Using Interactive Graphical Tools to Overcome the “Collage Effect” in Collaborative Learning Environments.

Christopher R. Slade  
Department of Computer and Information Sciences,  
Brigham Young University - Hawai’i  
sladec@byuh.edu

Daniel D. Suthers  
Department of Information and Computer Sciences,  
University of Hawai’i at Manoa  
suthers@hawaii.edu

Abstract

The “collage effect” occurs when participants in a collaborative learning environment post non-collaborative contributions to a shared workspace, as opposed to contributions that are co-constructive. The resulting artifacts become a collage of individual contributions that do not fully engage the participants in the collaborative learning process. Most online, asynchronous collaborative tools only allow participants to participate with textual interactions, which discourage co-construction in ways discussed in the paper. Graphical tools could provide better affordances for co-construction.

HTML provides little support to represent interactive graphical elements in online environments. However, with SVG and AJAX, we can create interactive graphical tools in an online environment. Using SVG and AJAX, we implemented concept maps in an online learning environment called MapCourse. We used MapCourse in several different asynchronous online assignments in two different courses. Our analyses suggest that concept maps encourage more co-constructive contributions than wikipages.

1. Introduction

Because collaborative learning is an effective way to help students build knowledge [15,4], many collaborative learning tools are being developed in asynchronous, online learning environments. Collaborative tools like blogs, wikipages, and threaded discussions are common tools found in most online courses. These tools help people share their ideas; however there is a tendency for these artifacts to become a collage of written expressions from various participants that do not engage directly with each other. We term this phenomenon as the “collage effect”.

The “collage effect” occurs because merging contributions in some media requires extra effort. For example, wikipages allow participants to merge separate written contributions. However, we have noticed that because of the structured nature of written text, i.e. grammar, sentence structure, paragraph structure, headings, etc., written contributions do not afford themselves to be easily modified by other participants. Due to possible social pressures, participants may be hesitant to edit another participant’s written contribution, especially in an educational setting. If a participant is inclined to edit another participant’s contribution, he is required to merge the written structure of his contribution to the written structure of the original contribution. Thus participants are more likely to include their contributions below other contributions and avoid merging the contributions into a coherent artifact. The resulting artifact becomes a collage of separate contributions.

In order for students to gain the most from collaborative learning, they need to merge their contributions together to build new knowledge [4,14,15]. Interactive graphical tools (IGT) may provide better affordances for students to do so than purely textual notations. The notational systems used in IGT prompt relevant contributions and support integration by making both the organization of expressed information and what is structurally missing more salient [19,20,17]. In contrast, the textual notation’s viscosity and low visibility of structure and dependencies make it more difficult to modify [1].

Yet, IGT have been limited in Web-based learning environments mostly because of the difficulty in creating them. HTML (hyper-text markup language) only provides a means of marking up textual elements. Graphical elements on the web are created by images, which, unfortunately, are not interactive. Previously, the only options for creating IGT on the web were with third-party plug-ins (like Adobe Flash), or Java applets. Both of these technologies require a separate development effort and require that the IGT be separated from the textual elements created with HTML.

Scalable Vector Graphics (SVG) is an emerging web standard that extends HTML to include graphical elements, such as lines and shapes. Combining SVG with asynchronous JavaScript and XML (AJAX) allows us to create web-based IGT. AJAX allows the client (or web browser) to send interac-
tions to the server without refreshing the entire page. It also allows the server to respond by updating elements on the page without refreshing the entire page. SVG and AJAX have three advantages over other ways of developing IGT on the web. First, developing graphical elements with SVG only requires a single development environment and does not require a separate development effort to create an applet or plugin. Second, SVG and HTML blend together much easier, which allows developers to include textual elements defined in HTML inside or right along side graphical elements defined in SVG. There is no boundary or separation between graphical elements and textual elements. Third, SVG does not require any separate downloads or plug-ins to be installed. It is supported by the latest versions of the Web's most popular browsers including Internet Explorer (as of version 9), Firefox, Safari, Google Chrome, and Opera. Also, unlike browser plug-ins, SVG is supported by mobile devices like cell phones and touch tablets such as the iPad.

We used SVG and AJAX to implement concept maps in a web-based, collaborative learning environment called MapCourse. MapCourse also contains wikipages and discussion threads. Our concept maps include the traditional features and allow participants to drag in other collaborative artifacts, such as wikipages, discussion posts, and even other concept maps, to be included as a concept in the map. We choose to implement concept maps because they have a long history of effectively supporting collaborative learning [2,11,12,20,21]. We used MapCourse in several online, asynchronous assignments in order to measure the collaborative effectiveness of wikipages and concept maps. The contributions of this work are, first, a new method to easily create IGT in Web-based collaborative learning environments, and second, a study that suggests that concept maps (an IGT) suffer less from the “collage effect” and encourage more co-construction than wikipages (a textual tool).

This paper will proceed as follows: First, we describe how SVG and AJAX are used to create concept maps in a Web-based environment. We then describe our procedures and methods of analysis of our online assignments. We finish with a discussion of our results.

2. MapCourse

MapCourse is Web-based collaboration environment that includes wikipages, Web links, document/file uploads, and concept maps. MapCourse is based on the design of disCourse [18], an online community and learning environment designed in collaboration with Sam Joseph and Viil Lid, but is a complete reimplementation with the addition of concept maps. Like disCourse, MapCourse is divided into shared workspaces that include links to the artifacts created within the workspace. The links to the artifacts are located in left and right sidebars, and the artifacts are grouped by the type of artifact (wikipage, concept maps, etc.). There is also a sidebar box that displays the workspace's participants. Each artifact has an independent discussion thread that is displayed below the artifact, when the artifact has been selected for viewing in the main content area. Each workspace has a designated main artifact that is displayed upon first entering the workspace. The main artifact can either be a wikipage, to provide a textual description of the workspace, or a concept map, to provide graphical navigation of the workspace. Figure 1 and Figure 2 show the workspace with a wikipage and a map respectively.

2.1. Other Methods to create IGT

The most common way of creating IGT in an online environment is with a browser plugin (Adobe Flash) or a Java applet. These methods require two development efforts. One development effort is needed to create the web application, and another development effort is needed to create the IGT. Creating IGT this way requires developers to physically separate the IGT from the web site. These methods also require the users to install either the plugin or Java. As stated above, some mobile platforms, such as cell phones and touch tablets, do not support these technologies.

Using these and other technologies several people have developed online concept maps. The most comprehensive tools are the CmapTools [3]. With CmapTools, users can create concept maps and post them online. All maps are viewable with a web browser, however to create or edit the concept maps, users need to download and install an application. Concept-Connector tools use an applet to implement concept maps in an online environment [10]. Commercial products that include concept maps have also been developed [9,16]. All of these technologies require a browser plugin or applet.

SVG and AJAX have an advantage because they do not require any additional plug-ins or downloads. SVG is mostly used to create online maps such as Google Maps or to create dynamic, data driven graphs. GeoBoost [6] is a geo-data visualization tool that uses SVG and AJAX. It also includes annotation tools built with SVG and AJAX. iMapping Wikis [8] is a zoomable-interface wiki. It allows users to create small wiki-like contributions, and then drag-and-drop these contributions in the workspace. Users can also
link contributions together. iMapping Wikis requires users to download and install it, but the authors suggest using SVG and AJAX as an alternative method of implementation.

2.2. Using SVG and AJAX to create IGT

We use a compound XHTML document [24] to create concept maps in MapCourse. A compound XHTML document allows us to blend textual elements defined with HTML with graphical elements defined with SVG. With SVG’s foreign object tag, we can include elements defined in HTML inside of graphical elements defined in SVG. We can then scale, translate, and rotate both textual elements defined in HTML and graphical elements defined in SVG. The ability to blend the elements together allows us to create IGT that contain the same font, look, and design of the other HTML elements. Thus, the IGT are not separated from the textual elements. Also, the IGT do not need a separate development effort to create, since they can be created by just generating an XHTML document.

To create the concept maps in MapCourse, we use the rectangles and lines from SVG to create the nodes (or concepts) and links respectively. Inside the rectangles we include HTML tags to place the text of the concepts. We also use HTML tags to place the text along the links. With SVG, we are able to translate the text to the appropriate location in the map.

We then use AJAX to send and receive updates from the server. This allows users to edit a concept’s text in place, and drag a concept within the map. Users can connect concepts by clicking on a link button in the concept, moving the mouse to another concept, and clicking on the concept. Users can delete a concept by dragging the concept off of the map. To create a new concept, users click on an open region within the map, and a form appears asking the user to enter the text of the concept. Users can also create a new concept by dragging any artifact in the sidebars into the concept map. Concepts created from sidebar artifacts include a link to view that artifact. All of these interactions are done without refreshing the page.
We want to determine whether students participate in a more co-constructive manner by using a wikipage (a textual tool) or by using a concept map (an IGT). The first author, in his capacity as instructor of two different face-to-face classes, designed some asynchronous online assignments that needed to be completed with MapCourse. The first class is a discrete mathematics course, and the second class is a beginning programming course. Undergraduate Computer Science, Information Technology, and Information Systems majors take both classes. After each assignment, the instructor gave each student contribution a score based on whether the contribution is co-constructive or adds to the “collage effect.” Students themselves were only evaluated on their level of participation in the assignments, and not on the quality of their contributions. Students were never given feedback on how they used the wikis or concept maps (with one exception noted below), as the objective was to observe their natural appropriation of the media.

Assignments were introduced in a face-to-face class meeting and students were instructed to complete the assignments outside of class without discussing the assignment with other students except through MapCourse. The instructor had the students reconfirm that the students were not meeting face-to-face to complete the online assignments. Also, the log files do not indicate that two students were logged in at the same time and making contributions in a coordinated effort. If two students were using the same account only the logged in student would receive credit for the assignment.

Group formation was done face-to-face in class by randomly assigning students to groups. Students

Figure 2. Screenshot of a concept map created by students in the beginning programming course.
that were not present in class were assigned to groups by the instructor, and were required to notify the other students in their group through the discussion threads in MapCourse.

3.1. Discrete Mathematics Procedures

In the Discrete Mathematics course, the instructor assigned the students to collaboratively create a midterm and final exam study guide in MapCourse. After describing the assignment, the instructor demonstrated how to use both the wikipage and the concept map tools. Because MapCourse was used to post the class schedule and homework assignments, students were familiar with the basic usage of the site, including previous assignments that only required participation in the discussion threads. The objective of this assignment was to encourage students to collaborate in the examination preparation process.

The instructor divided the class into groups of three to four students, and assigned each group a topic from the course. Students were assigned to provide a summary of the topic, definitions, and answers to some sample test questions. The students were given the freedom to use a wikipage, a concept map, or both. They were also permitted to create as many wikipages or concept maps as they deemed necessary. There were a total of 22 students in the course. For the midterm, five groups were created, and for the final seven groups were created. Each group was instructed to work completely online without meeting face-to-face. The students had one week to complete the assignment before taking the test. An example of student work in discrete mathematics is shown in Figure 1.

3.2. Beginning Programming Course Procedures

In the beginning programming course, the students were assigned to collaboratively create design documents in MapCourse. Because beginning programming students typically struggle with problem decomposition, the instructor assigned the students to create design documents that focused on breaking down programming assignments into smaller subtasks. Then the instructor assigned the students to give each subtask a programming construct (for loop, if statement, etc.) that would complete that subtask. For object-oriented based projects, we assigned the students to decompose the problem into classes, with members and actions. There were a total of three assignments. For the first two assignments, students were given a wikipage to complete the assignment. In the last assignment students were given concept maps to complete the assignment. The first two assignments are sequential programming assignments, and the last assignment is an object-oriented assignment.

In the first assignment, students were assigned to draw a multiplication table on the screen. Through nested for loops, students needed to draw boxes to create the table and then fill in the boxes with the multiplication results. In the second assignment, students were assigned to create a picture of a starry night. They opened a file that contained the coordinates of each star. As they read the star location from the file, they would mark their position in a two-dimensional array that represented their canvas. They would use the array as a guide to color each pixel of their canvas. In the last assignment, students were assigned to create a vending machine program that included dispensers and a cash register. An example of student work on this assignment is shown in Figure 2.

Although there are differences in the sequential programming assignments done with wikipages and the object-oriented assignment done with concept maps, the process is the same for both assignments. First, students break up the problem into smaller problems. In the sequential assignments, this consists of decomposing the problem into subtasks; in the object-oriented assignment, this consists of decomposing both the problems into subtasks and the data into smaller units (like decomposing a vending machine into a dispenser and a cash register.). Next, for the sequential project, students need to arrange the subtasks in order, and for the object-oriented project, they need to organize the subunits and subtasks into classes by relating them to larger units. Finally, students need to decide which programming constructs can solve each subtask. Although there are differences in the assignments, the differences should not affect the collaborative score because the process is the same.

There were a total of 18 students in the class that were divided into groups of three or four students. Groups were reassigned for each of the three assignments. For the first wikipage assignment, there were five groups, for the second wikipage assignment there were six groups, and for the concept map assignment there were five groups. Students were given three days to design their project online, and then an additional three days to complete the assignment. Students were allowed to start the assignment before the designed period ended. Each group was instructed to work completely online without meeting face-to-face, and the programming part of the assignment was completed independently.
3.3. Analysis

To analyze the results, we logged and examined all of the interactions in MapCourse. For the wikipages, we collected a snapshot of each edit. From the snapshots we determined the contributions by feeding the snapshots into a program that highlights the changes. For concept maps, we collected a log of all the interactions that occurred in the map. We then score each contribution in the following manner.

For concept maps, we grouped concurrent interactions intended to accomplish a single task together. For example, if a user creates a concept, moves the concept to the right location, and links that concept to another concept, then all of those actions are considered one contribution. Initial contributions or contributions that are added and linked to the same user's contribution are not scored. A contribution where a student adds a concept into a map that was started by another student receives a score of one. When a student creates a concept and links it to a concept created by another user, it receives a score of two. When a student links two concepts created by another student it receives a score of three. When a student edits the text in the concepts or along the links that was originally created by another user, it also receives a score of three, unless the edit is just a grammatical change. Grammatical changes are not scored.

For wikipages, like concept maps, contributions from users editing or inserting text into their own previous contributions are not scored, as well as the initial contribution of each artifact. If a single wiki-page edit accomplishes more than one task, then it is divided into separate contributions, each of which will receive a score. A contribution that just adds text that is not directly related to other contributions receives a score of one as shown in Figure 3. A contribution that relates or refers to another contribution receives a score of two as shown in Figure 4. If a contribution relates or refers to two or more contributions it receives a score of three as shown in Figure 5. Edits that significantly change the meaning or clarify a previous contribution also receive a score of three. As with concept maps, grammatical or formatting changes are not scored.

When analyzing the results, it is important to not let the number of co-constructive contributions influence the conclusions. When editing a wiki-page, a user can add a lot more content per edit. Even though we separate each edit into multiple contributions, each contribution can add more information than by adding a concept into a concept map. Therefore we base our conclusions on the co-constructive quality, or average score, of the contributions and not the quantity of the contributions.

4. Results

First we will report the results of the discrete mathematics course and then report the results of the beginning programming course.

4.1. Discrete Mathematics Results

For both the midterm and final review, there were a total of seven wikipages and ten concept maps cre-
ated. For the midterm review, the students created three wikipages and five concept maps. For the final exam, the students created four wikipages and five concept maps. A combined summary of the contribution scores from the midterm and final assignments is found in Table 1.

Table 1. Summary of contributions of the discrete mathematics course.

<table>
<thead>
<tr>
<th>Score</th>
<th>Wikipages</th>
<th>Concept Maps</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Total Contributions</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>Average</td>
<td>1</td>
<td>2.45</td>
</tr>
</tbody>
</table>

Of the seven wikipages, only one has edits by more than one student. That edit is a “collage effect” edit. The first editor posted several definitions and then posted answers to sample test questions. The second editor then posted several more definitions below the answers and even repeated some of the previously posted definitions.

Some students posted their contributions in the discussion threads below the wikipages. Since these posts were not editable by other users, we did not score these posts. We did look at the replies, but there were very few replies and none of them would be given a score higher than one if they were posted in a wikipage, and thus would only decrease the average. All of the replies were other participants adding their own unrelated contributions. There were no discussion threads on the concept maps.

### 4.2. Beginning Programming Course Results

A summary of the contribution scores in the beginning programming course is found in Table 2. The table only contains results from the second wikipage assignment and the concept map assignment.

For the first wikipage assignment (the multiplication table assignment), all of the students participated only in the discussion threads. Half way through the assignment, we encouraged the students to edit the group’s wikipage through an additional demonstration. After the additional demonstration, only one student actually put edits into the wikipage. Since there is little relevant data to collect on this assignment, we do not discuss the results of this assignment further.

For the second wikipage assignment students were not given credit for participating unless they edited the wikipage, and a further demonstration was given. Some students still participated in the discussion threads but at least one student in each group edited the wikipage. For the concept map assignment, all the students participated by editing the concept map.

Most of the concept map edits consist of adding a concept and linking it to other concepts. However, there are a larger number of concept map edits where a participant edited the text of the concepts or links. There are a total of fifteen concept map contributions that received a score of three. Of these fifteen contributions, twelve of them occurred when a student edited the text of a concept or a link. No student edited a previous contribution on a wikipage; the students just added more text into the wikipage. Also, with the concept map assignment, only one student posted something in the discussion threads below the maps. The post was informing the group members of an additional group member, as seen in Figure 2.

Table 2. Summary of contributions of the beginning programming course.

<table>
<thead>
<tr>
<th>Score</th>
<th>Wikipages</th>
<th>Concept Maps</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>Total Contributions</td>
<td>12</td>
<td>42</td>
</tr>
<tr>
<td>Average</td>
<td>1.5</td>
<td>2.36</td>
</tr>
</tbody>
</table>

### 5. Discussion

The results from both classes show that concept maps produce higher average scores than wikipages. These results suggest that concept maps lead students to be more co-constructive than wikipages. We discuss reasons for and implications of these results in this section.

### 5.1. Interpretation of Results

One thing to note is that the discrete mathematics assignments were not designed to be collaborative assignments. Each group took their assigned topic and divided the work up between the members, after which each member worked independently on their assignment. Some groups negotiated the division through the discussion threads. In most of the groups, the first student to contribute picked a subject, and the students who contributed later filled in the gaps. In some instances, members repeated information posted by another member of their group, as shown in the one wikipage edit that occurred. Other evidence of individual work includes the stu-
Students posting their contributions in the discussion threads, which did not receive any co-constructive replies.

However, in the concept maps of the discrete mathematics assignment, there were several co-constructive edits and a higher average score. This means that students decided to edit another contributor’s work in a concept map, whereas in theikipages (and the discussion threads) the students never attempted to co-construct. This suggests that concept maps encourage co-construction, since the students were not required to edit other contributions.

Three possible reasons why concept maps encourage co-construction are the following. First, the linking affordance of concept maps encourages relating contributions to each other [19]. Second, because there is less written text, students may be able to more quickly analyze the concept map and discover missing or incorrect information. Third, our concept maps have a limited amount of working space, whereas wikipages can continue to grow. Once a participant fills up a concept map, he cannot add more concepts. This forces additional participants to edit what has already been posted.

The beginning programming results also suggest that concept maps lead students to be more co-constructive. Since the concept map assignment was given after the wikipage assignments, it could be suggested that the extra experience allowed the students to perform better. However, we doubt this is the case, since the students were never given feedback to correct their “collage effect” habits.

As shown in the results of both courses’ assignments, many students preferred to post their contributions in the discussion threads. One possible reason is the fact that before these assignments, in both courses, students were given a few assignments to participate in a discussion thread. Thus, the discussion threads were the most familiar tool prior to the assignment. Another possible reason could be a hesitancy to edit a shared workspace due to social pressures. We cannot conclude anything about the lack of use of the discussion threads below the concept maps in the beginning programming course because that assignment was given after the previous two assignments, and students were instructed to post their contributions in the artifact and not in the discussion threads.

5.2. Implications for Practice

In the discrete mathematics course, the assignments were designed to help the students review the course material and prepare for the examination. Students, for the most part, were able to choose the tool that would best represent their material. For example, students who needed long explanations or mathematical formulas used the wikipages, and students who needed to differentiate definitions of terms used concept maps. From an informal oral survey, most students felt that the assignment helped them review for the test. Unfortunately, there are too many outside factors to determine whether the assignments help improve test scores.

In the beginning programming course, our learning objective was to help students develop better problem solving skills through a co-constructive learning process. When using the wikipages, the students never developed their design enough to sufficiently help students develop a strategy to decompose the problem. Students tended to work more individually on the design, and the resulting artifacts provided little help for struggling students. However, on the concept map assignments, students participated in a more co-constructive manner. As a result, the resulting artifacts represented a more complete design that students could use to develop a working program. An examination of the students’ code showed that most students tended to follow the designs made with the concept maps.

Also, from our study, we noticed that students tend to use the tool (discussion threads) with which they have the most familiarity. Instructors might need to motivate their students to use new tools either through assessment or limiting the available tools. Our students preferred to use the discussion threads, and needed extra motivation to use the wikipages and concept maps.

These results suggest that when creating collaborative online assignments, educators can increase the co-constructive quality of their assignments by using IGT in conjunction with appropriate instructions. Additional IGT’s can be created to provide the correct affordance to guide the students in the co-constructive process. By using an IGT as a medium for collaboration, instructors can reduce the “college effect” and encourage students to co-construct.

5.3. Related Work

Other work has also shown that graphical tools help increase co-construction. In a study involving a science challenge, graphical mapping tools appear to be more effective in supporting cooperative learning than textual representations [19]. Knowledge maps with embedded notes have also been shown to more effectively support collaborative knowledge building [17]. By using a content-specific visualization tool called CoStructure-Tool, dyads were more effective in integrating prior knowledge than dyads that used a
general whiteboard [7]. These studies all suggest that IGT help improve collaborative learning.

Besides using IGT, practitioners can also use scripting [25] to encourage more co-construction. However, scripting can disturb the natural problem solving process, interfere with the dynamics of goal achievement, and increase cognitive load [5]. IGT seem to be a more natural way to encourage co-construction.

6. Conclusion

Most asynchronous online learning environments tend to include mostly textual tools in order for students to collaborate. This is probably due to HTML’s limitation to only represent textual information. By including IGT, we can address some of the deficiencies of purely textual environments. This work focuses on overcoming the “collage effect” by implementing concept maps (an IGT) in an asynchronous learning environment.

The contributions of this work are, first, demonstrating that with SVG and AJAX, we can create interactive graphical tools in an asynchronous online environment. We can create these tools without a separate development effort to create a browser plugin or applet. Users would not have to download any additional plug-ins or software in-order to use these tools. Also, these tools blend well with other elements in HTML making the interface seamless. Second, our study results show that concept maps lead students to be more co-constructive than wikipages. We suggest that IGTs can overcome the “collage effect” and help alleviate other problems that occur in asynchronous online learning environments.

Further experiments are needed to explore how IGT will work in other disciplines besides math and computer programming. Experiments in other fields of study will be beneficial to further explore how the affordances of wikipages and IGT affect co-construction. Since this study focuses on exploring the affordances of wikipages and concept maps and the effects of these affordances, more empirical studies that directly compare the tools can be performed.

We also hypothesize that concept maps can help reduce information overload [23], because the concept maps serve as a visually accessible external memory for the group and supports coordination of their activity. A further study would be needed to verify this. We plan to explore other tools that can be built using SVG and AJAX. IGT can be designed to help overcome topic drift in discussion threads [22]. Many of the new tools designed to create knowledge building environments [13,14] can be created in asynchronous online learning environments using SVG and AJAX. These tools can help participants quickly find the state of the conversation, discover where contributions are needed, and participate in a more co-constructive way.

7. Acknowledgements

We would like to thank the students who participated in the assignments. We would also like to thank Sam Joseph and Vil Lid for their prior design and implementation of discourse, which greatly helped in the creation of MapCourse.

8. References


tovich, R.R. Hoffman, J. Felto


r.htm?arnumber=4388996.

[7] F. Fischer, J. Bruhn, C. Grasel, and H. Mandl, “Fostering collaborative knowledge construc-


