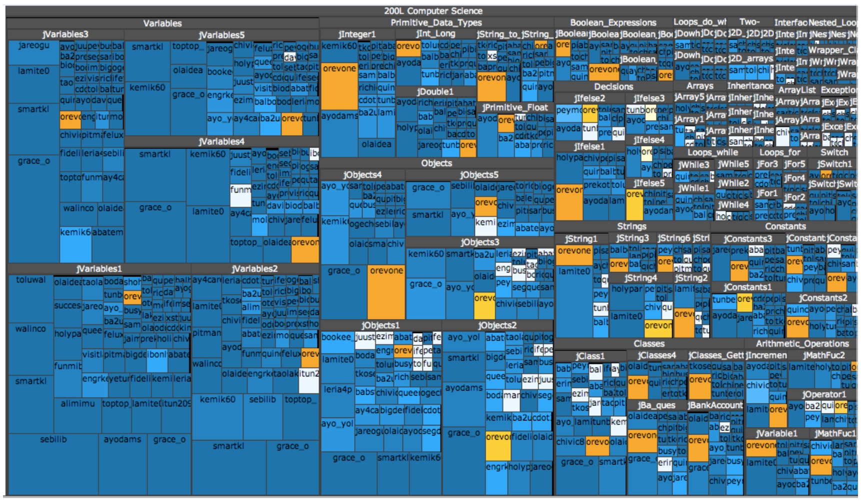


Figure 39. QuizMap rectangle color shades indication.

To illustrate the use of the TreeMap in the context of self-assessment quizzes, Figure 19 represents an overview of QuizMap. To answer a quiz, a student has to select the question from each topic in the QuizMap. Upon the selection, QuizMap will pop a separate window to display the question (Figure 28). Each question asks the student to predict the results of execution of a specific Java program (i.e., mentally execute theprogram and enter the final value of some variable of the text to be printed by the program.) All questions are parameterized, i.e., include a random parameter, which the system instantiates when the question is delivered to a student. As a result, the student can attempt to answer the same question multiple times with different values of the parameter, which helps to achieve the mastery level. The implementation and functionalities of parameterized self-assessment quizzes were described in detail in (Hsiao, et al., 2010).



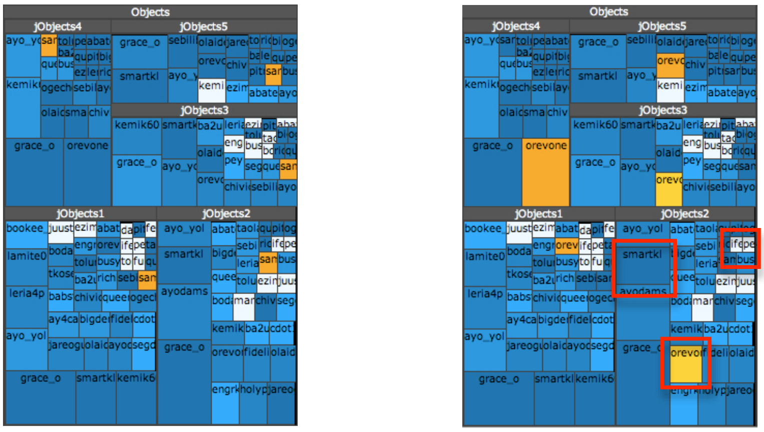


Figure 40. An overview of QuizMap (Olston & Chi); A zoom in view on topic Objects of two students, student A (bottom-left) & student B (bottom-right)

The bottom part of Figure 40 shows two zoom-in views of the same topic, Objects, for two students. It demonstrated, that the amount of work done by the student A (Figure 37 bottom left) was relatively the same amount of questions on this topic. The color indicates a roughly 70% successful rate across all questions that s/he attempted. It suggested that this student had been consistent on performing different complexity levels of questions. Such way of evaluation can also be found throughout the class on his/her model. However, the other zoom-in view of QuizMap by student B (Figure 40 bottom right) displayed a different scenario. The student focused on working on certain questions, especially the *jObject4* question, which reached relatively high attempts. Throughout the class, he also followed the similar pattern of work. He had more attempts on a particular set of questions repeatedly and achieved the 50~70% successful rate. It suggested that this student might have troubles in those topics. Therefore, he kept trying again and again on the same questions to improve himself. In this open social student model TreeMap, students are expected to identify the strengths and weaknesses of themselves and their peers. For example, in the example of lower QuizMap, the student was struggling (low successful rate) with the question *jObject2* under the topic *Objects*. QuizMap provides opportunities for him to discover stronger peers by recognizing dark blue rectangles and vice versa. This student should also realize that who have less success on this specific question by recognizing the lighter blue rectangles. Those students may have lower chances to help him achieve a better understanding in this question.

F.1. Study setup and data collection

We conducted a classroom study in the Programming and Algorithms course offered by University of Ibadan. The students were second year Computer Science majors. There are 86 students in the class – 52 male and 34 female. Out of them, 77 students were taking the course first time while 9 were repeating the course. The essence of the course is to build on the foundation they already have and teach Algorithm concept using Java and C++, thus enabling them build complete working program from the algorithm. Lectures were conducted through face-to-face interaction with the students. Assignments were submitted online by email attachment. Students already had introductory knowledge of Java in the first semester. Therefore, the QuizMap was introduced to the class as a supplemental tool. Students were encouraged to use QuizMap after being acknowledged that QuizMap quizzes will appear in the exam up to 10% of the marks. A major problem encountered by the students during the semester was the internet access issue. Access to internet in the school lab was only available for very limited hours which did not fit properly into the students' schedule most of the time. Sometimes, electricity was also a problem. As such, students could not use the computers in the laboratories at those times.

Table 13. Summary of the overall usage on QuizJET and QuizMap

		QuizJET	QuizMap
All	Users	16	65
	Total Attempts	1293	2961
Average	Attempts per user	80.81 ± 22.06	45.55 ± 6.67
	Success rate	42.63% ± 1.99%	79.30% ± 1.94%
	Distinct Topics	7.81 ± 1.64	4.55 ± 0.59
	Distinct Questions	33.37 ± 6.50	17.07 ± 2.78
	Sessions	3.75 ± 0.53	4.29 ± 0.54
	Pre-test Scores	9.56 ± 1.29	7.55 ± 0.49
	Post-test Scores	17.12 ± 0.86	13.25 ± 0.60

F.2. Evaluation results

We analyzed the log data on students' interaction with the social visualization on the self-assessment quizzes (QuizMap) and compared the usage with the data from a comparable Object-Oriented Programming class at the University if Pittsburgh where students accessed the self-assessment quizzes using a traditional course portal with no visualization (QuizJET). Table 1 shows the basic statistics on both systems. There were 65 students who used the QuizMap. They made 2961 attempts to the questions, on average 45.55 questions per student. Students achieved 79.30% on average successful rate on answering the self-assessment questions. On average, students tried 4.55 distinct topics, 17.07 distinct questions and had 4.29 visits on the QuizMap. Comparing to QuizJET, the students who worked with QuizMap made less attempts and explored fewer topics (it could be related to the computer and internet access problems). Despite that, they achieved almost a much higher success rate. This level of success rate is typical for question access mediated by adaptive navigation support

(Brusilovsky & Sosnovsky, 2005a; Brusilovsky, et al., 2009; Hsiao, et al., 2010). This provides some evidence that social navigation support is comparable to classic adaptive navigation support by its effectiveness. To obtain more reliable evidence a study should be repeated with more comparable groups.

What is the mechanism of social guidance? How this approach based on the "collective wisdom" of a student community can guide students to the right questions as successfully as classic knowledge-based guidance? In our past work, we found evidence that in social guidance systems stronger students with better understanding of the subject lead the way discovering most relevant resources and creating guidance trails for weaker students. In order to investigate the social guidance effect in QuizMap, we categorized students into two groups, strong and weak, based on their pre-test scores (ranging from a minimum of 0 to a maximum of 20). Strong students scored 10 or higher points in the pre-test, and weak students scored less than 10 points. In Figure 41, we plotted all attempts over the course period. X-axis denoted as course period; Y-axis denoted as the topic complexities sorted from easy to complex. Blue data points represent strong students and orange points are the weak ones. We found that both strong and weak students started simultaneously on the easy topics. However, over time stronger students tended to run ahead of weaker ones. Weaker students approached new topics after the stronger ones had already explored it. Such behavior is more noticeable for more complex topics. The pattern indicated that stronger students, indeed, guided the weaker ones to the proper questions. This allowed weaker students to achieve success rate and post-test scores that are close to those of stronger students. At the end of the course, they narrowed the knowledge gap and achieved higher learning gain that stronger ones (Table 14).

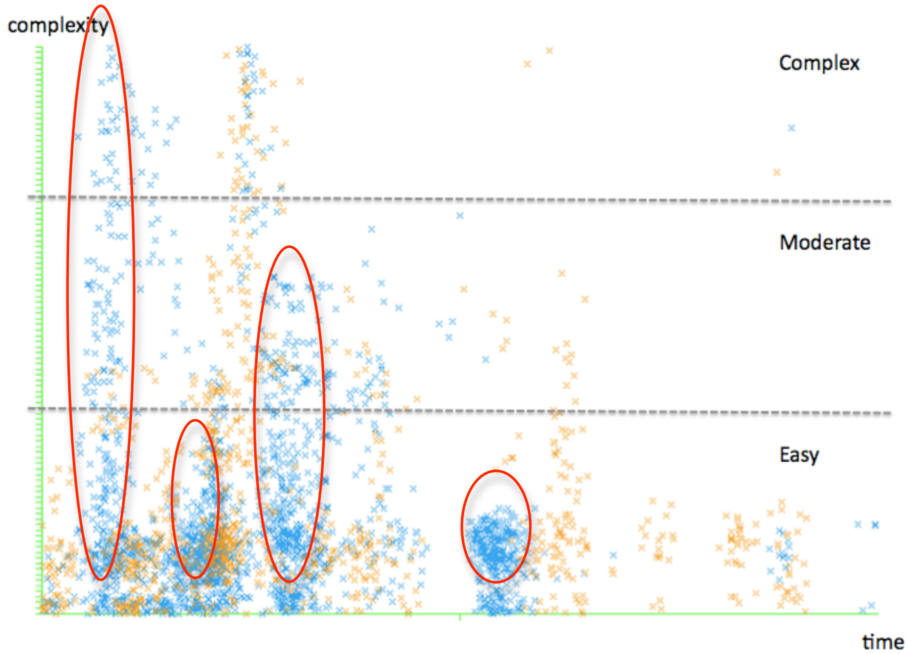


Figure 41. Strong students guided weak students to explore the topics overtime. Blue data point represents strong students' attempt; orange data points represents the weak ones.

Table 14. QuizMap usage by strong/weak student

Parameters	**Weak (n=29)**	**Strong (n=22)**
Attempts	33.17 ± 6.89	54.18 ± 13.40
Success Rate	77.91% ± 3.30%	83.29% ± 2.70%
Distinct Topics	3.93 ± 0.83	5.18 ± 1.06
Distinct Questions	13.37 ± 3.64	20.23 ± 4.99
Average Sessions	3.52 ± 0.51	4.00 ± 0.67
Learning Gain (post-pre)	7.55 ± 0.89	3.22 ± 1.12
Pre score	4.86 ± 0.53	11.1 ± 0.35
Post score	12.41 ± 0.96	14.32 ± 0.98

F.3. Summary and Discussions

QuizMap is a novel approach to integrate social navigation for self-assessment questions with open user model in a TreeMap interface. The hierarchical representation of TreeMap was implemented to help students visualize both, the state of their knowledge and the progress of the whole class. Color contrasts between personal progress and the progress of others students were used to provide social guidance. The classroom demonstrated that QuizMap visualization provided effective social guidance allowing students to achieve high quality of learning. The effect was comparable with the impact of traditional knowledge-based guidance. The potential key to the success of the social guidance is the trailblazing behavior of stronger students who explored the topics and left the trace for weaker students to follow. In general, student satisfaction with QuizMap was high. However, there is also a piece of evidence that the QuizMap approach may not be optimal for larger classes that generate too many cells on the TreeMap, causing it to become too crowded.

APPENDIX G

PARALLEL INTROSPECTIVEVIEWS

The integration of QuizJET with IntrospectiveViews is intended to provide students with a holistic and easy-to-grasp view on their progress and relate it to the progress of other students in the class. We called it *Parallel IntrospectiveViews*. The goal of the system is not only to help students in accessing the right learning content at the right time, but also to motivate them to progress and perform better. It merges two systems, namely the QuizJET system (Hsiao, et al., 2010) for the authoring and delivery of parameterized questions for the Java programming language and the IntrospectiveViews interface (Bakalov, König-Ries, Nauerz, & Welsch, 2010a, 2010b) for visualization of semantic user models. Parallel IntrospectiveViews offers visualization of student progress on QuizJET questions of an Object-Oriented Programming course. The visualization consists of two panes: the left pane displays the student's own model, whereas the right one displays someone else's model. By default, the right pane displays the average progress of the entire class, but the student can switch to the progress of a specific classmate by selecting the classmate's name from the combobox menu located at the top of the right pane. Each pane visualizes the student progress in the form of a pie chart consisting of circular sectors representing the class lectures. The lectures are displayed in a clockwise order denoting their pre-requisite sequence, i.e., the order they are taught in the class. Lectures may consist of one or several topics, which are represented as annular sectors placed within the circular sector of the corresponding lecture. For example, in Figure. 42, Basic Concepts is the first lecture in the Introduction to Object-Oriented Programming class and consists of three topics, namely, Variables, Objects, and Classes. The radius (width) of annular sectors denotes the amount of readings, quizzes, and exercises assigned to the topic. In a similar way, the span of circular sectors indicates the amount of learning content assigned to the corresponding lecture. Such a representation allows the student to easily estimate the amount of work she has to spend on each individual topic or lecture.

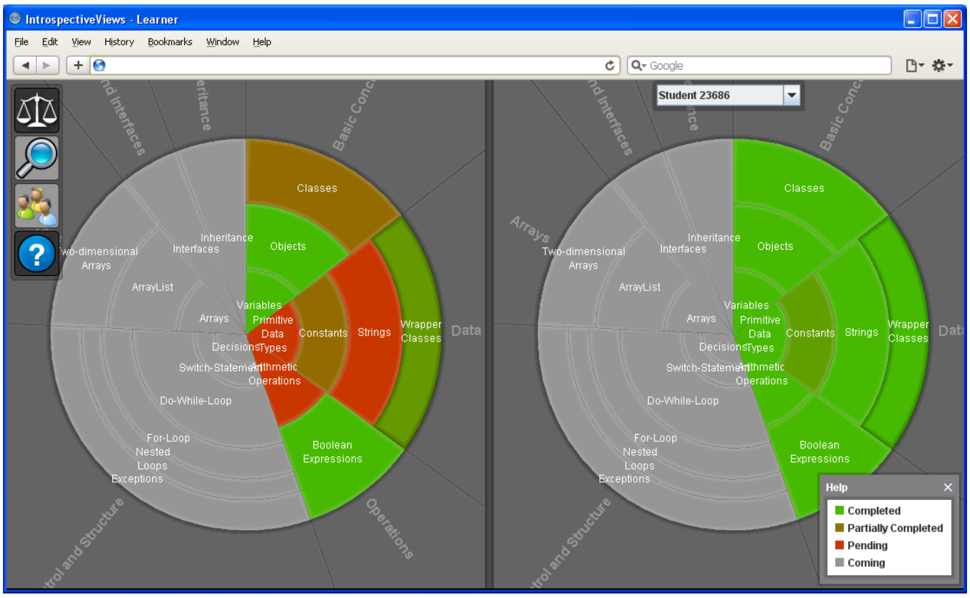


Figure 42. Parallel IntrospectiveViews. Left pane – visualization of the student's own progress; right pane – visualization of a peer's progress. The circular sectors represent the lectures and the annular sectors represent the topics of individual lectures. The shades of the sectors indicate whether the topic has been covered and for the covered ones, denote the progress the student has made. Color screenshots available at: http://www.minerva-portals.de/research/introspective-views/.

The shade of each annular sector denotes whether the topic has been covered and, for the covered ones, indicates the progress the student has made with respect to the topic. The sectors painted grey represent the topics that have not been covered yet, whereas the sectors painted a shade from the color range red to green represent the sectors that have been already covered. For the covered topics, the interface displays the student progress. The progress, in the current implementation, is the ratio of successfully completed quizzes to the total quiz count in the topic. If the ratio equals 0, i.e., no quiz has been successfully completed, the sector is painted red. If it equals 1, i.e., all quizzes have been completed, the sector appears green. The shades in the range between red and green denote partial completion of the quizzes.

Similar to the original design of IntrospectiveViews (Bakalov, et al., 2010a, 2010b), the current implementation supports a number of information visualization tasks postulated by Bed Shneiderman (Shneiderman, 1996). It allows getting an overview of the entire model, but it also allows zooming into a certain part of it in order to get a better view, which is especially important for visualizing models that consist of a large number of topics. Also, it provides details on demand, e.g., by clicking a sector, the interface will display the contents of the corresponding topic, in the current implementation, the list of quizzes for the topic Figure. 43. For each quiz, the interface provides a visual cue indicating the student progress and displays the total number of attempts the student has made on the quiz and the number of successful attempts. By clicking a quiz label the interface will display the quiz in a new window. In that way, having found uncompleted quizzes, the student can quickly open each of them and complete the pending task.

Such visualization can help the student to plan her class work by providing an overview of her progress in the class and showing the topics that she has already completed and the ones that she has to work on. In addition to that, the ability to view someone else's progress allows the student to quickly find the peers that can help with a difficult topic or quiz. For example, if the student experiences difficulties in completing some quizzes, using the parallel views, she can find a classmate who has already successfully completed those quizzes and ask for help. Finally, the ability to view the average progress of the entire class allows the student to relate her progress to the one of the whole class and estimate whether she is ahead or behind of the class on average.

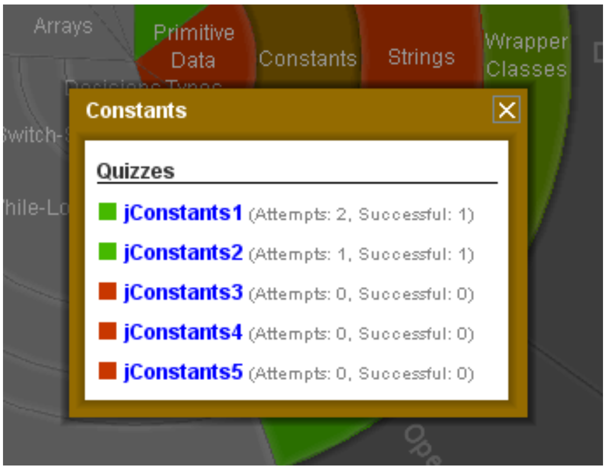


Figure 43. Parallel IntrospectiveViews. Quizzes of the selected topic.

G.1. Study setup and data collection

To assess the impact of QuizJET with Parallel IntrospectiveViews, we have conducted a thorough evaluation in a semester-long classroom study. The study was performed in an undergraduate Object-Oriented Programming course offered by the School of Information Sciences, University of Pittsburgh in the Fall semester of 2010. All students received access to self-assessment quizzes through the IntrospectiveViews (IV) interface. The system was introduced to the class at the beginning of the course and served as a non-mandatory course tool over the entire semester. Of the 32 students enrolled in the course, 18 actively used the system. All student activity with the system was recorded. For every student attempt to answer a question, the system stored a timestamp, the user's name, the question, quiz, and session ids, and the results (right or wrong). We also recorded the frequency and timing of student model access and comparisons. Pre- and post- tests were administered at the beginning and the end of the semester in order to measure the gain in students' learning. At the end of the semester, the students were asked to provide their subjective feedback about the system and its features by completing the evaluation questionnaire.

G.2. Evaluation results

We found that the social visualization of student models with IntrospectiveViews resulted in a 39% increase in the average attempts compared to the traditional course portal. The students also explored more topics, tried more distinct questions, and accessed the system more frequently. In brief, we observed an increase in all usage parameters similar to that it was observed in a very different JavaGuide interface. At the same time, the increase in usage was not as high as in the case of JavaGuide. As a result, no significant difference on the usage level was found between IV and the portal as well as between IV and JavaGuide.

Since the student own knowledge visualization was relatively similar in IV and JavaGuide, a slighter increase in student activity in IV could be attributed to the social side of open social student modeling. While the access to social data could encourage less active users to do more work, it can also discourage very active users from jumping too much ahead of the class. As a result, the difference between the most active and least active users is getting smaller. Evidence that this is really happening is the observed 25% decrease in standard deviations for the number of attempts. In turn, the class as a whole became a bit less adventurous than in non-social JavaGuide, exploring fewer questions and topics (this is because the variety of topics come to some extent from more active users who run ahead of the class). This effect can be also observed in IV, especially the session level. While the amount of work per session increases for IV, question and topic coverage stays the same.

In sum, as a whole, social guidance provided by the access to class progress mediates the motivating effect of progress visualization by making the whole class a bit less adventurous and more conservative than without social guidance tools. An interesting question is whether a more conservative increase in the amount of work and variety of explored context is a good or a bad thing. Our evidence shows that it might actually be a good thing. As Table 15 shows, students using social visualization in IV achieved the highest success rate (a ratio of correct solutions to total attempts) among all conditions. This is significantly higher than for the portal case, $F(1,32)= 11.303$, $p<.01$. The growth of the success rate demonstrates that knowledge-based and social guidance combined are more effective in guiding the students to appropriate questions that they are ready to handle than knowledge-based guidance alone. The community wisdom does matter.

Table 15. Summary of Basic Statistics of System Usage

		1	**2**	**3**
		QuizJET w/ IV	**QuizJET w/ Portal**	**JavGuide**
	Parameters	**n=18**	**n=16**	**n=22**
Average User Statistics	Attempts	113.05 ± 15.17	80.81 ± 22.06	125.50 ± 20.04
	Success Rate	71.35% ± 3.39%	42.63% ± 1.99%	58.31% ± 7.92%
	Distinct Topics	9.06 ± 1.39	7.81 ± 1.64	11.77 ± 1.19
	Distinct Questions	36.5 ± 5.69	33.37 ± 6.50	46.18 ± 5.15
Average User Session Statistics	Attempts	27.51	21.55	30.34
	Distinct Topics	2.20	2.31	2.85
	Distinct Questions	8.88	8.9	11.16
Average Sessions		4.11 ± 0.70	3.75 ± 0.53	4.14 ± 0.75
Pre-test score (M ±SE)		6.38 ± 1.12	9.56 ± 1.29	4.97 ± 0.85
Post-test score (M ±SE)		13.71 ± 1.00	17.12 ± 0.86	
Normalized Knowledge Gain		0.43 ± 0.07	0.36 ± 0.05	
		IntrospectiveViews		
Average Comparison mode	Class on Average	3.33 ± 0.71		
	Peers	6.83 ± 2.25		
	Topics	4.00 ± 0.79		
	Questions	4.67 ± 1.36		

The assumptions about the impact of social features of IV can be validated only if we can show some evidence that these features were really used by students. To collect this evidence, we looked at how students use the provided ability to compare their models with those of their peers' models. We found that students compared their own models to the models of their peers on the average of 6.83 times on average. This is strong evidence that the social features were used and that they had a chance to provide social guidance by affecting student question selection. But can we really argue that peer progress data could guide the student to appropriate topics and questions? Could it be just curiosity? To answer this question, we checked how many times a topic and a question were accessed from the peer model chart rather than from the students' own model of knowledge. We found that on average, students compared to their peers on 4 topics and made 4.67 attempts on the questions initiating from the peers' chart. The final question is whether the guidance obtained by visiting progress data of their peers benefited student learning. We found a correlation between the frequency of peer model comparisons and the learning gain. The more the students compared to their peers, the higher post-quiz scores they received (r= 0.34, p=0.004).

Out of the 18 IV users, 13 completed the questionnaire. For the purpose of analysis, we classified 17 questions into 5 categories. From the usefulness perspective, 84.5% of the students strongly agreed or agreed that the clockwise pie-chart design helped them to understand how the class content is organized. 76.9% of the students agreed or strongly agreed that the interface helped them to identify their weak points. 84.6% of the students agreed that the interface helped them to access the quizzes. 61.5% of the students agreed that the comparison mode motivated them to progress on the quizzes. However, there were 76.9% of students who did not think the comparison mode allowed them to identify a classmate to help them on difficult topic regardless of the positive effects of using the comparison mode (proven in the previous section). The results suggested that the students generally had a high opinion of agreement on the usefulness of the system and indentified the system's inability to find a comparable peer from the current design. Considering the Ease of Use & Ease of Learning in the system, students found it easy to learn how to use the system (92.3%), easy to remember how to use it (92.3%) and learned how to use it quickly (84.6%). They considered that the interface was easy to use (76.9%), it was user friendly (69.2%) and required fewest steps to accomplish the task[7] (66.7%). There was not a single strong disagreement with the questions of this category. In the category of Satisfaction, students liked the system. 76.9% were strongly satisfied with the system. They determined that the interface was fun (69.2%) and pleasant (76.9%) to use. 91.3% of the students would recommend it to their classmates. In terms of Privacy and Data Sharing, 84.6% of the students appreciated the feature of comparing their progress with others. 69.2% of them felt comfortable in sharing their progress with others. However, some of them had concerns on sharing the data with others. 15.4% of them do not want to share any data with others at all. 30.8% of them would like to selectively share data with others, for example, display the model anonymously or selectively share the data (either their progress or success). We also investigated the reasons of why students view the progress of other students. We found that 46.2% of the students viewed others progress out of curiosity. 46.2% of them knew the ones they viewed are good students or are good at specific topic. To extend the current model on aspects other than progress, we also collected students' opinions on such attributes as success rate, selected topics, good progress and good success rate. 46.2% of the students are willing to share everything to everyone. 23.1% are willing to share their overall progress to selected people. 23.1% of them would only share the good progress or success rate to everyone. Only 1 student (7.6%) was extremely private and was not willing to share anything to anyone. The results indicated that students were generally positive toward the data sharing idea provided the privacy management to make them feel in charge.

[7] One of the survey participants did not answer this question (B.3). The percentile was calculated based on the responses from the remainder of the participants.

G.3. Sumamry and Discussion

We observed that the parallel IntrospectiveViews interface caused an increase in all the usage parameters in comparison to a regular portal-based access system. While the increase was slightly smaller and conservative in comparison to the similar increase caused by our earlier system (JavaGuide) non-social open student modeling interface of our earlier system JavaGuide, the IntrospectiveViews interface allowed the student to achieve a higher success rate in answering the questions. In addition, the system and most of its features were highly praised by the students.

The current results are encouraging and suggest new challenges for the future work. Based on our experience, we identified few possible areas for improvement in the future.

- Adaptive navigation support: based on our previous experiences (Hsiao, et al., 2010) adaptive navigation support can dramatically increase the likelihood of answering the questions correctly. Therefore, the current design can be further improved with the additions of adaptive navigation support feature such as providing icon abstractions etc.
- Personalized guidance: the positive correlation between comparison with peers and learning gain encourages us to further look at the effects of comparison between students with different levels of knowledge; for example, a recommendation about whose models to explore.
- Privacy management: students have different levels of concerns about the privacy side for data sharing. Therefore, in the future, we have to enable the privacy setting in a sensitive manner to accommodate assorted scenarios.
- Visualizing models of multiple peers: to help users to navigate through the peers' models, the interface should be able to display multiple models at a time. The next

version will contain a pane listing miniature copies of progress pie charts of all classmates. The user will be able to sort peers by overall progress, progress in a given topic, name, and other attributes.

APPENDIX H

PROGRESSOR

Progressor is an open social student modeling visualization. It provides students with a holistic and easy-to-grasp view on their progress and allows relating it to the progress of other students in the class. It is an enhancement from our previous study with the system, Parallel IntrospectiveViews , which was presented in APPENDIX G as well as in (Bakalov, 2010; Hsiao, et al., 2011). Progressor is a merger of two our earlier works, namely the QuizJET system (Hsiao, et al., 2010) for the authoring and delivery of parameterized questions for the Java programming language and the IntrospectiveViews interface (Fedor Bakalov, et al., 2010a, 2010b) for visualization of semantic user models.

The visualization of Progressor is presented in Figure. As our previous study (Hsiao, et al., 2011) shows, Parallel IntrospectiveViews not only helped students in understanding the organization of class lectures and accessing quizzes, but also caused 28% increase of the number of attempts on questions. However, we believed that the motivational effects can be even stronger if students are provided with a ranking of their peers by progress. To check this hypothesis we developed a ranked list of model previewer. In Progressor (Fig. 44 & 45) the user is provided with a sortable list of thumbnails of pie charts representing progress of the user's peers. Unlike the list of student names in the previous version, the list with thumbnails of progress charts provide an easy to grasp overview of the peers' performance. Apart from sorting students by name and model access, the user can sort peers' models by overall progress and by progress in a certain topic. By choosing the option for sorting by progress, the interface will sort the models from the highest to the lowest progress. Also, it will display the models of the three students with the highest progress on the top. We believe that by displaying the progress of top students in such a manner can make the rest of the class eager to catch up with them. The sorted list rudimentarily contains the thumbnail of the user's own progress, which allows determining his/her ranking in the class with respect to either the overall progress or the progress in a selected topic. Also, the list contains a thumbnail with the average progress of the entire class. We believe that such a way for relating the student's own progress to the progress of other individual students and the class on average can be a strong motivating factor for completing quizzes in a timely manner and achieving better scores.

Figure 44. Progressor: Peers' progress are displayed as thumbnails and listed at the side of the user's own model.

In addition to the preview of peers' progress shown as thumbnails, the user can obtain a detailed view on the progress of an individual peer. By clicking the thumbnail of a certain student, the interface will turn into the one-to-one comparison mode (similar to the one shown in Figure. 45). We intend to simplify the process of selecting a peer for comparison from the drop-down menu which was originally located on the top of the right pane in Parallel IntrospectiveViews. In the comparison mode, the user can obtain detailed information about the peer's progress, including the information about the progress on individual quizzes. Also Progressor has a function for privacy management. The user can grant and discontinue access to his/her progress data for each peer individually. The pie charts of the closed models are shaded in dark grey. The interface allows sending requests for access to models of other peers. The privacy settings for each peer are displayed on the peer's thumbnail as two arrows: the left arrow indicates the peer's access to the user's data and the right arrow indicates the user's access to the peer's data. The arrows may be in one of the three shades: green – access granted, red – no access, yellow – access requested. By clicking an arrow, the user can change the access status for each peer individually, e.g., by clicking the left arrow the user can grant and discontinue the peer's access to the own model. In such a way the user should be able to quickly define the desired progress sharing settings. Moreover, we enhanced the color shades to a better granularity by adopting the standard color scheme, which was implemented to establish an understanding for inferring to students' progress in detail.

Figure 45. Progressor: Peers model comparison.

H.1. Experimental setup and data collection:

To assess the impact of our technology, we have conducted a thorough evaluation in a semester-long classroom study. The study was performed in an undergraduate Object-Oriented Programming course offered by the School of Information Sciences, University of Pittsburgh in the 2011 Spring semester. Progressor was introduced to the class at the beginning of the course and served as the non-mandatory course tool over the complete semester period. All student activity with the system was recorded. For every student attempt to answer a question, the system stored a timestamp, the user's name, the question, quiz, and session ids, and the correctness of the answer. We also recorded the frequency and timing of student model access and comparisons, sorting. Pre- and post- tests were administered at the beginning and the end of the semester for measuring the students' learning gain. At the end of the semester, the students were asked to provide their subjective feedback about the system and its features by filling the evaluation questionnaire. To obtain a deeper understanding of how students interact with Progressor, we compared the student work with self-assessment quizzes through Progressor (Column 3 in Table. 4) with two other comparable classes: a class that accessed quizzes using a traditional course portal with no social visualization (Column 1 in Table. 16), and a class that accessed self-assessment quizzes through Parallel IntrospectiveViews (Column 2 in Table. 4). Capitalizing our past experiences with open student modeling in JavaGuide (Hsiao, et al., 2010) & Parallel IntrospectiveViews (Hsiao, et al., 2011) we expected that the new implementation of Progressor would not only encourage the students to work more with the system, but also utilize the traces of social guidance and result in a better learning outcome.

Table 16. Systems usage summary

		1	2	3
		QuizJET	QuizJET w/ IV	Progressor
	Parameters	n=16	n=18	n=30
Average User Statistics	Attempts	80.81±22.06	113.05±15.17	205.73±40.46
	Success Rate	42.63%±1.99%	71.35%±3.39%	68.39%±4.32%
	Distinct Topics	7.81±1.64	9.06±1.39	11.47±1.34
	Distinct Questions	33.37±6.50	36.5±5.69	52.7±6.92
	Sessions	3.75±0.53	4.11±0.70	8.4±1.39
Pre-test score (M ±SE)		9.56±1.29	6.38±1.12	3.53±0.56
Post-test score (M ±SE)		17.12±0.86	13.71±1.00	14.61±0.64
Normalized Knowledge Gain		0.36±0.05	0.43±0.07	0.57±0.05

H.2. System Usage

Among 51 registered students, 30 students in the class used Progressor on a regular basis during the course period. On average, students made 205.73 attempts and obtained 68.39%

success rate in answering the self-assessment questions. Students achieved significant higher attempts on Progressor than in the two other conditions described earlier, $F_{13}(1, 61)= 6.957$, $p<.05$, $\eta^2=.102$; $F_{23}(1, 61)= 4.174$, $p<.05$, $\eta^2=.064$. In addition, students received significantly higher success rate than accessing self-assessment questions through a traditional portal (QuizJET), $F_{13}(1, 61)= 12.043$, $p<.01$, $\eta^2=.165$. To unveil the *Course Coverage* that students explored through the new interface, we calculated the number of distinct topics attempted by the student and the number of distinct questions attempted by the student. We found that students tried 11.47 distinct topics and 52.7 distinct questions on Progressor averagely. There were 46.86% and 57.92% increase respectively compared to the usage of no social visualization interface.

In Table 17, we summarize the usage for each social feature implemented in the open social student modeling interfaces. We counted how many times the students compared to the *class on average* model or any peer models from the class, how many topics and questions were attempted from their peers' models. We found that students compared to their peers more often. They attempted twice more topics through their peers' interfaces in Progressor than in QuizJET with IntrospectiveViews. There was a 35.5% increase of the questions attempts made through the peers' interfaces. Both evidence demonstrated that Progressor drove students to increase the course coverage by exploring more topics and questions through the open social student modeling interfaces. However, we did not find that students compared the class on average frequently in Progressor. This result was not a surprise. We assumed the thumbnail feature had already provided the snapshot comparisons. We hypothesized that students made explicit clicks on their peers' models to perform comparisons were driven by their self-motivation. It led us to further hypothesize that the Top 3 models appeared on the thumbnail list motivate students to not only work more but also work carefully to achieve a better success. To verify these hypotheses, we performed a deeper analysis in the next two sections.

The statistics of sorting features is also reported in Table 5. Since sorting was a new feature in Progressor, we did not have a base line to compare to. Our question was whether the sorting feature had any impacts on students. We found that there were significant positive correlations between the frequencies of peer model sorting and question attempts and success rate, $r= 0.75$, $p< .01$; $r= 0.76$, $p< .01$. The results gave us some cues about how competitive the students were. The more they worked on the questions and the more they succeeded in answering the questions, the more they cared about where they were ranked in the class.

Table 17. Open social student model interfaces usage summary

		QuizJET w/ IV	Progressor
	Parameters	n=18	n=30
Average Comparisons	Class on Average	3.33±0.71	0.22±0.07
	Peers	6.83±2.25	8.78±1.31
	Topics	4.00±0.79	9.00±1.39
	Questions	4.67±1.36	6.33±1.12
		Sort by progress	34
		Sort by name	25
		Sort by open	43

H.3. Motivational Effects: Confirming the Value of Open Social Student Modeling

In our pre-study (Hsiao, et al., 2011), we found that the combination of knowledge-based and social guidance together guided the students to better success in answering the quizzes. However, the results also showed that the interface actually mediated the motivating effects of progress visualization making the whole class a bit less adventurous and more conservative than without social guidance. Therefore, the new design of social visualization – Progressor - was intended to enhance the motivational values of Open Social Student Modeling. To evaluate the enhancement, we looked at the effects on the system usage. The major visual improvement from the previous design is the thumbnails preview on peers' models compared to the original dropdown box with the list of names to access peer student models. Although both interfaces require only one click to begin the detail models comparisons, the dropdown box design mimicked the models by a list of names which may demand the users more mental efforts to associate their own performance to their peers'. In contrast, the thumbnails in Progressor provide the users with a straightforward way to navigate through their peers' models, which we assume to yield a more competitive and yet motivating environment to let students be aware of where they are in contrast to the entire class. The system usage data showed that the students achieved significantly higher success rate than no guidance at all, $F_{13}(1, 61)= 12.043$, $p<.01$, $\eta^2=.165$. Such results were consistent with previous study in demonstrating the knowledge-based and social guidance together do a better job in guiding the students to questions that they are ready to handle than knowledge-based guidance alone. However, in Progressor, students significantly attempted more of the self-assessment questions than QuizJET on a course portal with no visualization condition ($F_{13}(1, 61)= 8.805$, $p<.01$, $\eta^2=.117$) with a fairly huge variance. There are two possible reasons to explain the huge variance. Either the students were highly motivated and competitive to do more work or some of the students hated the system and had very limited use of it in contrast to those who loved it and used it heavily. To investigate this issue we looked at two other parameters such as the average number of distinct questions had been solved per student and the average amount of time spent on the system (Table 18). We found that students explored more topics and tried significantly more distinct questions. In addition, the amount of time spent on the system (in terms of the sessions) was doubled. These numbers demonstrated the students were more engaged in Progressor than the other conditions. The average amount of time spent on the system was so long which allowed us to rule out the second possibility of disliking the system and caused the variance.

H.4. The Evidence of Social Guidance

In our past work (Brusilovsky, et al., 2011; Hsiao, et al., 2010), we found evidence that in social guidance systems, stronger students were better understanding of the subject and ended up leading the way to discover most relevant resources and creating guidance trails for the weaker students. Thus, to capitalize the impact of social features provided by IntrospectiveViews (Hsiao, et al., 2011), we not only reported the evidence of usage (Table 16&17) on the social features in Progressor, but also performed a deeper analysis on student activities.

By taking into account of the lecture coverage associated with all students' actions, Figure 46 visualizes over 6100 transactions of all the question attempts performed through Progressor. Based on the visual display of the Top 3 student models and the rest of class ones, we color coded the activities into orange and blue. Orange ones represent the activities generated by Top 3 students and blue ones are the rest of the students. The time of the action is marked as the X axis and the question complexity goes by Y axis from easy to complex. We found 4 interesting zones within this figure. Zone "A" contains the *current* activity that students performed along the lecture stream of the course. Students had been working with the system very consistently throughout the course schedule. Zone "B" represents the region of preparation for final exam. Therefore all ranges of complexities of questions were attempted. Noted the weaker students made the most of the attempts in this zone. Zone "C" contains all of the attempts which students were self-motivated to work for mastery of the subject. Zone "D" contains the attempts, which students performed ahead of the course schedule. Surprisingly, this zone actually included a substantial proportion of the attempts. It demonstrated the system was actually inviting students to challenge themselves to move a little bit ahead of the course pace instead of passively progressing. It resolved our concerns from previous design and converted a conservative environment to a more adventurous one. In addition, we found that the Top 3 students were leading the adventure of future-topics along the class. Such effect was consistent as the course became more complex and complex. Moreover, in order to further examine the social guidance effect, we categorized the students into two groups based on their pre-test scores (ranging from a minimum 0 to a maximum 20). Due to the pre-test scores were positive skewed, we split the two groups by setting the threshold at score 7. Strong students scored 7 points or higher (7~13) and weak students scored less than 7 (0~6). We found that the strong students generally explored the questions ahead of the weak ones. The effect was especially noticeable in the *social navigation adventure* zone (Figure 43). Strong students worked on Progressor first and left the implicit good traces for weak students to follow up. All zones remain the same patterns as Figure 47.

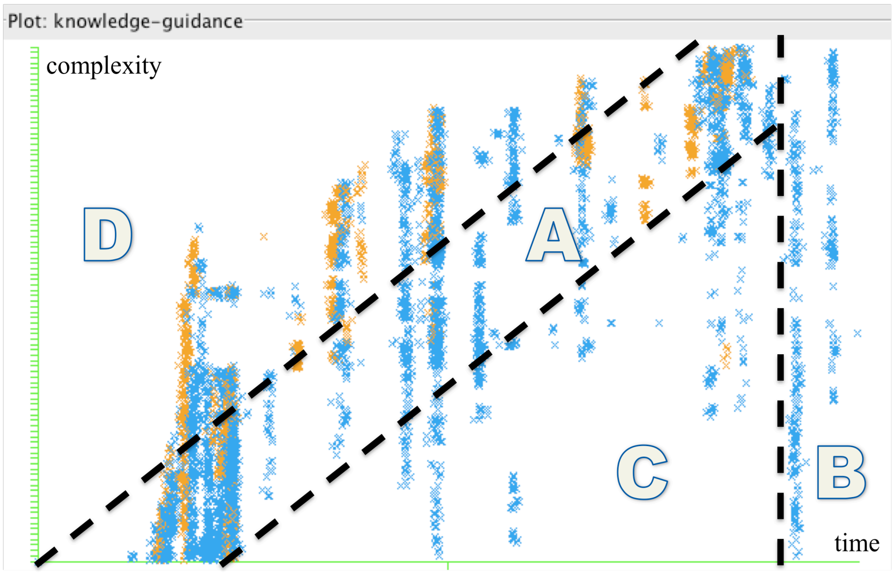


Figure 46. Time distribution of all attempts performed by the students through Progressor. X axis is the Time; Y axis is the complexity of the quizzes. Orange dots represent the Top3 students' actions; blue ones are the actions belonged to the rest of the class. Zone "A" – lecture stream, zone "B" – final exam cut, zone "C" – self-motivated work with the material from earlier lectures, and zone "D" – social navigation adventure.

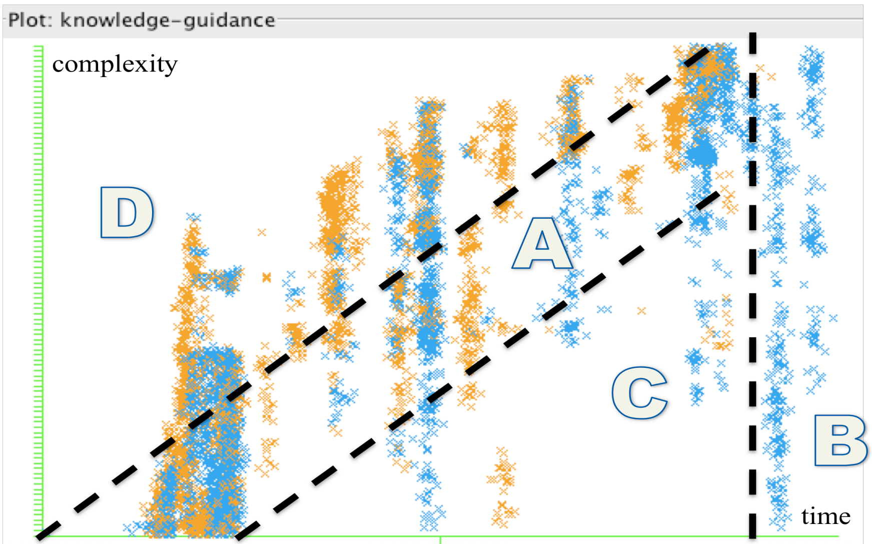


Figure 47. Time distribution of all attempts performed by the students through Progressor color coded by strong(orange) and weak(weak) knowledge levels.

H.5. The Effects on Student Learning

To evaluate the impacts of the open student modeling interface on student learning, we measured the differences between pre- and post- tests. We found that in all conditions (QuizJET, IV and Progressor), the students achieved a significant knowledge growth as measured by pre- and post- test scores, $t_1(15)= 6.108$, $p< .01$, $t_2(17)= 7.203$, $p< .01$, $t_3(29)= 14.053$, $p< .01$. Unfortunately, the assumption of homogeneity of variance was not met, Brown-Forsythe $F(2, 61)= 12.95$, $p< .05$. Due to the group that used Porgressor was particularly weak, potentially it had bigger room to improve. Therefore, in order to have fair comparison across groups, we calculated the Normalized Knowledge Gain based on formula (1). A one-way between-subjects analysis of variance was performed on the normalized knowledge gain as a function of 3 different interfaces (QuizJET, QuizJET w/ IV, and Progressor). The assumption of homogeneity of variance was met, Brown-Forsythe $F(2, 61)= 3.126$, $p>.05$. The assumption of normality was met for all systems except QuizJEt w/ IV (Table 6). All other assumptions were met. We found that student obtained significant normalized knowledge gain by working on the self-assessment questions through Progressor (M= 0.572, SE= 0.050) than QuizJET (M= 0.361, SE= 0.050), $F(1, 61)$= 1.263, p<.05, η^2= .021. It should be noted that all three studies were performed in a non-controlled classroom context where the systems were used as just supplementary course tools. The students were able to learn the subject by many ways besides the self-assessment questions from the systems.

$$NKG = \frac{posttest - pretest}{\max score - pretest} \quad (1)$$

Table 18. Test of normality of normalized knowledge gain for each system interface.

System	Shapiro-Wilk W	df	p
QuizJET	0.936	16	0.302
QuizJET w/ IV	0.873	18	0.020
Progressor	0.958	30	0.275

H.6. Subjective Evaluation

To examine the students' attitudes toward Progressor, we requested students to fill out an evaluation questionnaires at the end of the semester. The responses from students who actually used the systems over the semester have been analyzed. There were 31 students filled-in the questionnaire, 17 male and 14 female (1 student only provided his opinions about a desired course tool without using the system). In the survey, there were 22 questions, including the usability of GUI elements to users' satisfaction of the interface in general. Users were asked to evaluate the questions on a 5-points Likert scale and to provide free-text comments as they wish. We further break down the 22 questions into 5 categories, including Usefulness, Ease of Use, Ease of Learning, Satisfaction and Privacy&Data Sharing (Table 4). We found that male users had more positive attitude toward Progressor than female did across all categories, except the Ease of Learning one (not significant). The opinions on Progressor of gender differences were reported in Figure 48. On average, we do not see significant differences between genders. However, if we zoom in to examine the detail opinions by questions, we found that male students held significant higher positive attitude than female students did in two aspects: male students considered the interface helped them to plan the class work and they thought the comparison mode of the interface helped them to find the classmates who can help them on difficult topics ($F(1, 29^{8}) = 16.588$, $p < .01$; $F(1, 30) = 4.598$, $p < .05$).

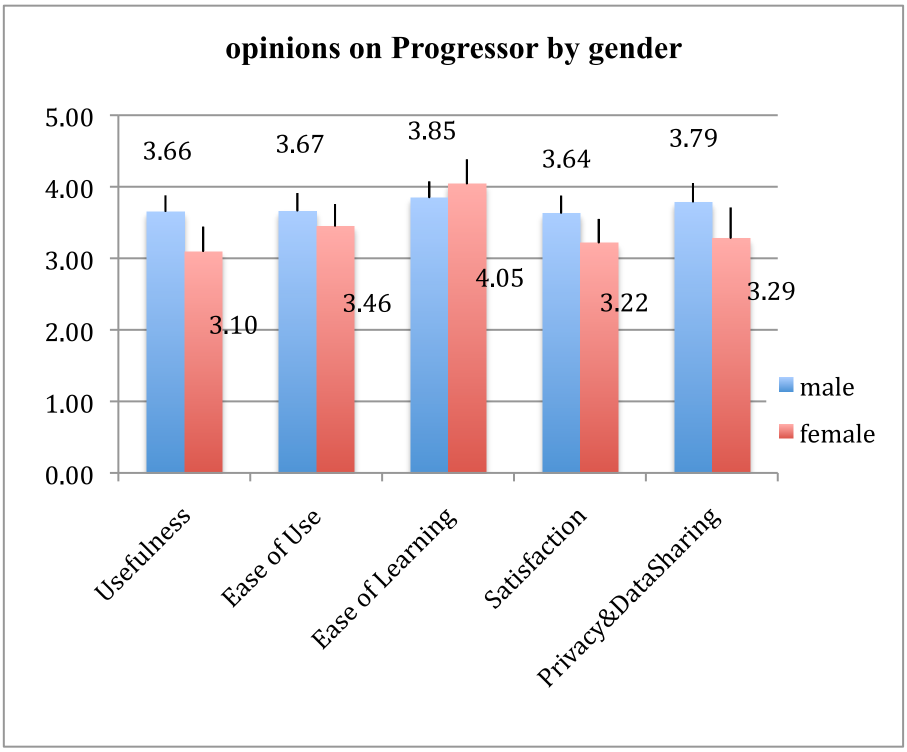


Figure 48. Students opinions on Progressor by gender

An itemized survey is presented as following. In *Usefulness* category, 74.2% of the students strongly agreed or agreed the open social student modeling interface helped them to understand how the class content is organized. 74.2% of the students agreed or strongly

[8] One of the survey participants did not answer this question (A.3). The percentile was calculated based on the responses from the remainder of the participants.

agreed that the interface helped them to identify their weak points. On the question whether the interface helped students to plan their class work, students had sporadic responses. 50% of them held the mutual point of views. Due to the system was not required for the class, we can understand that students did not have strong opinions toward agreement or disagreement. 51.6% of the student thought the interface really helped them to access the quizzes. 54.8% of them strongly agreed or agreed the thumbnails view motivated them to progress on the quizzes. 51.6% of the students agreed that the comparison mode motivated them to progress on the quizzes. In the question regarding to whether the comparison mode allowed them to identify a classmate to help them on difficult topic, student had assorted attitudes. Despite we did not obtain a consistent positive agreement as we expected, we did not get a majority of disagreement either, which was an improvement from previous design. The results suggested that the students generally had a positive attitude toward the system usefulness. Considering the *Ease of Use & Ease of Learning* in the system, students found it easy to remember how to use it (83.8%) and learned how to use it quickly (77.4%) They considered that the interface was easy to use (70.9%), user friendly (70.0%) and easy to learn how to use the system (70.9%). Students had 83.4% of mutual to strongly agreement about the interface requiring fewest steps to accomplish the task. There was only 3.33% strong disagreement across all the questions of this category. In the category of *Satisfaction*, students liked the system. 61.3% were strongly satisfied with the system and 61.3% of them would recommend it to their classmates. They generally valued the interface from neutral to strongly agree that it was fun (61.7%) and pleasant (87.1%) to use. In terms of *Privacy and Data Sharing* perspectives, most of the students had positive attitude toward this categories of questions. They had roughly the same proportion of agreement ranging from neutral to strongly agree on displaying their models anonymously and generally appreciated the feature of access by requests. 54.8% of the students appreciated the feature of comparing their progress with others. 61.3% of them felt comfortable in sharing their progress with others. Figure 49 shows the detail percentages for each question.

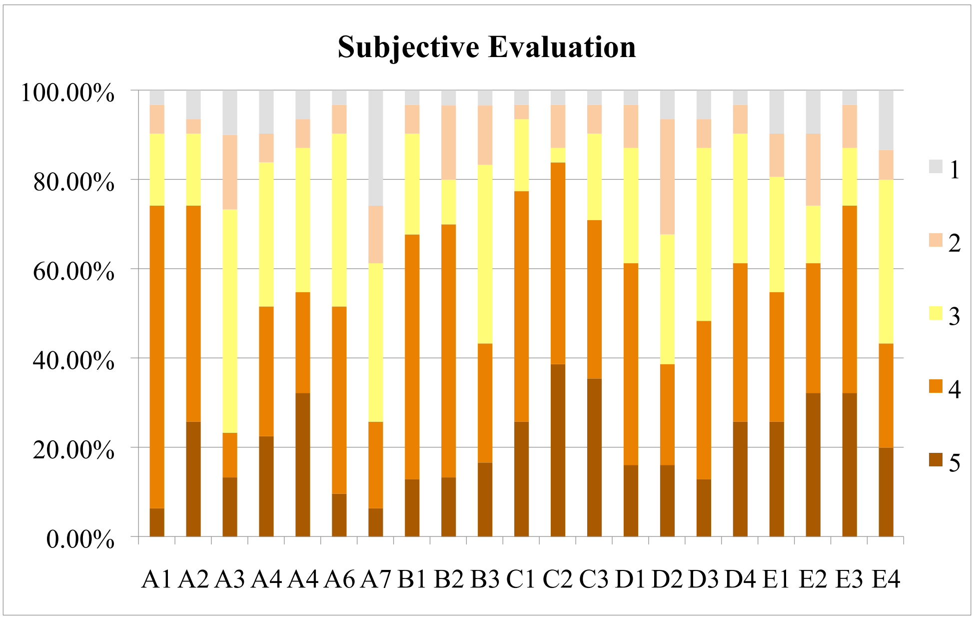


Figure 49. Summary of subjective evaluation on Progressor

H.7. Evaluation Summary

Progressor is an innovative Web-based interface, which was designed to help students to find most relevant resources in a large collection of parameterized self-assessment questions for programming. The interface was built based on the open social student models. Students were able to navigate through all their peers' models and to perform comparisons from one to another. A semester-long study was conducted and cross-compared with several similar conditions. We found that students used the non-mandatory systems heavily. We also confirmed the motivational values of personalized social guidance provided by the Progressor. The results showed that the interface encouraged students to explore more topics and motivated them to do some work ahead of the course schedule. A deeper analysis of the social guidance mechanism revealed that the top students, which provided an implicit social guidance for the rest of the class, successfully led the way to discover most relevant resources creating good trails for weaker students. In addition, we also found that there were significant positive correlations between the frequencies of peer model sorting and question attempts and success rate. The more they worked on the questions and the more they succeeded in answering the questions, the more they cared about where they were ranked in the class. The study results also demonstrated that students were more engaged in the system by spending more time in working with self-assessment questions, attempting more questions and achieving higher success rate in answering them. At last but not least, students highly praised the interface and liked to recommend it to other courses as well. Moreover, male were appeared to be more favorable to the interface in general. Significant differences between male and female were found in their opinions regarding to the usefulness of the interface.

APPENDIX I

PRE- AND POST- TESTS OF JAVA KNOWLEDGE

Question 1

Consider the following code segment:

```
public class MyTester {
  public static void main(String[] args) {
    int i = 14;
    int j = 20;
    int k;

    k = j / i * 7 % 4;
}
```

What is the final value of the variable k: ________

Question 2

Consider the following code segment.

```
.......
  int myYear = 2012;
  String myText = new String("Hello, IS17!");
  int result = 0;

  if (myText.length() > 20)
  {
    result = 1;
    if (myText.length() < 30 && myYear >= 2012)
      result += 5;
  }
  else
    if (myYear >= 2000)
      result += 10;
    else
      result += 100;
.......
```

What is the final value of the variable result**: _________**

Question 3

Consider partial implementation of the class Rectangle**:**

```
public class Rectangle {

  private double x;
  private double y;
  private double height;
  private double width;

  public Rectangle (double x, double y, double height, double width) {
    this.x = x;
    this.y = y;
    this.height = height;
    this.width = width;
  }

......
}
```

Assume, that one more method has been added to the class:

```
  public void magnify (int ratio) {
    height = height * ratio;
    width = width * ratio;
  }
```

What would be the output of the following code fragment using the new method?

```
......
Rectangle myBox = new Rectangle(50, 40, 10, 10);
myBox.magnify(3);
System.out.println(myBox.getHeight());
System.out.println(myBox.getWidth());
......
```

Output:

____

____

____

____

Question 4

For each of the following 3 code segments, what is the final value of result**?**

<u>Code segment 1</u>:

```
int i = 3;
int result = 0;
while (i < 4) {
  result = result + i;
  i++;
}
```

result: __________

<u>Code segment 2:</u>

```
int i = 4;
int result = 0;
do {
  result = result + i;
  i++;
} while (i < 4);
```

result: __________

<u>Code segment 3:</u>

```
int result = 0;
for (int i = 5; i > 0; i--)
  result = result + i;
```

result: __________

Question 5

What would be the output of the following code fragment:

```
int[] data = new int[5];
for (int i = 0; i < 5; i++)
  data[i] = i*i;
data[2] += 1;

System.out.println(data[2]);
```

Output:

____

____

____

____

Question 6

What would be the output of the following code fragment:

```
ArrayList<Double> list = new ArrayList<Double>();
list.add(1.1);
list.add(2.2);
list.add(3.3);
list.remove(0);
for(Double d : list)
  System.out.println(d);
```

Output:

____
____
____
____

Question 7

Class Rectangle **implements interface** Shape**, that declares method**
public boolean contains (double x, double y)
// Tests if the specified coordinates are inside the boundary of the Shape.

The implementation of the method contains **in class** Rectangle **is following:**

```
public boolean contains(double x, double y) {
  double x0 = getX();
  double y0 = getY();
  return (x >= x0 && y >= y0 && x < x0 + getWidth() && y < y0 + getHeight());
}
```

What will be the output of the following code fragment:

```
Shape box = new Rectangle( 0, 0, 10, 20);
System.out.println(box.contains(50, 10));
```

Output:

____
____
____
____

Question 8

Consider the fragment of Class ColoredRectangle **:**

```
public class ColoredRectangle extends Rectanlge
{

  String color;

  public ColoredRectangle(double x, double y, double h, double w, String c)
  {
    super(x, y, h, w);
    color = c;
  }

  public String getColor() {
    return color;
  }

.......

}
```

What will be the output of the following code fragment using ColoredRectangle**:**

```
  ColoredRectangle box = new ColoredRectangle (20, 10, 40, 30, “Blue”);
  System.out.println(box.getColor());
  System.out.println(box.getHeight());
  System.out.println(box.getWidth());
```

Output:

____

____

____

____

Question 9

Take into account information in questions 7 and 8.
Consider the following statement:

```
ColoredRectangle box = new ColoredRectangle(0, 0, 30, 50, “Green”);
```

Which of the following conditions return false**:**

a) if (box instanceOf Object)

b) if (box instanceOf ColoredRectangle)

c) if (box instanceOf Point)

d) if (box instanceOf Rectangle)

e) if (box instanceOf Shape)

f) if (box instanceOf BankAccount)

g) if (box instanceOf ArrayList)

Question 10

What is the output of the following code segment?

```
int a = 4 + 4;
int b = 5 + 5;

if (a != b)
 System.out.println(" Not equal ");

if (a == b)
 System.out.println(" Equal ");
```

Output:

____

____

____

____

APPENDIX J

USER FEEDBACK QUESTIONNAIES

A. Usefulness

1. The interface helps me to understand how the class content is organized. (1 2 3 4 5)
2. The interface helps me to identify my weak points. (1 2 3 4 5)
3. The interface helps me to plan my class work. (1 2 3 4 5)
4. The interface helps me to access the content (examples and quizzes). (1 2 3 4 5)
5. The colors indication of the progress is clear. (1 2 3 4 5)
6. Viewing my classmates' progress motivates me to progress on mine. (1 2 3 4 5)
7. Sorting the progress helps me to find who can help on difficult topics. (1 2 3 4 5)
8. Sorting the success helps me to find who can help on difficult topics. (1 2 3 4 5)
9. The annotated examples have been really helpful. (1 2 3 4 5)
10. The self-assessment quizzes have been really helpful. (1 2 3 4 5)

B. Ease of Use

1. The interface is easy to use. (1 2 3 4 5)
2. The interface is user friendly. (1 2 3 4 5)
3. The interface requires the fewest steps possible to accomplish what I want to do with it. (1 2 3 4 5)

C. Ease of Learning

1. I learned how to use the interface quickly. (1 2 3 4 5)
2. I easily remember how to use the interface. (1 2 3 4 5)
3. It is easy to learn how to use the interface. (1 2 3 4 5)

D. Satisfaction

1. I am satisfied with the interface. (1 2 3 4 5)
2. The interface is fun to use. (1 2 3 4 5)
3. The interface is pleasant to use. (1 2 3 4 5)
4. I would recommend the interface to my classmates. (1 2 3 4 5)

E. Privacy and Data Sharing

1. I like the idea of comparing my progress with other students. (1 2 3 4 5)
2. I feel comfortable sharing my progress with others. (1 2 3 4 5)
3. I do not mind that my progress is displayed anonymously in the average progress of the entire class. (1 2 3 4 5)

F. Other comments and suggestions for improvement:

____

____

____

BIBLIOGRAPHY

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from Examples: Instructional Principles from the Worked Examples Research. *REVIEW OF EDUCATIONAL RESEARCH, 70*(2), 181-214.

Azevedo, R., Guthrie, J. T., & Seibert, D. (2004). The role of self-regulated learning in fostering students' conceptual understanding of complex systems with hypermedia. *Journal of Educational Computing Research, 30*(1), 87–111.

Bakalov, F., Hsiao, I.-H., Brusilovsky, P., & Konig-Ries, B. (2011, 18-22 Sept. 2011). *Progressor: Personalized visual access to programming problems.* Paper presented at the 2011 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Pittsburgh, PA.

Bakalov, F., König-Ries, B., Nauerz, A., & Welsch, M. (2010a, June 22-24, 2010). *IntrospectiveViews: An Interface for Scrutinizing Semantic User Models.* Paper presented at the 18th International Conference on User Modeling, Adaptation, and Personalization (UMAP 2010), Big Island, HI, USA.

Bakalov, F., König-Ries, B., Nauerz, A., & Welsch, M. (2010b). *Scrutinizing User Interest Models with IntrospectiveViews* Paper presented at the 18th International Conference on User Modeling, Adaptation, and Personalization Big Island of Hawaii, USA.

Barolli, L., Koyama, A., Durresi, A., & De Marco, G. (2006). A web-based e-learning system for increasing study efficiency by stimulating learner's motivation. *Information Systems Frontiers, 8*(4), 297-306.

Bra, P. D., & Calvi, L. (1998). AHA! An open Adaptive Hypermedia Architecture. [doi: 10.1080/13614569808914698]. *New Review of Hypermedia and Multimedia, 4*(1), 115-139.

Brandt, J., Dontcheva, M., Weskamp, M., & Klemmer, S. R. (2010). *Example-centric programming: integrating web search into the development environment*. Paper presented at the Proceedings of the 28th international conference on Human factors in computing systems.

Brna, P. (1998). Searcing for examples with a programming techniques editor. *Journal of Computing and Information Technology, 6*(1), 13-26.

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction, 6*(2-3), 87-129.

Brusilovsky, P. (2001a, June 23-26, 2001). *Adaptive Educational Hypermedia.* Paper presented at the Tenth International PEG conference, Tampere, Finland.

Brusilovsky, P. (2001b). Adaptive hypermedia. *User Modeling and User Adapted Interaction, 11*(1/2), 87-110.

Brusilovsky, P. (2001c, October 23-27, 2001). *WebEx: Learning from examples in a programming course.* Paper presented at the WebNet'2001, World Conference of the WWW and Internet, Orlando, FL.

Brusilovsky, P. (2003). Adaptive navigation support in educational hypermedia: the role of student knowledge level and the case for meta-adaptation. *British Journal of Educational Technology, 34*(4), 487-497.

Brusilovsky, P. (2007). Adaptive navigation support. In P. Brusilovsky, A. Kobsa & W. Neidl (Eds.), *The Adaptive Web: Methods and Strategies of Web Personalization* (Vol. 4321, pp. 263-290). Berlin Heidelberg New York: Springer-Verlag.

Brusilovsky, P., Chavan, G., & Farzan, R. (2004, August 23-26, 2004). *Social adaptive navigation support for open corpus electronic textbooks.* Paper presented at the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2004), Eindhoven, the Netherlands.

Brusilovsky, P., & Eklund, J. (1998). A study of user-model based link annotation in educational hypermedia. *Journal of Universal Computer Science, 4*(4), 429-448.

Brusilovsky, P., Eklund, J., & Schwarz, E. (1998, 14-18 April 1998). *Web-based education for all: A tool for developing adaptive courseware.* Paper presented at the Seventh International World Wide Web Conference, Brisbane, Australia.

Brusilovsky, P., Grant, N., Hsiao, S., Moore, K., & Sosnovsky, S. (2007, October 15-19, 2007). *Personalized E-Learning for Distance Courses in Community Colleges.* Paper presented at the World Conference on E-Learning, E-Learn 2007, Quebec City, Canada.

Brusilovsky, P., Hsiao, I. H., & Folajimi, Y. (2011). QuizMap: Open Social Student Modeling and Adaptive Navigation Support with TreeMaps

Towards Ubiquitous Learning. In C. Kloos, D. Gillet, R. Crespo Garc√≠a, F. Wild & M. Wolpers (Eds.), (Vol. 6964, pp. 71-82): Springer Berlin / Heidelberg.

Brusilovsky, P., & MIller, P. (2001). *Course Delivery Systems for the Virtual University*. Amsterdam: Elsevier Science and International Association of Universities,.

Brusilovsky, P., & Pesin, L. (1994, May 16-19, 1994). *An intelligent learning environment for CDS/ISIS users.* Paper presented at the The interdisciplinary workshop on complex learning in computer environments (CLCE94), Joensuu, Finland.

Brusilovsky, P., & Sosnovsky, S. (2005a, June 27-29, 2005). *Engaging students to work with self-assessment questions: A study of two approaches.* Paper presented at the 10th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE'2005, Monte de Caparica, Portugal.

Brusilovsky, P., & Sosnovsky, S. (2005b). Individualized exercises for self-assessment of programming knowledge: An evaluation of QuizPACK. *J. Educ. Resour. Comput., 5*(3), 6.

Brusilovsky, P., Sosnovsky, S., & Shcherbinina, O. (2004, November 1-5, 2004). *QuizGuide: Increasing the Educational Value of Individualized Self-Assessment Quizzes with Adaptive Navigation Support.* Paper presented at the World Conference on E-Learning, E-Learn 2004, Washington, DC, USA.

Brusilovsky, P., Sosnovsky, S., & Shcherbinina, O. (2005). *User Modeling in a Distributed E-Learning Architecture*. Paper presented at the 10th International Conference on User Modeling (UM'2001).

Brusilovsky, P., Sosnovsky, S., & Yudelson, M. (2009). Addictive links: The motivational value of adaptive link annotation. *New Review of Hypermedia and Multimedia, 15*(1), 97-118.

Brusilovsky, P., Sosnovsky, S., and Shcherbinina, O. (2004). *QuizGuide: Increasing the Educational Value of Individualized Self-Assessment Quizzes with Adaptive Navigation Support*. Paper presented at the World Conference on ELearning, E-Learn 2004.

Brusilovsky, P., Sosnovsky, S., and Yudelson, M. (2004). *Adaptive Hypermedia Services for E-Learning.* Paper presented at the Proceedings of Workshop on Applying Adaptive Hypermedia Techniques to Service Oriented Environments at the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Eindhoven, the Netherlands.

Brusilovsky, P., & Yudelson, M. (2008). From WebEx to NavEx: Interactive Access to Annotated Program Examples. *Proceedings of the IEEE, 96*(6), 990-999.

Bull, S. (2004, September 29 - October 3, 2004). *Supporting learning with open learner models.* Paper presented at the 4th Hellenic Conference on Information and Communication Technologies in Education, Athens, Greece.

Bull, S., & Britland, M. (2007, June 25, 2007). *Group Interaction Prompted by a Simple Assessed Open Learner Model that can be Optionally Released to Peers.* Paper presented at the Workshop on Personalization in E-learning Environments at Individual and Group Level at the 11th International Conference on User Modeling, UM 2007, Corfu, Greece.

Bull, S., & Kay, J. (2007). Student Models that Invite the Learner In: The SMILI:() Open Learner Modelling Framework. *International Journal of Artificial Intelligence in Education, 17*(Volume 17, Number 2 / 2007), 89-120.

Burow, R., & Weber, G. (1996, June 12-14, 1996). *Example explanation in learning environments.* Paper presented at the Third International Conference on Intelligent Tutoring Systems, ITS-96, Berlin.

Butler, D. L. (1998). The strategic content learning approach to promoting self-regulated learning: A report of three studies. [doi:10.1037/0022-0663.90.4.682]. *Journal of Educational Psychology, 90*(4), 682-697.

C. Plaisant, G. C., University of Maryland; C. Lukehart, D. Schiro, J. Ryan, ChevronTexaco. (2003). *Using Visualization Tools to Gain Insight Into Your Data*. Paper presented at the SPE Annual Technical Conference and Exhibition.

Card, S. K., Mackinlay, J. D., & Shneiderman, B. (1999). *Readings in Information Visualization*.

Cecez-Kecmanovic, D., & Webb, C. (2000). Towards a communicative model of collaborative Web-mediated learning. *Australian Journal of Educational Technology, 16*(1), 73-85.

Chen, Z.-H., Chou, C.-Y., Deng, Y.-C., & Chan, T.-W. (2007). Active Open Learner Models as Animal Companions: Motivating Children to Learn through Interacting with My-Pet and Our-Pet. *International Journal of Artificial Intelligence in Education, 17*(Volume 17, Number 2 / 2007), 145-167.

Davidovic, A., Warren, J., & Trichina, E. (2003). Learning benefits of structural example-based adaptive tutoring systems. *IEEE Transactions on Education, 46*(2), 241-251.

De Bra, P., & Calvi, L. (1998). AHA! An open Adaptive Hypermedia Architecture. *The New Review of Hypermedia and Multimedia, 4*, 115-139.

Dieberger, A. (1997). Supporting social navigation on the World Wide Web. *International Journal of Human-Computer Interaction, 46*, 805-825.

Dieberger, A., Dourish, P., Höök, K., Resnick, P., & Wexelblat, A. (2000). Social navigation: Techniques for building more usable systems. *interactions, 7*(6), 36-45.

Dijkstra, P., Kuyper, H., Werf, G. v. d., Buunk, A. P., & Zee, Y. G. v. d. (2008). Social Comparison in the Classroom: A Review. *REVIEW OF EDUCATIONAL RESEARCH, 78*(4).

Dimitrova, V., Self, J., & Brna, P. (2001, July 13-17, 2001). *Applying interactive open learner models to learning technical terminology.* Paper presented at the 8th International Conference on User Modeling, UM 2001, Berlin.

Dolmans, D. H. J. M., De Grave, W., Wolfhagen, I. H. A. P., & Van Der Vleuten, C. P. M. (2005). Problem-based learning: future challenges for educational practice and research. [10.1111/j.1365-2929.2005.02205.x]. *Medical Education, 39*(7), 732-741.

Ellis, J. B., Wahid, S., Danis, C., & Kellogg, W. A. (2007). *Task and social visualization in software development: evaluation of a prototype*. Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems.

Falchikov, N., & Boud, D. (1989). Student Self-Assessment in Higher Education: A Meta-Analysis. *REVIEW OF EDUCATIONAL RESEARCH, 59*(4), 395-430.

Falchikov, N., & Goldfinch, J. (2000). Student Peer Assessment in Higher Education: A Meta-Analysis Comparing Peer and Teacher Marks. *REVIEW OF EDUCATIONAL RESEARCH, 70*(3), 287-322.

Farzan, R., & Brusilovsky, P. (2008). AnnotatEd: A social navigation and annotation service for web-based educational resources. *New Review in Hypermedia and Multimedia, 14*(1), 3-32.

Fedor Bakalov, I.-H. H., Brusilovsky, Birgitta König-Ries. (2010, February 2011). *Visualizing Student Models for Social Learning with Parallel IntrospectiveViews*.

Fekete, J. D., & Plaisant, C. (2002, 2002). *Interactive information visualization of a million items.* Paper presented at the Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on.

Feldman, N. S., & Ruble, D. N. (1977). Awareness of social comparison interest and motivations: A developmental study. *Journal of Educational Psychology, 69*(5), 579–585.

Festinger, L. (1954). A theory of social comparison processes. *Human Relations, 7*, 117-140.

Freeman, L. C. (2000). Visualizing Social Networks. *Journal of Social Structure, 1*.

Gershon, N., & Page, W. (2001). What storytelling can do for information visualization. *Commun. ACM, 44*(8), 31-37.

Gillan, D. J., & Callahan, A. B. (2000). A Componential Model of Human Interaction with Graphs: VI. Cognitive Engineering of Pie Graphs. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 42*(4), 566-591.

Gómez Albarrán, M. (2005). Teaching and learning of programming: A survey of supporting software tools. *The Computer Journal, 48*(2), 130-144.

Graham, C. R., & Trick, T. N. (1997, 5-8 Nov 1997). *An innovative approach to asynchronous learning using Mallard: application of Java applets in a freshman course.* Paper presented at the Frontiers in Education Conference, 1997. 27th Annual Conference. 'Teaching and Learning in an Era of Change'. Proceedings.

Grigoriadou, M., Papanikolaou, K., Kornilakis, H., & Magoulas, G. (2001, July 14, 2001). *INSPIRE: An INtelligent System for Personalized Instruction in a Remote Environment.* Paper presented at the Third workshop on Adaptive Hypertext and Hypermedia, Sonthofen, Germany.

Heer, J., & Boyd, D. (2005, 23-25 Oct. 2005). *Vizster: visualizing online social networks.* Paper presented at the Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on.

Henry, N., & Fekete, J. D. (2006). MatrixExplorer: a Dual-Representation System to Explore Social Networks. *Visualization and Computer Graphics, IEEE Transactions on, 12*(5), 677-684.

Henson, L., Dews, T., Lotto, M., Tetzlaff, J., & Dannefer, E. (2005). A mastery learning model for assessing competency of medical students using portfolios. *Journal of Clinical Anesthesia, 17*(8), 663 - 664.

Henze, N., & Nejdl, W. (2001). Adaptation in open corpus hypermedia. *International Journal of Artificial Intelligence in Education, 12*(4), 325-350.

Ho, Y.-L., Chang, P.-Y., Chen, I.-X., & Yang, C.-Z. (2010, 16-18 Dec. 2010). *Exploring user-centric interpersonal relationships in social networks using information visualization techniques.* Paper presented at the Computer Symposium (ICS), 2010 International.

Hsiao, I.-H., Bakalov, F., Brusilovsky, P., & König-Ries, B. (2011). *Open Social Student Modeling: Visualizing Student Models with Parallel IntrospectiveViews.* Paper presented at the 19th International Conference on User Modeling, Adaptation and Personalization (UMAP 2011), Girona, Spain.

Hsiao, I.-H., & Brusilovsky, P. (2008, October, 27-31, 2008). *Modeling Peer Review in Example Annotation.* Paper presented at the 16th International Conference on Computers in Education (ICCE'2008), Taipei, Taiwan.

Hsiao, I.-H., & Brusilovsky, P. (2011). The Role of Community Feedback in the Student Example Authoring Process: an Evaluation of AnnotEx. *British Journal of Educational Technology, 42*(3), 482-499.

Hsiao, I.-H., Brusilovsky, P., & Sosnovsky, S. (2008, November 17-21, 2008). *Web-based Parameterized Questions for Object-Oriented Programming.* Paper presented at the World Conference on E-Learning, E-Learn 2008, Las Vegas, USA.

Hsiao, I.-H., Sosnovsky, S., & Brusilovsky, P. (2009, September 29- October 2, 2009). *Adaptive Navigation Support for Parameterized Questions in Object-Oriented Programming.* Paper presented at the 4th European Conference on Technology Enhanced Learning (ECTEL 2009), Nice, France.

Hsiao, I.-H., Sosnovsky, S., & Brusilovsky, P. (2010). 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.

Hudson, S. E. JFlex - The Fast Scanner Generator for Java. Retrieved Jan. 02, 2009, from http://jflex.de/

Hudson, S. E. (1999). CUP Parser Generator for Java. Retrieved Jan.02, 2009, from http://www.cs.princeton.edu/~appel/modern/java/CUP/

Johnson, D. W., Johnson, R. T., & Smith, K. A. (1998). Cooperative Learning Returns to College: What Evidence is There That it Works? *Change: The Magazine of Higher Learning, 30*(4), 26-35.

Kashy, E., Thoennessen, M., Tsai, Y., Davis, N. E., & Wolfe, S. L. (1997, 5-8 Nov 1997). *Using networked tools to enhance student success rates in large classes.* Paper presented at the Frontiers in Education Conference, 1997. 27th Annual Conference. 'Teaching and Learning in an Era of Change'. Proceedings.

Kavcic, A. (2004). Fuzzy User Modeling for Adaptation in Educational Hypermedia. *IEEE Transactions on Systems, Man, and Cybernetics, 34*(4), 439-449.

Kay, J. (1997). *Learner know thyself: Student models to give learner control and responsibility.* Paper presented at the ICCE97, International Conference on Computers in Education, Malasia, Kuching, Sarawak.

Kay., J. (2008). Lifelong Learner Modeling for Lifelong Personalized Pervasive Learning. *IEEE Transaction on Learning Technologies, 1*(4), 215-228.

Koedinger, K. R., & Corbett, A. (2006). *Cognitive Tutors: Technology bringing learning science to the classroom*. New York, NY, USA: Cambridge University Press.

Kornell, N., & Son, L. K. (2009). Learners' choices and beliefs about self-testing. [doi:10.1080/09658210902832915]. *Memory, 17*(5), 493-501.

Kraker, P., Wagner, C., Jeanquartier, F., & Lindstaedt, S. (2011). *On the way to a science intelligence: visualizing TEL tweets for trend detection*. Paper presented at the Proceedings of the 6th European conference on Technology enhanced learning: towards ubiquitous learning.

Kurhila, J., Miettinen, M., Nokelainen, P., & Tirri, H. (2006). Educo- A Collaborative Learning Environment Based on Social Navigation Adaptive Hypermedia and Adaptive Web-Based Systems. In P. De Bra, P. Brusilovsky & R. Conejo (Eds.), (Vol. 2347, pp. 242-252): Springer Berlin / Heidelberg.

Lamping, J., Rao, R., & Pirolli, P. (1995). *A focus+context technique based on hyperbolic geometry for visualizing large hierarchies*. Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems.

Lindstaedt, S. N., G\, \#252, Beham, n., Kump, B., & Ley, T. (2009). *Getting to Know Your User --- Unobtrusive User Model Maintenance within Work-Integrated Learning Environments*. Paper presented at the Proceedings of the 4th European Conference on Technology Enhanced Learning: Learning in the Synergy of Multiple Disciplines.

Linn, M. C. (1992). How can hypermedia tools help teach programming. *Learning and Instruction, 2*, 119-139.

Mabbott, A., & Bull, S. (2004). Alternative Views on Knowledge: Presentation of Open Learner Models

Intelligent Tutoring Systems. In J. C. Lester, R. M. Vicari & F. b. Paragua√ßu (Eds.), (Vol. 3220, pp. 131-150): Springer Berlin / Heidelberg.

Maceachren, A. M., Wachowicz, M., Edsall, R., Haug, D., & Masters, R. (1999). Constructing knowledge from multivariate spatiotemporal data: integrating geographical visualization with knowledge discovery in database methods. *International Journal of Geographical Information Science, 13*(4), 311-334.

Mansmann, F., Fischer, F., Keim, D. A., & North, S. C. (2009). *Visual support for analyzing network traffic and intrusion detection events using TreeMap and graph representations*. Paper presented at the Proceedings of the Symposium on Computer Human Interaction for the Management of Information Technology.

Mazza, R., & Dimitrova, V. (2007). CourseVis: A graphical student monitoring tool for supporting instructors in web-based distance courses. [doi: 10.1016/j.ijhcs.2006.08.008]. *International Journal of Human-Computer Studies, 65*(2), 125-139.

Melis, E., Andres, E., Budenbender, J., Frischauf, A., Andrès, E. M. E., Goguadze, G., et al. (2001). ActiveMath: A web-based learning environment. *International Journal of Artificial Intelligence in Education, 12*(4), 385-407.

Melis, E., Andres, E., Budenbender, J., Frischauf, A., Goduadze, G., Libbrecht, P., et al. (2001). ActiveMath: A Generic and Adaptive Web-Based Learning Environment *International Journal of Artificial Intelligence in Education, 12*, 385-407.

Méndez, J. A., Lorenzo, C., Acosta, L., Torres, S., & González, E. (2006). A web-based tool for control engineering teaching. [10.1002/cae.20080]. *Computer Applications in Engineering Education, 14*(3), 178-187.

Merat, F. L., & Dukki, C. (1997, 5-8 Nov 1997). *World Wide Web approach to teaching microprocessors.* Paper presented at the Frontiers in Education Conference, 1997. 27th Annual Conference. 'Teaching and Learning in an Era of Change'. Proceedings.

Millen, D. R., Feinberg, J., & Kerr, B. (2006). *Dogear: Social bookmarking in the enterprise.* Paper presented at the Proceedings of the SIGCHI conference on Human Factors in computing systems.

Mitrovic, A., & Martin, B. (2007). Evaluating the Effect of Open Student Models on Self-Assessment. *International Journal of Artificial Intelligence in Education, 17*(2), 121-144.

Olston, C., & Chi, E. H. (2003). ScentTrails: Integrating browsing and searching on the Web. *ACM Transactions on Computer-Human Interaction, 10*(3), 177-197.

Ostergren, M., Hemsley, J., Belarde-Lewis, M., & Walker, S. (2011). *A vision for information visualization in information science*. Paper presented at the Proceedings of the 2011 iConference.
Pashler, H., Bain, P., Bottge, B., Graesser, A., Koedinger, K., Mcdaniel, M., et al. (2007). *Organizing instruction and study to improve student learning (NCER 2007-2004)*. Retrieved from http://ncer.ed.gov.
Pintrich, P. R. (1999). The role of motivation in promoting and sustaining self-regulated learning. [doi: 10.1016/S0883-0355(99)00015-4]. *International Journal of Educational Research, 31*(6), 459-470.
Pintrich, P. R., & de Groot, E. V. (1990). Motivational and self-regulated learning components of classroom academic performance. [doi:10.1037/0022-0663.82.1.33]. *Journal of Educational Psychology, 82*(1), 33-40.
Pirolli, P. L., & Anderson, J. R. (1985). The role of learning from examples in the acquisition of recursive programming skills. *Canadian Journal of Psychology, 39*, 240-272.
Rao, R., & Card, S. K. (1994). *The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information*. Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence.
Renkl, A. (1997). Learning from worked-out examples: A study on individual differences. [doi: 10.1016/S0364-0213(99)80017-2]. *Cognitive Science, 21*(1), 1-29.
Shepherd, M. M., Briggs, R. O., Reinig, B. A., Yen, J., & Jay F. Nunamaker, J. (1995). Invoking social comparison to improve electronic brainstorming: beyond anonymity. *J. Manage. Inf. Syst., 12*(3), 155-170.
Shneiderman, B. (1996). *The eyes have it: A task by data type taxonomy for information visualizations.* Paper presented at the Symposium on Visual Languages, Washington D.C.
Shneiderman, B. (2004). Treemaps for Space Constrained Visualization of Hierarchies: an historical summary of Treemap research and applications. from http://www.cs.umd.edu/hcil/treemaps/
Sigala, M. (2007). Integrating Web 2.0 in e-learning environments: a socio-technical approach. International Journal of Knowledge and Learning. *International Journal of Knowledge and Learning, 3*(6), 628–648.
Skold, M. (2008). Social Network Visualization. [Master Thesis]. *Master Thesis. Royal Institute of Technology.* .
Sosnovsky, S., & Brusilovsky, P. (2005, July 26, 2005). *Layered Evaluation of Topic-Based Adaptation to Student Knowledge.* Paper presented at the Fourth Workshop on the Evaluation of Adaptive Systems at 10th International User Modeling Conference, UM 2005.
Svensson, M., H\, K., \#246, Laaksolahti, J., & Waern, A. (2001). *Social navigation of food recipes*. Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems.
Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. [doi: 10.1016/0364-0213(88)90023-7]. *Cognitive Science, 12*(2), 257-285.
Sweller, J., van Merrienboer, J., & Paas, F. (1998). Cognitive Architecture and Instructional Design. *Educational Psychology Review, 10*(3), 251-296.
Titus, A. P., Martin, L. W., & Beichner, R. J. (1998). Web-based testing in physics education: methods and opportunities. *Comput. Phys., 12*(2), 117-123.
Tufte, E. R. (1990). *Envisioning Information*. Cheshire, CT: Graphics Press.

van Merriënboer, J. J. G., & Sweller, J. (2005). Cognitive Load Theory and Complex Learning: Recent Developments and Future Directions. *Educational Psychology Review, 17*(2), 147-177.
Vassileva, J. (2008). Toward Social Learning Enviroments. *IEEE Transaction on Learning Technologies, 1*(4), 199-214.
Vassileva, J., & Sun, L. (2007). Using Community Visualization to Stimulate Participation in Online Communities. e-Service Journal. *Special Issue on Groupware, 6*(1), 3-40.
Vassileva, J., & Sun, L. (2008). Evolving a Social Visualization Design Aimed at Increasing Participation in a Class-Based Online Community. *International Journal of Cooperative Information Systems (IJCIS),, 17*(4), 443-466.
Verginis, I., Gouli, E., Gogoulou, A., & Grigoriadou, M. (2011). Guiding Learners into Reengagement through the SCALE Environment: An Empirical Study. *Learning Technologies, IEEE Transactions on, 4*(3), 275-290.
Veroff, J. (1969). *Social comparison and the development of achievement motivation.* New York: Sage.
Viegas, F. B., & Donath, J. (2004). Social network visualization: Can we go beyond the graph. *Workshop on Social Networks, CSCW, 4*, 6-10.
Vygotsky, L. S. (1978). *Mind and society: The development of higher mental processes*. Cambridge, MA: Harvard University Press.
Wang, H.-C., Li, T.-Y., & Chang, C.-Y. (2004, 30 Aug.-1 Sept. 2004). *Adaptive presentation for effective Web-based learning of 3D content.* Paper presented at the Advanced Learning Technologies, 2004. Proceedings. IEEE International Conference on.
Wattenberg, M. (1999). *Visualizing the stock market*. Paper presented at the CHI '99 extended abstracts on Human factors in computing systems.
Weber, G. (1996). Individual selection of examples in an intelligent learning environment. *Journal of Artificial Intelligence in Education, 7*(1), 3-31.
Weber, G., & Brusilovsky, P. (2001). ELM-ART: An adaptive versatile system for Web-based instruction. *International Journal of Artificial Intelligence in Education, 12*(4), 351-384.
Wexelblat, A., & Maes, P. (1999). *Footprints: history-rich tools for information foraging*. Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit.
Williams, G. C., Bialac, R., & Liu, Y. (2006). Using online self-assessment in introductory programming classes. *J. Comput. Small Coll., 22*(2), 115-122.
Windhager, F., Zenk, L., & Federico, P. (2011). Visual Enterprise Network Analytics - Visualizing Organizational Change. *Procedia - Social and Behavioral Sciences, 22*(0), 59-68.
Yudelson, M., & Brusilovsky, P. (2005, July 18-22, 2005). *NavEx: Providing Navigation Support for Adaptive Browsing of Annotated Code Examples.* Paper presented at the 12th International Conference on Artificial Intelligence in Education, AI-Ed'2005, Amsterdam, the Netherlands.
Zapata-Rivera, J.-D., & Greer, J. E. (2000). *Inspecting and Visualizing Distributed Bayesian Student Models.* Paper presented at the 5th International conference Intelligent Tutoring Systtems.
Zimmerman, B. J. (1990). Self-Regulated Learning and Academic Achievement: An Overview. [doi: 10.1207/s15326985ep2501_2]. *Educational Psychologist, 25*(1), 3-17.