A Comparative Study of Visual Cues for Annotation-Based Navigation Support in Adaptive Educational Hypermedia

Roya Hosseini
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA 15260
roh38@pitt.edu

Peter Brusilovsky
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA 15260
peterb@pitt.edu

ABSTRACT

Adaptive link annotation is one of the most well-known adaptive navigation support technologies that aims to guide hypermedia users to the most relevant information by personalizing the appearance of hyperlinks. Past work assumed no difference between different interface implementations of personalization approaches that are conceptually the same. The goal of the current study was to determine whether the choice of visual cues does matter by conducting a user study with several alternative designs for link annotation in interactive code examples.

Keywords

adaptive navigation support; link annotation; code examples

1. INTRODUCTION

Adaptive navigation support is a group of core technologies for adaptive hypermedia [1]. The idea of adaptive navigation support (ANS) is to guide hypermedia users to the most relevant information by personalizing the appearance of hyperlinks on every page that the user visits. Arguably, the most popular and the most explored among ANS technologies is adaptive link annotation, which augments hypertext links with dynamic and personalized visual cues [2]. In the early days of adaptive hypermedia, research on adaptive link annotation focused on altering the text anchor (such as changing the style, size, or the color of the link's font). However, more recent projects have explored various ways to augment links with meaningful icons. Icon-based link annotation allows the nature of personalization to be expressed more clearly while avoiding any negative impact on overall link readability.

Over the years, some efficient ANS approaches have been established and evaluated by different teams. Moreover, many teams have suggested different sets of icons to implement conceptually the same personalization approach (such as knowledge-based or prerequisite-based annotations). For example, to show the amount of knowledge on a topic, INSPIRE [11] and NavEx [4] used a fillable shape (Figure 1); Progressor [9] used a color gradient from red (poor knowledge) to green (good knowledge) (Figure 2); and Mastery Grids [10] used a green color of different intensities (light for little knowledge, dark for more knowledge) (Figure 3). While each of these ANS approaches was typically evaluated and was proven to be efficient, none of the studies attempted to separate the impact of the specific personalization approaches (for example, showing the amount of knowledge

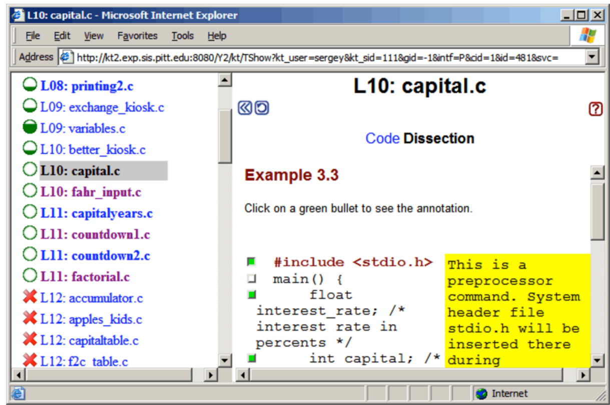

Figure 1: NavEx uses a fillable shape (shown in the left panel) to represent student's knowledge in an example.

Figure 2: Progressor uses a red-to-green color gradient to represent student's knowledge in a topic.

gained by a user on a specific page) from the impact of specific icons used to implement this approach (for example, showing the amount of knowledge with check marks of different size or with an icon of a partially filled glass). It was implicitly assumed that the choice of icons to implement an adaptation approach does not matter, and that only the approach itself does. However, some pioneering studies in the area of personalized interfaces [6, 12] indicated that different interface-level implementations of the same functionality are not equal and might affect users in different ways. In this paper, we present our attempt to compare different implementations of the same ANS approach. The goal of this study was to determine whether the choice of specific icons for ANS affects user perception and overall performance.

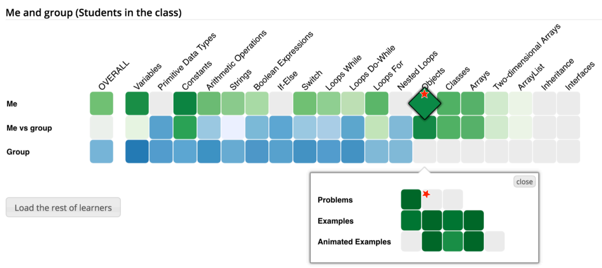

Figure 3: Mastery grids uses different green color intensities (in the first row) to represent student's knowledge in a topic.

2. CONTEXT

The research presented in this paper was motivated by the need to select visual cues for adaptive link annotation in interactive program examples produced by the WebEx system [4]. WebEx program examples are hyperlinked code examples, where each code lines could be linked with text that explains the purpose of the line and/or the results produced when this code is executed. Explanations are usually hidden, which makes the code example look clear, but clicking a code line of interest provides access to the associated explanation. The original WebEx system has no link annotation; however, more recent versions used a simple history-based link annotation: code lines already accessed by the user were annotated with check marks, as shown in Figure 4.

The goal of our current project was to extend WebEx with more advanced knowledge-based link annotation that could guide users to the most appropriate lines by showing how much knowledge about the concepts presented in each line the user has already mastered. Two intelligent technologies make this functionality feasible: the ability to automatically identify programming concepts associated with each line of code using a concept parser [7], and a dynamic student model that maintains the current level of student knowledge for each concept [13]. Using the current level of knowledge for concepts associated with each line, we can calculate how much knowledge associated with each line is already known to the learner. We expected that visualizing this dynamically (i.e., displaying the amount of knowledge as a visual cue next to the line) could help users to select the most important lines. We also wanted to directly recommend the most important lines for the user to explore.

One of the challenges in this process was choosing visual cues to express the amount of knowledge behind each code line. First, as shown in the previous section, past research explored a whole range of approaches to present the current level of knowledge using visual cues, but provided no guidance on how to select the most appropriate approach for a specific target context. Second, we wanted to use three different kinds of visual cues in parallel (one for knowledge-based annotation, one for history-based annotation, and one

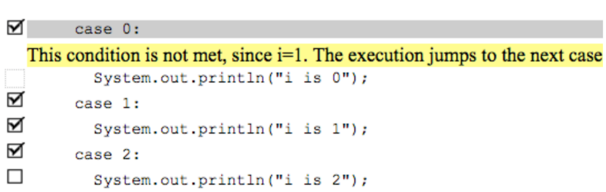

Figure 4: A partial view of an annotated example with a check mark annotation for clicked lines.

to mark recommended lines); however, existing research provided no guidance on how best to combine visual cues.

To resolve this challenge, we decided to run a formal study. The aim of the study was to determine the best knowledge-based annotation approach and find the best way to combine it with the history-based annotation and direct recommendation approaches. The remaining part of the paper presents the candidate approaches that we selected for the study, the organization of the study, and its results.

3. ANNOTATION DESIGN CHOICES

This section discusses design alternatives for icon-based adaptive link annotation in code examples produced by the WebEx system. We review visual cues for showing student knowledge behind each line, lines viewed in the past, and recommended lines.

3.1 Knowledge-Based Annotation

Perhaps one of the most valuable piece of information that can facilitate navigation within an example is making the student aware of how much he or she knows about the programming concepts that are used in different lines of code examples. This could be done by displaying a dynamic icon that expresses the level of student knowledge next to each link. This knowledge-based ANS is one of the most popular in adaptive hypermedia with many designs explored so far. For our study, we selected three previously explored ANS designs for displaying the amount of knowledge.

The first design used a "filling" metaphor, displaying icons with different levels of filling to show the knowledge behind each line. This kind of design was explored in the past in [11, 4]. Five discrete filings were defined, from 0% to 100%, with 25% increments to represent 0% to 100% knowledge behind the line. This design is referred to as $A1$ (see design $A1$ in the *knowledge-based annotation* column of Table 1). The second design ($A2$), explored earlier in [10], used different intensities of the green color. As student knowledge increases, the green color of the icons becomes darker (see design $A2$ in the *knowledge-based annotation* column of Table 1). The third design ($A3$), explored earlier in [9], used a gradient from orange to green colors for the icons, relative to the knowledge of the student. As student knowledge increases, the color of the icon changes from dark orange through yellow into dark green (see design $A3$ in the *knowledge-based annotation* column of Table 1).

3.2 History-Based Annotation

The idea of history-based annotation is to mark links that lead to the already explored parts of hyperspace [5]. In our case, the links in an example can be annotated to show lines that have already been viewed by the student in the past. When the goal is to find new information, it helps to focus on the lines that have not yet been explored. When the goal is to review an example again, it helps to focus on lines that have already been explored. Two designs were explored for the history-based annotation of lines. In each design icons of lines were changed to help student distinguish lines that had been viewed from lines that had not been viewed yet.

The first design ($B1$) borrowed the Web browser design that changes the color of visited links from blue to purple: the icons next to lines that were viewed by the student are filled with a purple color. Since this history-based annotation must be used jointly with knowledge-based annotation,

Table 1: Design alternatives for annotation of links in an annotated example

	Knowledge-based annotation	History-based annotation $B1$	History-based annotation $B2$	Recommendation $C1$	Recommendation $C2$
$A1$					
$A2$					
$A3$					

there were three possible combinations: $B1(A1)$, $B1(A2)$, and $B1(A3)$ shown in column $B1$ of Table 1. The second design ($B2$) followed the approach used in the current version of WebEx (Figure 4): a check mark sign over the bullet indicates the visited lines. Three combinations of this design are presented in column $B2$ of Table 1.

3.3 Recommendations

Students often refer to examples when seeking help with a problem. When they come to an annotated example, they need to locate lines with explanations that could be most helpful in solving that problem. An adaptive system can help by recommending the most useful example lines taking into account the target problem and the state of student knowledge. The typical method of recommending items to a user is to offer them as a ranked list so that the most valuable item is placed on the top. However, this method is not applicable in case of recommending helpful example lines, because the order of example lines cannot be changed. Therefore, we have to mark recommended lines with special visual cues that should be recognizable along with knowledge-based and history-based cues.

Two designs were explored for the recommendation of an example line. The first design ($C1$) simulates bold font used; for example, in [3], by increasing the width of the icon border to indicate recommended lines. The second design $C2$ used a red star as an indicator of recommendation, just as in [8]. Similar to history-based annotation, the recommendation was used with knowledge-based annotation designs $A1$–$A3$. Columns $C1$ and $C2$ of Table 1 illustrate how the knowledge-based annotations and recommendations were combined.

4. THE STUDY

We designed and conducted a user study to assess design alternatives for the three types of icon-based ANSs reviewed above. We recruited one pilot and 31 regular participants who were undergraduate ($n = 8$) and graduate ($n = 23$) students at University of Pittsburgh, mostly from the School of Information Sciences. Each session lasted about 30-45 minutes, and participants were compensated with $10.

The procedure started with presenting the goal of the study. Then, a printout of an annotated example interface was presented to subjects to explain the nature of annotated examples and the idea of adding visual cues to example lines to present information about student knowledge, browsing history, and recommendations. Once the subject declared that the explanations were clear and that he or she was ready for the next step, alternative designs were introduced one by one, starting with three designs for knowledge-based annotation of code lines ($A1$, $A2$, and $A3$), continuing with two designs for history-based annotation ($B1$, $B2$), and ending with two designs for line recommendation ($C1$, $C2$). The designs were shown with the full set of icons for each kind of annotation, as shown in Table 1. After introducing each design, the subject was asked to provide an opinion about *each* design alternative by answering a 5-item questionnaire. Table 2 summarizes three questionnaires that were used for designs $A1$–$A3$, Table 3 summarizes two questionnaires for designs $B1$–$B2$, and Table 4 summarizes two questionnaires for designs $C1$–$C2$. The subject was asked to answer each question by using a five-point scale that ranged from 'Strongly Disagree' to 'Strongly Agree' in questions 1-4, and from 'Very Difficult' to 'Very Easy' in question 5.

After collecting subject's opinion about all of the designs, the subject was asked to perform three tasks:

Task 1 provided three code examples that were annotated according to three different knowledge-based ANS alternatives, i.e., $A1$–$A3$. The subject was asked to find the lines that showed *minimum* and *maximum* knowledge in each example and then she/he had to select the design that made finding the lines with *minimum* and *maximum* knowledge easier. All subjects received the same three representations in a random order.

Task 2 provided three annotated code examples and asked the subject to circle the *already accessed lines*. Each example used a combination of knowledge-based annotations $A1$–$A3$ and history-based annotations $B1$–$B2$, which indicated the accessed lines. In total, six combinations were used for the examples shown in this task: $B1A1$, $B1A2$, $B1A3$, $B2A1$, $B2A2$, $B2A3$. However, to avoid overload, each subject had to work with three of these six combinations. Odd-numbered subjects received combinations $B1A1$, $B1A3$, and $B2A2$, while even-numbered subjects received $B1A2$, $B2A1$, and $B2A3$. Annotated examples were shown to each subject in a random order. At the end of the task, the subject had to select the design that made finding the accessed lines easier.

Task 3 provided three annotated code examples and asked the subject to circle the *recommended lines* in each one. Each example used a combination of knowledge-based annotations $A1$–$A3$, combined with annotations $C1$–$C2$ for showing recommended lines. In total, six combinations were used

Table 2: Knowledge-based annotation questionnaire

1	Being able to see knowledge-based progress using $X/Y/Z$ is useful
2	I think $X/Y/Z$ correctly reflects knowledge-based progress
3	The state of the bullet in designs with $X/Y/Z$ helps me to distinguish lines that I have to pay more attention to
4	I am motivated to click on lines with $U/V/W$, meaning lines that I have less knowledge in them
5	How easy is for you to remember the meaning of $X/Y/Z$?
	X: filled bullets Y: bullets with shades of green color Z: bullets with shades of orange to green color U: lower bullet fillings V: lighter green bullets W: orange or yellow bullets

Table 3: History-based annotation questionnaire

1	Being able to see clicks is useful when X/Y is used for showing clicked lines
2	I think X/Y help me correctly distinguish not clicked lines from clicked lines
3	Using X/Y for clicked lines helps me distinguish lines that I have to pay more attention to them
4	I am motivated to click on lines with U/V, meaning lines that I have not clicked before
5	How easy is for you to remember the meaning of X/Y for bullets?

X: purple color	Y: check mark
U: green color bullets	V: bullets without a check mark

Table 4: Recommendation questionnaire

1	Being able to see recommended lines is useful when X/Y is used for showing recommendations
2	I think X/Y help me correctly distinguish recommended lines
3	Using X/Y for recommended lines helps me distinguish lines that I have to pay more attention to them
4	I am motivated to click on lines with U/V, meaning lines that are recommended by the system
5	How easy is for you to remember the meaning of X/Y for bullets?

X: thick border	Y: red star
U: thick border bullets	V: bullets marked with a red star

for examples shown in this task: $C1A1$, $C1A2$, $C1A3$, $C2A1$, $C2A2$, and $C2A3$. Similar to Task 2, only half of these representations were shown to each subject. Odd-numbered subjects received combinations $C1A1$, $C1A3$, and $C2A2$, while even-numbered subjects received $C1A2$, $C2A1$, and $C2A3$. The examples were shown in a random order to each subject. At the end of the task, the subject was asked to select the design that made finding the recommended lines easier.

The content of all code examples consisted of 13 lines of code related to arithmetic operations in Java, out of which 8 lines had icon-based annotations that used one of the examined design alternatives. Figure 5 shows a sample annotated example, as presented to the subjects in Task 1. In each example presented during the tasks, the example lines were shuffled while the resulting code was kept meaningful. The reason for shuffling lines across different designs was to rule out variations in answers that could be caused by different levels of line complexity, and also to avoid the task becoming trivial, so that one could perform the task correctly by finding the right answer in one representation and pasting that answer in other representations.

5. DATA ANALYSIS

The alternative designs were evaluated using data collected from both questionnaires and tasks. We discarded data from one subject whose written answers to the tasks and questionnaires contradicted with their verbally expressed preferences. Among the 30 subjects considered for data analysis, 16 were females (4 undergraduates, 12 graduates) and 14 were males (4 undergraduates, 10 graduates).

The reliability of the questionnaire used in each design was assessed by measuring its internal consistency using Cronbach's *alpha*. Overall, all questionnaires were found to have an acceptable degree of internal consistency among the five questions (Cronbach's alpha statistic was greater than 0.6 for the $A1$ questionnaire, greater than 0.7 for the $B2$ questionnaire, and greater than 0.85 in the other questionnaires). Thus, all questions were retained for further analysis.

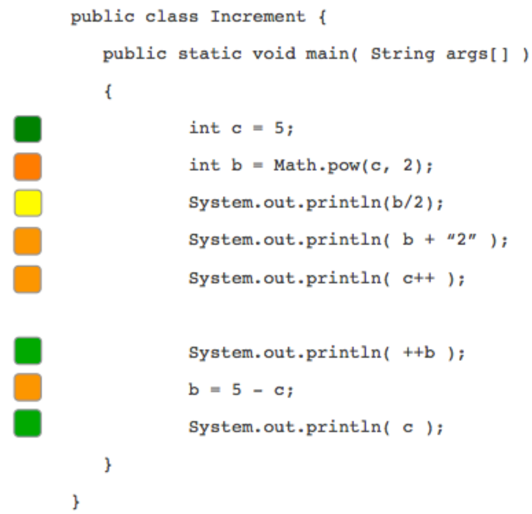

Figure 5: An example with design $A3$ used in Task 1.

5.1 Are Visual Cues Perceptually Different?

We analyzed the questionnaire data to explore the differences between user out-of-context ANS preferences. Responses were coded on a scale of 1 to 5, where higher scores indicate a greater satisfaction with the design. Since the correlations between the five questions was high, responses over all five questions in each questionnaire were aggregated to calculate a *preference score* for each design. To account for the correlation of within-subject observations, generalized estimating equations were used in comparisons of preference scores for (1) knowledge-based annotation, (2) history-based annotation, and (3) recommendation designs. The generalized estimating equation (GEE) model was estimated using a log link, a gamma distribution for the skewed dependent variable (i.e., preference score), and an exchangeable covariance structure. An alpha level of .05 was used to judge the statistical significance in all models.

(1) Comparisons of knowledge-based annotation designs. Overall, design $A1$ was found to have a higher preference score ($M = 4.37, SE = 0.02$), as compared to designs $A2$ ($M = 3.47$, $SE = 0.03$) and $A3$ ($M = 3.87, SE = 0.03$). The means of preference scores in these three designs were compared to test if the null hypothesis of no perceptual difference was true (i.e., $H_0 : A1 = A2 = A3$). The GEE analysis rejected H_0 by revealing a significant effect of design factors on preference scores ($\chi^2(2, N = 30) = 20.08$, $p < .001$). Bonferroni-corrected pairwise comparisons showed that the difference between means of preference scores was significant between the $A1$ and $A2$ designs (adjusted p-value <.001) and marginal when $A3$ was compared to $A1$ (adjusted p-value=.061) and $A2$ (adjusted p-value=.087).

(2) Comparisons of history-based annotation designs. To test whether there was any difference between the two history-based annotation designs (i.e., $H_0 : B1 = B2$), the means of preference scores in designs $B1$ and $B2$ were compared by performing a GEE analysis. The design factor was found to have a significant effect on the preference score ($\chi^2(1, N = 30) = 18.52$, $p < .001$). On average, design $B2$ received a higher preference score ($M = 4.63, SE = 0.02$), as compared

to design $B1$ ($M = 3.85$, $SE = 0.04$), and the difference was significant (adjusted p-value<.001).

(3) Comparisons of recommendation designs. Similar to the comparisons in (1) and (2), GEE analysis was performed to compare the means of the preference scores for design $C1$ and $C2$. The significant effect observed for the design factor rejected the null hypothesis: $H_0 : C1 = C2$ ($\chi^2(1, N = 30) = 27.66$, $p < .001$). Design $C2$ received a significantly higher preference score ($M = 4.67, SE = 0.02$), as compared to design $C1$ ($M = 3.85$, $SE = 0.03$)(adjusted p-value<.001).

Also, a follow-up analysis indicated no interactions between either preference score and gender (females/males), or between preference score and education level (undergraduate/graduate) of the subjects in the study.

These results demonstrate that selection of visual cues within *the same* adaptation approach significantly impacts user perception of ANS interfaces. The designs that used filled bullets ($A1$) performed significantly better than the design that used shades of green color ($A2$) (alpha level 0.1 or less) and considerably better than the second-best design ($A3$), which used a progression of orange to green colors. The design that annotated an example link with a check mark ($B2$) was significantly better than design that used the purple color ($B1$). Similarly, the design that annotated an example link with a red star ($C2$) received significantly higher preference compared to the design that used the thick border for the bullet ($C1$).

5.2 How Does Context Affect Preference?

While the comparison of designs by user direct out-of-context perception was important, we considered it to be an insufficient measure. We believed that comparing ANS designs in-context (i.e., a situation where users have to decode visual cues in search of most appropriate lines of real code examples) could provide more reliable data about the value of each design. We expected to reveal differences between designs by analyzing user performance and in-context perceptions collected during their work on tasks. However, there was almost no variation in the *performance* of subjects across the three tasks. All subject performed Tasks 2–3 correctly, and only three failed at performing Task 1. Thus, we focused on subjects' in-context *opinions* about the most efficient design. To compare out-of-context and in-context perception, we analyzed the percentage of people who favored designs $A1$, $A2$, and $A3$ before working on the task (i.e., out of context) and after working on the task (i.e., in context). To determine out-of-context preferences where users were not able to compare the designs directly, the design that received the highest preference score in questionnaires $A1$–$A3$ was selected as the favored design of each subject. For in-context preferences, we used the design that the subject explicitly selected as the most efficient in the task.

For the knowledge-based ANS, out of 30 subjects, 15 preferred $A1$, 1 preferred $A2$, 11 preferred $A3$, 1 preferred $A1A2$, 1 preferred $A2A3$, and 1 preferred all designs equally before performing Task 1. While the first task separately asked the user about the most efficient interface for finding lines with minimum and maximum knowledge, 29 out of 30 subjects selected the same design. For the minimum task, out of 30 subjects, 26(86.7%) preferred $A1$, 3(10%) preferred $A3$, and only 1(3.3%) preferred $A2$; namely, user preferences changed considerably in task context. Figure 6a illustrates how the favored design changed after performing Task 1. To simplify the comparison, the four subjects that had more than one preferred design were counted as favoring each of those designs. Most importantly, while design $A3$ was a considerable out-of-context contender, assessing the design in-context caused 9 of its 11 supporters to switch fully to $A1$. In other words, while the orange-to-green gradient colors looked to be a good idea before the study was performed, it was clearly harder to use this color scheme in-context to find the lines with the most or the least knowledge.

The favored designs for history-based annotation and recommendation of links also changed for some subjects after assessing the designs in context of Tasks 2 and 3. Table 5 shows how user perception changed after assessing designs in the context of performing Tasks 2 and 3. Out of 29 subjects who answered Task 2, in both the odd and even groups, those who favored design $B1(n = 4)$ or equally preferred $B1$ and $B2$ ($n = 5$) switched to versions of $B2$. In the odd group, out of 11 subjects who favored design $B2$ before the task, 3 switched to $B1$ in its combination $B1A1$. This is an interesting effect of both in-context and combination effect that shows that a generally weaker option $B1$ appeared to be more reasonable in combination $B1A1$, especially when the top design $B2A1$ was not an option. The preference for a recommendation design was more stable and changed for only two subjects. Out of 30 subjects who answered Task 3, among the odd group, one subject who favored $C2$ selected $C1A1$ and among the even group, one subject who favored $C1$ selected $C2A1$ as the most efficient design. Figure 6b combines the odd and even groups and shows the change in favored designs for annotating links with browsing history and recommendation. The number of subjects who favored design $B2$ increased after performing Task 2, from 86.2%(25 out of 29) to 89.7%(26 out of 29), while the number of supporters for design $B1$ decreased from 31%(9 out of 29) to 10.3%(3 out of 29). In the same figure, the number of people who fully or partly favored design $C1$ decreased from 20%(6 out of 30) to 10%(3 out of 30). The number of people who favored $C2$ decreased by one subject, from 93.3%(28 out of 30) to 90%(27 out of 30).

Taken together, these results show that the user assessment of different ANS design options could considerably change when working with them in realistic contexts and in combinations with other visual cues. In contrast, note that in all cases, the top designs $A1$–$B2$–$C2$ identified in out-of-context assessments increased their standing above other designs during in-context evaluation.

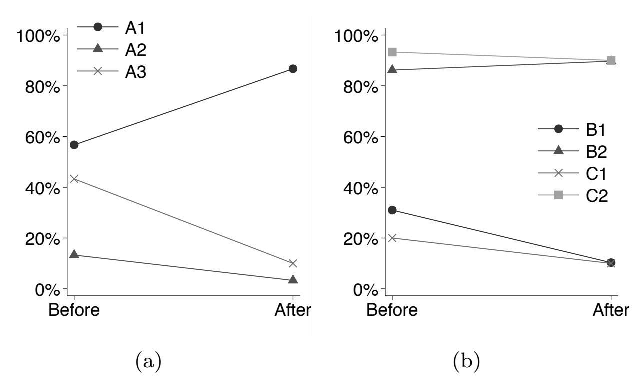

Figure 6: Percent of subjects favoring a design before and after performing (a) Task 1, and (b) Task 2-Task 3.

Table 5: #Subjects favoring a design before and after tasks

Before Task	After Task					
	Odd group			Even group		
	B1A1	B1A3	B2A2	B1A2	B2A1	B2A3
B1 (n=4)	0	0	2	0	1	1
B1B2 (n=5)	0	0	2	0	3	0
B2 (n=20)	3	0	8	0	9	0
	C1A1	C1A3	C2A2	C1A2	C2A1	C2A3
C1 (n=2)	0	0	0	1	1	0
C1C2 (n=4)	0	0	1	1	1	1
C2 (n=24)	1	0	13	0	9	1

6. DISCUSSION

This paper demonstrates that two or more alternatives for selection of visual cues within the same conceptual ANS approach might differ significantly from the perspectives of user perception and task performance. We investigated this question while comparing designs for annotating example links with information about student's knowledge, browsing history, and recommendations. Our findings stress the need to pay attention to designing visual cues, not simply to the approaches themselves.

The presented study had some limitations that must be addressed in future work. First, the order of presenting design alternatives in the first part of the study was the same for all subjects, which made the order a potential factor in obtained results. Second, the even-odd setting used in Tasks 2 and 3 did not allow to distinguish between top odd-numbered and even-numbered combinations, i.e., ($B2A1$ vs. $B2A2$) and ($C2A1$ vs. $C2A2$). Finally, the impact of combinations on user preferences in knowledge-based annotations was not assessed, since combinations were used only in tasks 2 and 3 and that asked about the most efficient design for showing clicked or recommended lines. However, despite these limitations, the study answered our main research question, helped us to pick the best design options, and learn important lessons about differences of user perception of visual cues both in and outside of the task context.

For future work, we plan to complement the current study with a classroom study that collects quantitative data on link usage and investigates the impact of the best designs on student learning.

7. ACKNOWLEDGMENTS

This research was partially supported by the Advanced Distributed Learning Initiative contract W911QY13C0032.

8. REFERENCES

[1] P. Brusilovsky. Adaptive hypermedia. *User Modeling and User Adapted Interaction*, 11(1/2):87–110, 2001.

[2] P. Brusilovsky. *Adaptive navigation support*, volume 4321 of *Lecture Notes in Computer Science*, pages 263–290. Springer-Verlag, Berlin, 2007.

[3] P. Brusilovsky, J. Eklund, and E. Schwarz. Web-based education for all: A tool for developing adaptive courseware. In *Seventh International World Wide Web Conference*, pages 291–300, 1998.

[4] P. Brusilovsky and M. Yudelson. From webex to navex: Interactive access to annotated program examples. *Proceedings of the IEEE*, 96(6):990–999, 2008.

[5] B. de La Passardiere and A. Dufresne. Adaptive navigational tools for educational hypermedia. In I. Tomek, editor, *ICCAL'92, 4-th International Conference on Computers and Learning*, volume 602 of *Lecture Notes in Computer Science*, pages 555–567, artc095, 1992. Springer-Verlag.

[6] J. Dron. Discovering the complex effects of navigation cues in an e-learning environment. In G. Richards, editor, *World Conference on E-Learning, E-Learn 2005*, pages 2026–2033. AACE, 2005.

[7] R. Hosseini and P. Brusilovsky. Javaparser: A fine-grain concept indexing tool for java problems. In *The First Workshop on AI-supported Education for Computer Science (AIEDCS) at the 16th Annual Conference on Artificial Intelligence in Education, AIED 2013*, pages 60–63, 2013.

[8] R. Hosseini, I.-H. Hsiao, J. Guerra, and P. Brusilovsky. What should i do next? adaptive sequencing in the context of open social student modeling. In *10th European Conference on Technology Enhanced Learning (EC-TEL 2015)*, pages 155–168, 2015.

[9] I. H. Hsiao, F. Bakalov, P. Brusilovsky, and B. Koenig-Ries. Progressor: social navigation support through open social student modeling. *New Review of Hypermedia and Multimedia*, 19(2):112–131, 2013.

[10] T. Loboda, J. Guerra, R. Hosseini, and P. Brusilovsky. Mastery grids: An open source social educational progress visualization. In *9th European Conference on Technology Enhanced Learning (EC-TEL 2014)*, pages 235–248, 2014.

[11] K. A. Papanikolaou, M. Grigoriadou, H. Kornilakis, and G. D. Magoulas. Personalising the interaction in a web-based educational hypermedia system: the case of inspire. *User Modeling and User Adapted Interaction*, 13(3):213–267, 2003.

[12] E. I. Sparling and S. Sen. Rating: How difficult is it? In *the Fifth ACM Conference on Recommender Systems*, pages 149–165. ACM Press, 2011. Need to move to reproducing results, Publish code and results.

[13] M. Yudelson, P. Brusilovsky, and V. Zadorozhny. A user modeling server for contemporary adaptive hypermedia: An evaluation of push approach to evidence propagation. In *11th International Conference on User Modeling, UM 2007*, pages 27–36. Springer Verlag, 2007.