QuizVIBE: Accessing Educational Objects with Adaptive Relevance-Based Visualization

Jae-wook Ahn, Peter Brusilovsky, Sergey Sosnovsky
School of Information Sciences
University of Pittsburgh
Pittsburgh PA 15260
jaa38@pitt.edu, peterb@pitt.edu, sas15@pitt.edu

Abstract: A number of research teams are working to structure personalized access to the online repositories of educational resources. The goal of personalized access is to help students locate resources that match their individual goals, interests, and current knowledge. The project presented in this paper is focused on the least explored branch of personalized access research – adaptive visualization. Here, we present the QuizVIBE visualization system, which provides personalized access to a repository of parameterized questions for evaluating programming knowledge. The system combines spatial, relevance-based visualization with adaptive annotation of the resources.

1. Introduction

Due to the exponential growth of educational resources on the Web, there exists a great vacuum for effective tools to facilitate student access to relevant information. Users urgently need these tools in order to find high-quality resources that will match their individual goals, interests, and current knowledge. This problem has been partially solved by Web search engines (Brin & Page, 1998) and digital libraries search services (Lagoze et al., 2002). However, the resources they locate have distinct presentation styles, target audiences, and coverage. In order to fulfill student's needs within the scope of these abundant and varied resources, personalized information access has been introduced as a solution. Despite recognition of the need for personalized information access, the educational domain has been one of the areas where this approach has thus far lacked success. In that domain, users differ in their interests, goals, skills, knowledge, and learning styles, so the importance of personalization cannot be underestimated.

An example of the need for personalized access as well as a possible solution has been provided by our QuizPACK (Quizzes for Parameterized Assessment of C Knowledge) and QuizGuide projects. QuizPACK is an open corpus Web-based system for authoring and delivery of parameterized quizzes for programming-related courses (Sosnovsky, Shcherbinina & Brusilovsky, 2003). When we crossed the 100-question level in our bank of parameterized self-assessment questions, we realized that it would be a real challenge for a student at any stage of her learning to select the set of most appropriate questions – those that matched her educational goals, level of understanding, and interests. To help students select the most relevant self-assessment quizzes, we developed QuizGuide, an adaptive hypermedia system (Brusilovsky, Sosnovsky & Shcherbinina, 2004). QuizGuide provides personalization with its adaptive annotation technology, displaying to the student their current knowledge level for each course topic as well as the current personal relevance of each of these topics. Personalization is created with a user model server, called CUMULATE, stores the application usage transaction data and infers various user features from this data (Brin & Page, 1998).

Figure 1 shows the QuizGuide interface, which is comprised of two parts, the quiz navigation area and the quiz presentation area. The quiz navigation area on the left side provides a list of topics that are linked to quizzes, each consisting of a set of questions. The icons next to the topic titles represent the student's knowledge level and relevance of the topics to their current learning goal.

Even though QuizGuide has been successful in helping users find relevant resources that match their current knowledge level, the ordered list provided by QuizGuide does not give students the ability to access relevant resources according to their specific interests. Learning goals often include several concepts and students sometimes face situations where they must distinguish between questions based on similarity to multiple concepts. For instance, students might wish to avoid concepts they have already learned, while pursuing multiple new concepts. This lack of distinction can be an obstacle for students who wish to fully achieve their learning goals.

In this paper, we present QuizVIBE, an adaptive visualization system, which combines relevance-based document visualization and adaptive annotation. It can provide users with information about the relationships between questions (documents) and concepts (POIs) according to their similarity values. The concepts are represented as Point of Interest (POIs) while the questions are laid out by their similarities to the POI so that users can easily understand which question is more related to a certain concept and can thus be guided to access questions relevant to their learning goals. It can also provide adaptive annotation information which is shared with QuizGuide. Section 2 describes the user interface of QuizVIBE which gives adaptive access to questions. Section 3 describes the system visualization mechanism and section 4 presents the question annotation mechanism. Section 5 concludes this paper with a summary of the current stage of our work and its future outlook.

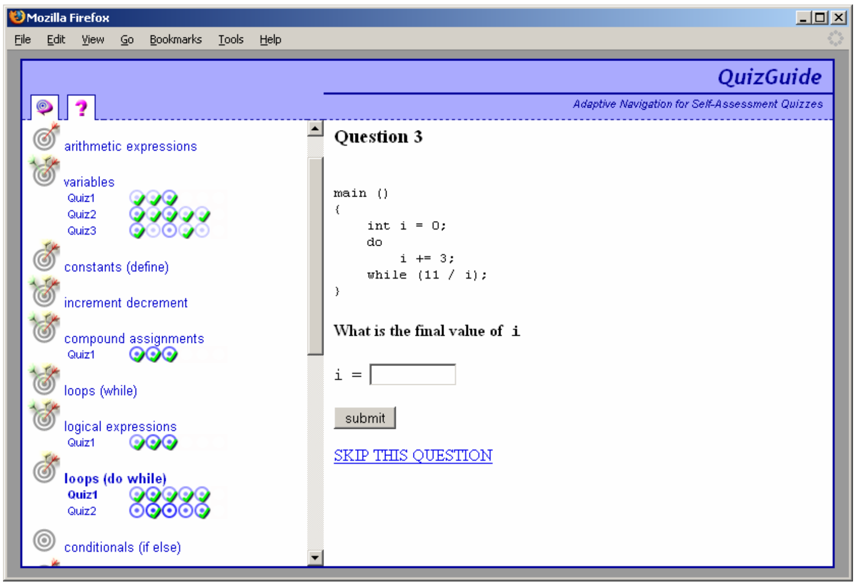


Figure 1: The QuizGuide interface

2. QuizVIBE Interface

VIBE (Visual Information Browsing Environment) is a document visualization tool which was originally developed in Molde College in Norway and the School of Information Sciences at the University of Pittsburgh (Olsen & Korfhage, 1994). Since its first development, various extensions have been introduced. WebVIBE (Morse & Lewis, 1997) incorporated the metaphor of a naive view of magnetism, VR-VIBE (Benford et al., 1995) extended the original VIBE toward a more interactive 3D display, and BIBE (Biological Information Browsing Environment) (Heidorn, 2001) implemented the VIBE framework for biological domains. We implemented and improved the VIBE framework in terms of educational resources and incorporated it as a component for the ADVISE (Adaptive Document VISualization for Education) visualization suite. ADVISE is a collection of Web-based document visualization applications for education. It includes the spring force algorithm-based ADVISE 2D, the 3D document visualization tool ADVISE 3D, and ADVISE VIBE. The QuizVIBE system presented in this paper was implemented using ADVISE VIBE. Like other ADVISE tools, ADVISE VIBE provides an adaptive visualization of a set of learning objects (documents) and allows users to explore them. The adaptive and interactive nature of visualization is aimed at helping the users select the most relevant object. Discovering an object of interest, one can double-click on its icon to open it. Unique to ADVISE VIBE is the nature of the supported visualizations and interactive exploration.

Figure 2 shows the QuizVIBE visualization helping the student to select the most appropriate self-assessment questions for Lecture 9. The lecture includes 12 concepts such as “int,” “printf,” and “add,” which serve as POI, represented with small square icons. The QuizGuide repository includes 8 relevant questions, which are visualized with target icons. The system arranges the questions by their relative similarity to the visible POIs. The document (question) is considered similar to a POI (a domain concept) if it’s connected (indexed) with the corresponding concept. The weight of this connection determines the level of similarity. The more similar the document is to a certain POI, the closer the document is positioned to that POI. For example, on Figure 2, the questions are positioned on the lower left side of the screen. This means that they are more similar and related to such concepts as “variable_declaration,” “variable_initialization,” “add,” and “simple_assignment” than to the concepts “char” and “include.”

The target icons representing individual questions dynamically change their color intensity to reflect the user’s individual progress and knowledge level. A lighter color (lower intensity) means that the current user’s knowledge level is higher while a deeper color (higher intensity) means they have a lower knowledge level. Such color coding attracts attention to the questions with a higher number of unmastered concepts. Some icons also embed green check marks, which represent the progress of the user (Figure 5). A detailed description of the adaptive knowledge-based and progress-based annotation of questions is given in section 4.

In addition to the target icon, each question is annotated with a bar icon. The bar represents the cumulative difficulty of the question, which is calculated by averaging question error rates of all students (Figure 2). We have adopted two visual clues for this single icon, length and color. It presents a corresponding part of the blue-to-red color spectrum, where only blue means lower difficulty and whereas the addition of red means higher difficulty. Therefore, if a question is very difficult, the length of the spectrum icon grows to the maximum and shows the entire spectrum. Accordingly, when a question is very easy, the icon shrinks to show only the small blue portion of the spectrum.

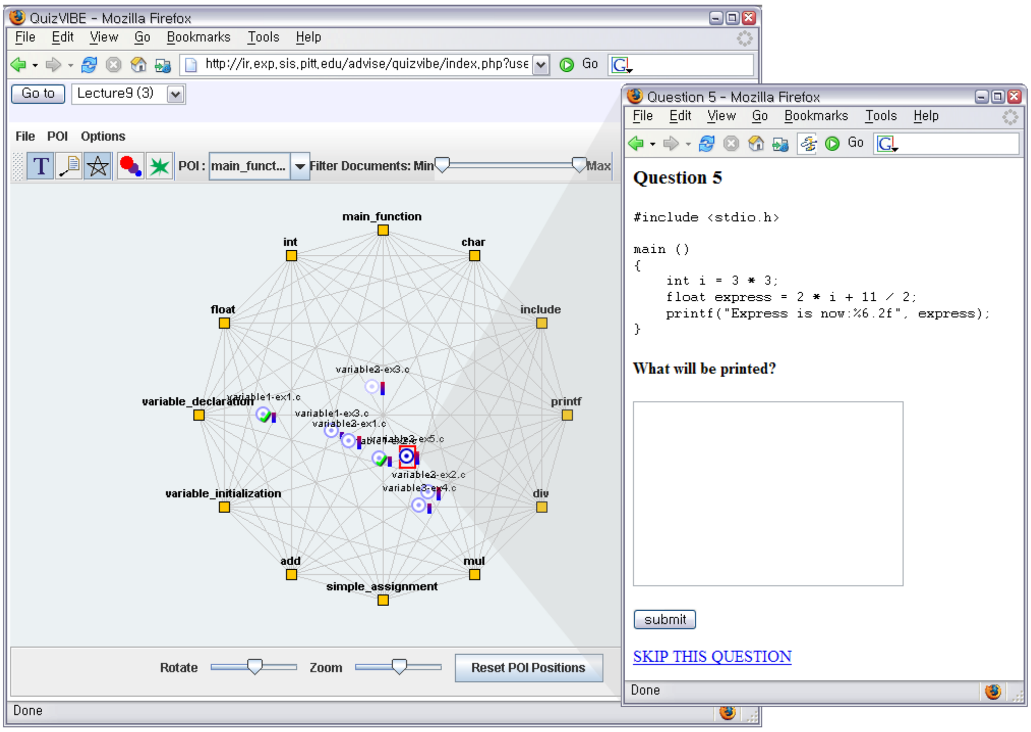


Figure 2: A QuizVIBE visualization with one opened question.

QuizVIBE provides a highly interactive environment to explore questions, concepts, and their relationships. Along with such common tools as zooming, panning, and viewpoint rotation (which are supported by all ADVISE systems), QuizVIBE supports several specific relevance-based exploration methods. Most important is the ability to move and re-arrange the POIs. When the user moves a POI (using a mouse), the system instantly re-positions all documents, maintaining an accurate representation of their similarities to the POI's. On the other hand, when the user moves a POI, she can observe how related documents follow the move. The distances they travel reveals very obviously their similarity to the re-positioned POI. If one document is more similar to the POI than another, it will travel a greater distance. VIBE provides an option to show the trails of these movements. In the example shown on Figure 3, a user enabled the trail option then moved the POI "float" to the upper left side of the screen. This caused 6 questions to follow the POI, because they are related to this concept. The trailing lines they made have different lengths, according to the magnitudes of their similarities to the moved POI. Therefore, the user can compare the lengths of the trails to see how each document is related to the POI. The longer the trail is, the more the document has moved and thus the more similar it is to the POI. In Figure 3, the leftmost question has the longest trailing line and therefore we can assume that it is the most related to the concept "float."

Figure 4a shows three additional similarity-based tools: *Similarity*, *Radar*, and *Filtering*. All three are based on the similarity values between questions and visible POIs. When the *Similarity* tool is enabled, pointing to a question (for example, "variable2-ex3.c," as shown in Figure 4a) causes a disc icon to appear next to each POI representing the strength of the POI's similarity to the selected question. The discs are color-coded, with red representing a higher similarity and blue representing a lower similarity. The diameter of the discs also indicates different levels of similarity, with a bigger disc representing a higher similarity. In the example shown on Figure 4a, the most similar concept to the current question is "float" on the upper left side of the screen and the most dissimilar concept is "main_function," which has the smallest disc.

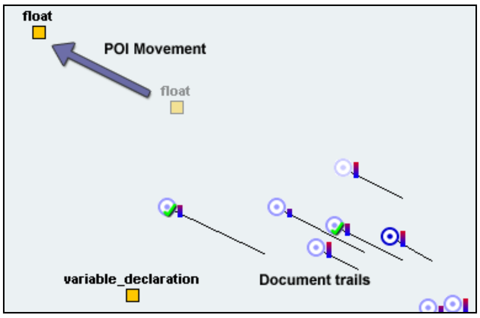
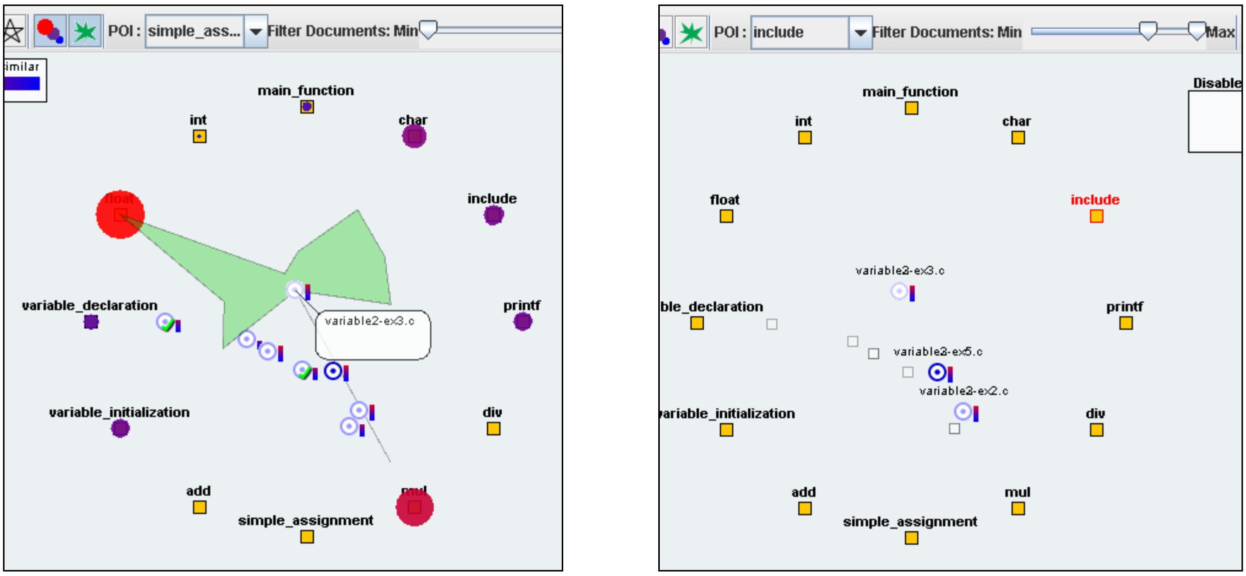


Figure 3: Document trails



a) b)

Figure 4: QuizVIBE visualization aid functionality
a) The *Show Similarity* and *Radar* options; **b)** Filtering out irrelevant questions

Radar is a variation of the *Star* option of the original VIBE, which showed the influence of POIs to a document by the length of lines from the document icon to the POIs (Olsen & Korfhage, 1994). The *Radar* graph is created by connecting the end points of those lines and filling the geometric figure with a translucent green color. The *Radar* or *Star* option lets users estimate the relative magnitudes of similarities between a document and multiple POIs, which may be not easy when only using the "discs" in the *Show Similarity* option. In Figure 4a, we can assume that the similarity between the current question and the POI "variable_initialization" (lower left) might be around 0.5, judging by the length of the corresponding line on the Radar graph. With these functionalities, users can more easily perceive the relationships between questions and POIs, even without moving POIs around on the screen.

Relevance filtering allows the exclusion of questions with a low similarity to the chosen POI. In Figure 4b, the POI "include" is currently selected. By manipulating the double slider, the lower and upper thresholds of relevance have been set (around 0.6 and 1.0 respectively). The questions with similarity to "include" that is smaller than the lower threshold or greater than the upper threshold have been filtered out and shrunk to small square icons with no titles. With the help of this function, the screen is much easier to read, and it is easier to locate questions with the required similarity to the concept of interest.

3. QuizVIBE Visualization Mechanism

3.1. Document Positioning

As mentioned above, VIBE includes two main elements, POIs (Points Of Interest) and documents. Each POI is a reference point and can be understood as a keyword or a concept which may be of interest to the user. Documents are actual resources to be explored. VIBE places the POIs and documents on a two-dimensional space so that users can understand the relationships between each document and the POIs. Users can get visual clues before they actually locate and access documents they need. When a user chooses and places icons arbitrarily to represent the POIs on a screen, the positions of every document are automatically calculated and moved on the screen, so that similar documents are positioned closer to the POIs.

VIBE focuses on the similarity ratios between documents and POIs for this task. Each document is located at a position representing the ratios of its similarities to each of the reference points, the POIs. Each POI is given keywords which are used for calculating the influence or the similarity value between the POI and a document. Below is a brief description of the algorithm by which VIBE places the documents (Olsen et al., 1993).

(1) If a document is influenced by one POI only, its icon is positioned on top of this POI.

(2) If a document is influenced by two POIs, the position of this document will be on the line between these POI's, closer to the one that has the higher similarity score.

(3) If a document is influenced by more than two POIs, a temporary position is determined between the first two POIs as in step (2), and then a new position between the temporary position and the next POI is determined in the same manner as in step (2).

The influence is measured by the similarity of each document to the POIs. For example, if a document has similarity 0.3 and 0.6 to POI p_a and p_b respectively, the similarity ratio to these two POIs is 1:2 and the document is placed at a one-third position from p_b on the line connecting those two POIs, because it is twice as similar to p_b than p_a. Positions of all the documents to be displayed are calculated separately, in this way.

3.2. Document Representation and Similarity Calculation

VIBE can visualize any document as long as similarity values for the observable POIs are provided. In QuizVIBE, each question is understood as a document and concepts are represented as POIs. Therefore, we adopted the vector space model (Salton, 1989) in order to calculate the document to POI (question-to-concept) similarities. In this model, documents are represented as term vectors and each component contains the importance or the weight of the term. Likewise, questions are understood as concept vectors and their components stored the importance or the weight of the concepts.

Currently, QuizVIBE includes 50 concepts and 171 questions, hence the whole corpus can be converted to a matrix with 171 rows (questions) and 50 columns (concepts).

The importance of each of the concepts is expressed as the well-known TF-IDF (Term Frequency times Inverse Document Frequency). TF is the number of occurrences of a term in a document, or frequency of a concept in a question in this case. IDF is an inverse of DF (Document Frequency), which means the number of documents which have the term in question. Here, it is the inverse of the number of questions which have the corresponding concept. IDF can decrease the effect of very common terms, which appear in many documents and thus have greater DF values. The importance of a term in a document increases with the frequency of the term in the document and decreases with the number of documents containing the term (Korfhage, 1997). Therefore, if a concept does not appear in a question, the corresponding component in a vector is filled with 0s and if it appears at least once, the component is filled with TF-IDF weights.

POIs are represented as concept vectors. There is no great difference between them and question vectors except that they have binary (1 or 0) weights for concepts they represent. For example, if a POI represents a concept "while," all the components of its concept vector are filled with 0s except for the one component for concept "while," which is given the weight 1.

When every question and concept vector is ready, the question-to-concept vector similarity values can be calculated. The cosine similarity coefficient was used for this calculation. It is the cosine angle between two vectors and ranges from 0 to 1. If two vectors are completely similar, they have a similarity coefficient of 1 and if they are completely dissimilar their cosine similarity coefficient value is 0.

After this calculation, each question will have similarity values for all concepts (POIs) which are then fed into the VIBE visualization module, determining the relative positions of questions to concepts placed on the screen.

4. Questions and their Annotations in QuizVIBE

The actual learning objects being accessed by students in QuizVIBE are questions served by the QuizPACK system. The architecture and evaluation of this system have been reported in a number of papers (see, for example, Sosnovsky, Shcherbinina & Brusilovsky, 2003). In the current section, we briefly summarize its most important features with respect to the learning content.

Every question in QuizPACK presents a simple C program (see Figure 2 on the right side). To answer the question, students must analyze the program and predict either the final value of a target variable or an output printed by the program. After an answer is submitted, QuizPACK generates feedback indicating whether the question has been answered correctly or not. If incorrect, the right answer is provided.

The most important property of QuizPACK is the parameterized nature of its questions. Every time a question is accessed, one of the numerical parameters in the text of the question is dynamically instantiated with a random value within preset limits. In the case of a formal assessment, this gives the teacher the opportunity to create cheating-proof questions; and if a student uses QuizPACK as a self-assessment tool, this feature provides her/him with a chance to perform a more interesting, but focused study on specific concepts – by changing the parameters used every time the same question is asked.

4.1. Indexing

Every question in QuizVIBE is associated with a set of concepts, which are used in two ways. First, they serve as points of interest in the visualization layout, and second, they provide units to measure student progress within the system. This section briefly describes the approach we use for indexing questions with concepts; to learn more about it, see Brusilovsky et al., 2005. The indexing algorithm we developed consists of two stages. In the first stage we perform the automatic extraction of programming concepts from the question text. In the second stage, concepts are assigned to the roles they play in the questions: prerequisite or outcome.

4.1.1. Automated Concept Extraction

Traditionally, the automatic extraction of grammatically meaningful structures from textual content and the determination of concepts on that basis is a task for the special class of programs called parsers. In our case, we have developed a parsing component with the help of two well-known UNIX utilities: lex and yacc. This component processes the source code of a C program and generates a list of concepts used in the program. Each programming structure in the parsed content is indexed by one or more concepts, depending upon the amount of knowledge students need to have learned in order to understand the structure. It is necessary to mention that each concept represents not simply a keyword found in the code, but a grammatically complete programming structure.

Currently, about 50 concepts can be determined by the parser. This subset is based on the ontology of C programming we developed (Sosnovsky & Gavrilova, 2005). The total number of concepts in this ontology is 245. Only 185 of them are terminal concepts available for indexing. To facilitate the indexing of questions with concepts that are not available for automatic extraction, we developed an authoring interface, where the results of automatic indexing can be corrected or enriched manually (Brusilovsky et al., 2005).

4.1.2. Prerequisite/Outcome Identification

The results of the concept extraction stage are indexed lists for all questions. However, it is not enough for our purposes, since these lists are not yet connected to each other and the content does not yet possess any structure to provide a basis for the reliable inference of adequate, adaptive interventions. In the next stage, all concepts related to each question are divided into prerequisite and outcome concepts. Prerequisites are the concepts that students need to master before starting to work with each current question. Outcomes denote the learning goals of the question.

We use an original algorithm for the automatic identification of outcome concepts. This algorithm is flexible enough to be influenced in an instructor-specific way by the teacher of any course. The source of knowledge for this algorithm is a sequence of groups of questions. Each group is formed by gathering questions introduced in the same lecture, which thus share the same learning goal. Groups are chronologically ordered, based on course lectures, thus forming a sequential structure of learning goals for the course. The prerequisite/outcome division algorithm is based on the following assumptions:

(1) While analyzing a question from some lecture, concepts corresponding to this question but introduced in a preceding lecture are considered to be previously learned.
(2) In each question, all concepts introduced in the previous lectures are considered to be prerequisites, while the concepts first introduced in the current lecture are regarded as outcomes.
(3) The set of new concepts found in all questions associated with the lecture are considered to be the learning goal of the lecture.

The direct result of this algorithm is a fully-indexed set of content elements belonging to each lecture and a sequence of learning goals associated with the lectures.

4.2. The Adaptive Annotation of Questions

Whenever a student submits a correct answer to a question, her/his user model is updated for the concepts involved in this question. To help students realize their strengths and weaknesses and choose the appropriate question, we combined a VIBE visualization with adaptive annotation, a technique known from the field of adaptive hypermedia. Every document (question) on the layout is adaptively annotated with an icon reflecting two kinds of information:

(1) Progress made by all students on that specific question

(2) The system's view about the current student's knowledge, in regards to the outcome concepts of the specific question

Figure 5 summarizes the adaptive question annotation of QuizVIBE. The next two sections provide detailed explanation.

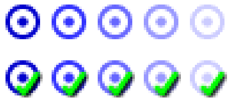


Figure 5: Adaptive annotation by color and shade

4.2.1. Progress-Based Annotation

Question progress is a binary entity. It reflects the fact that the specific question has been solved correctly at least once. The system does not differentiate between novel questions and questions which have been answered incorrectly. As soon as a student submits his first correct answer to a question, the appropriate icon receives a checkmark. This can help students choose between similar questions which are characterized by a highly intersected set of concepts. If one of these intersected questions has a checkmark, and the second does not, a student may still be interested in testing her/his knowledge of that set of concepts, as covered by these questions, and will thus be guided to choose the unanswered question over the passed one.

4.2.2. Knowledge-Based Annotation

QuizVIBE expresses its beliefs about student knowledge about a certain concept by attracting student attention to the questions where this concept plays an outcome role. The color of small target icons, representing questions, encodes the level of focus a student should have on these questions. The less cumulative knowledge a student has towards the outcomes of the question, the more intense the color of the corresponding icon. There are five levels (and correspondingly four thresholds to switch between these levels) the color may range from deep blue to very light bluish grey. When the low intensity of a target matches the color of the last level (lighter blue), the faded target is guiding the student to chose another question.

It is possible that a question which has been never accessed (or answered correctly) by a student may have the cumulative knowledge of the top level. The reason for such a scenario would be that the student increased her/his level of knowledge for all the concepts included in the question by working with other questions that include the same concepts. In this case, the question is annotated with a light blue target without a checkmark.

Since the roles the concepts play in questions depend on the learning goal the questions belong to, the described color-coding approach is also goal-related. Student knowledge of concepts will influence the annotation of only those questions where concepts act as outcomes. For example, the knowledge of the concepts "while loop" will influence the color of questions about the while loop, that is, those questions which use the while loop and belong to the learning goal "simple loops." At the same time, knowledge of the "while loop" will not change the color of a question using the "while loop" if it is associated with, say, the learning goal "arrays." However, positive answers on this "arrays"-question will change the systems record of the student's knowledge of the concept "while" and consequently the annotation generated for the "while"-questions. Hence, while students are working with higher learning goals, they implicitly increase their knowledge of prerequisite concepts, which is captured by the system.

4.2.3 Cross-Application Interaction

It is necessary to mention that the questions accessible from QuizVIBE can also be accessed from two other systems. Students are able to take QuizPACK quizzes assigned to a specific lecture on the Learning Portal. In this case, no navigational support is available for them. They are not provided with the information on the concepts involved in a question, progress they have made for a question or knowledge levels they have achieved for different concepts. Another way to access it is to use the QuizGuide system (Brusilovsky, Sosnovsky & Shcherbinina, 2004) which also applies adaptive annotation when navigating students through the same sets of questions. QuizVIBE inherits its colored target icons from QuizGuide, which has already been evaluated by students in a real class environment.

Since all three systems basically share the same content, all actions that a student performs in one system will influence the adaptive behavior of other systems (QuizPACK is not adaptive, therefore it can only act as a donor of activity rather than a receiver of cross-application adaptive interventions). For example, a student might submit an answer to a question accessed from QuizGuide. This action is recorded in the central user model. Afterwards, whenever he or she logins into QuizVIBE, the system requests the user model about the student, which contains knowledge and progress calculations contributed from both QuizGuide actions and QuizVIBE attempts.

5. Conclusions

In this paper, we implemented an adaptive quiz visualization tool, QuizVIBE, which supports adaptive annotation and POI-based VIBE visualization for Web-based open corpus educational resources. It can help students effectively find relevant questions based on information about their knowledge level, learning goal, and similarities between questions and concepts. It also supports various visualization features which can more efficiently guide users to achieve their goals. These features include zooming/rotating/panning/filtering, color-coding, icons, similarity clues, and a radar graph.

QuizVIBE can already track and log most user activity, such as opening documents, dragging POIs, zooming/rotating/panning the canvas, filtering documents, etc. However, since it was developed very recently, this user activity log data is not aggregated for evaluation purposes. Our future plans include using log information data to evaluate the system, as well as the addition of more features to effectively guide users to access relevant resources.

References

Benford, S., Snowdon, D., Greenhalgh, C., Ingram, R., Knox, I., and Brown C. (1995) VR-VIBE: A Visual Environment for Co-operative Information Retrieval. *Eurographics '95*, 349-360.

Brin, S., and Page, L. (1998) Anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International World Wide Web Conference* (Brisbane, Australia, Apr. 14 –18), 107–117.

Brusilovsky, P. (2004) KnowledgeTree: A distributed architecture for adaptive e-learning. In *Proceedings of The Thirteenth International World Wide Web Conference, WWW 2004* (Alternate track papers and posters), New York, NY, 17-22 May, 2004, ACM Press, 104-113.

Brusilovsky P., Sosnovsky S., and Shcherbinina O. (2004) QuizGuide: Increasing the Educational Value of Individualized Self-Assessment Quizzes with Adaptive Navigation Support. In Janice Nall and Robby Robson (eds.) *Proceedings of E-Learn 2004*, Washington, DC, USA: AACE, 1806-1813.

Brusilovsky, P., Sosnovsky, S., Yudelson, M., and Chavan, G. (2005) Interactive Authoring Support for Adaptive Educational Systems. In: C.-K. Looi, G. McCalla, B. Bredeweg and J. Breuker (eds.) *Proceedings of 12th International Conference on Artificial Intelligence in Education, AIED'2005* (Amsterdam, July 18-22, 2005). Amsterdam: IOS Press, 96-103.

Heidorn, P.B. (2001) A Tool for Multipurpose Use of Online Flora and Fauna: The Biological Information Browsing Environment (BIBE). *First Monday*. Volume 6(2).

Korfhage, R. R. (1997) *Information Storage and Retrieval*. Wiley Computer Publishing, New York, NY.

Lagoze, C., Arms, W.Y., Gan, S., Hillmann, D., Ingram, C., Krafft D.B., Marisa, R.J., Phipps, J., Saylor, J., Terrizzi, C., Hoehn, W., Millman, D., Allan, J., Guzman-Lara, S., and Kalt, T. (2002) Core services in the architecture of the national science digital library (NSDL), In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL 2002)*, 201-209.

Morse, E.L., and Lewis, M. (1997) Why information retrieval visualizations sometimes fail. *Proceedings of IEEE International Conferences on Systems Man and Cybernetics*, Orlando, FL, 12-15 October, 1997, 1680-1685.

Olsen, K.A., and Korfhage, R.R. (1994) Desktop visualization. In *Proceedings 1994 IEEE symposium on visual languages*, St. Louis, MO: IEEE. 2349-244.

Olsen, K.A., Korfhage, R.R., Sochats, K.M., Spring, M.B., and Williams, J.G. (1993) Visualization of a document collection: The vibe system. I*nformation Processing and Management* 29(1), 69-81.

Salton, G. (1989) *Automatic Text Processing*. Addison-Wesley Publishing Co., Reading, MA.

Sosnovsky S., and Gavrilova T. (2005) Development of Educational Ontology for C-Programming, In *Proceedings of International Conference "Knowledge-Dialogue-Solution" (KDS-2005)*, Varna, Bulgaria, June 20-24, 127-131.

Sosnovsky, S., Shcherbinina, O., and Brusilovsky, P. (2003) Web-based parameterized questions as a tool for learning. In A. Rossett (ed.) *Proceedings of World Conference on E-Learning, E-Learn 2003*, Phoenix, AZ, USA, November 7-11, 2003, AACE, 309-316.