



Translating Educational Theory Into Educational Software: A Case Study of the Adaptive Map Project

Jacob P. Moore, Virginia Tech

Jacob Moore is a PhD Candidate in the Department of Engineering Education at Virginia Tech.

Michel Paul Pascale

Michel Pascale is a student at Virginia Polytechnic Institute and State University, and was one of the developers of the Adaptive Map project.

Dr. Christopher B. Williams, Virginia Tech
Chris North, Virginia Tech

TRANSLATING EDUCATIONAL THEORY INTO EDUCATIONAL SOFTWARE: A CASE STUDY OF THE ADAPTIVE MAP PROJECT

Abstract

In this paper, the authors describe the development of an instructional software, where developers engaged in the process of translating educational theory into a cyber-learning tool, and the challenges encountered in evaluating its usability and effectiveness of the tool as a learning aid. Specifically, the authors reflect on their experience in creating the “Adaptive Map” – an instructional software designed to help students gain conceptual understanding of large stores of content information. This concept map -based system explicitly shows how discrete concepts are linked to the whole of the course with a large node-link diagram. This explicit mapping of expert knowledge structures has been shown to promote conceptual understanding in students. Because concept maps become visually cluttered and unusable when they get too large though, an interactive visualization tool was developed to maintain the advantages of concept maps as learning tools while managing the visual clutter in maps that cover entire courses or even an entire curriculum. In this paper, the authors discuss the process they took in integrating the educational literature with the information visualization literature to understand how to best make an information visualization that addresses educational goals. Results from a heuristic analysis using Munzner’s four level validation framework for an information visualization are also presented.

1. Background and Motivation

Technology has the potential to aid instruction, but the simple act of using technology to deliver instruction does not improve the instruction being delivered ^[1]. In order to have a positive impact on student learning, instructional technology developers must draw on what is known about how people learn and then use technology to improve the quality of the instructional materials. This often involves collaboration between researchers with backgrounds in education and those with backgrounds in software, or other technologies. This paper serves as a case study of one such instructional software development process, the development of the Adaptive Map digital textbook, so that future collaborations might gain insight into developing instructional technologies.

1.1 The Theory Behind the Adaptive Map

The Adaptive Map is an attempt to use the flexibility of modern software to improve the design digital textbooks in ways that was not previously possible. The Adaptive Map tool is a collection of content pages that collectively serve as a textbook for a course. In its current iteration, the Adaptive Map (available at www.adaptivemap.me.vt.edu) has been developed with content in engineering statics. Statics was chosen for an initial content area because it is a

foundational course for a variety of engineering majors, thus maximizing the potential impact of the tool.

The central design feature of the Adaptive Map tool is a concept map based navigation system. Research studies have previously found that by using expert generated concept maps to show how different topics in lesson are related, students are better able to understand and retain the content information presented ^[2-6]. Expert-generated concept maps serve as advance organizers ^[7] and improve understanding by mobilizing relevant prior knowledge the new content can be cognitively connected too ^[8].

However, concept maps have problems with scalability. When concept maps become too large and complex, users encounter “map shock” ^[9]. Map shock is a cognitive and affective reaction to large scale concept maps where learners become overwhelmed and disengage from processing the concept map. This diminishes the effectiveness of large expert generated concept maps as learning tools. Concept maps are therefore effective for organizing small amounts of content information in a way that students can better understand it, but they cannot currently be used to organize large quantities of information. This limits the effectiveness of concept maps as a way to enhance the effectiveness of course textbooks.

The symptoms of map shock match the more broadly defined phenomenon of cognitive overload, as described in cognitive load theory ^[10, 11]. In order to avoid map shock in learners, the visuals presented must be managed in a way that helps learners manage their cognitive load. In order to better manage the cognitive load of users, the authors used information visualization techniques to create the Adaptive Map’s interactive navigation system for large scale concept maps. This instructional tool was built to test the hypothesis that information visualization techniques could be used to increase the effectiveness of large scale concept maps as advance organizers by managing a user’s cognitive load and eliminating map shock.

1.2 Overview of the Adaptive Map Tool

The Adaptive Map software features semantic zooming techniques to enable the user to explore the large-scale concept map. With this technique the scope of the material being covered determines the level of detail presented in the visualization. By having the software limit the amount of information presented at any one time, the software is managing the cognitive load imposed on the learner.

The tool opens by presenting users with an overview of all the information covered at a high level of abstraction. At this overview level the topics are grouped into clusters of highly related ideas, similar to chapters in a traditional book. Each cluster is represented as a node in a graph, and the links between the nodes represent direct relationships between the topics in each of the clusters. These links are directed and generally flow from more basic prerequisite clusters at the top of the screen to more advanced post-requisite concepts at the bottom of the screen. The

links' line thickness is directly related to the number of direct connections between topics within those two clusters. An image of the overview can be seen in Figure 1.

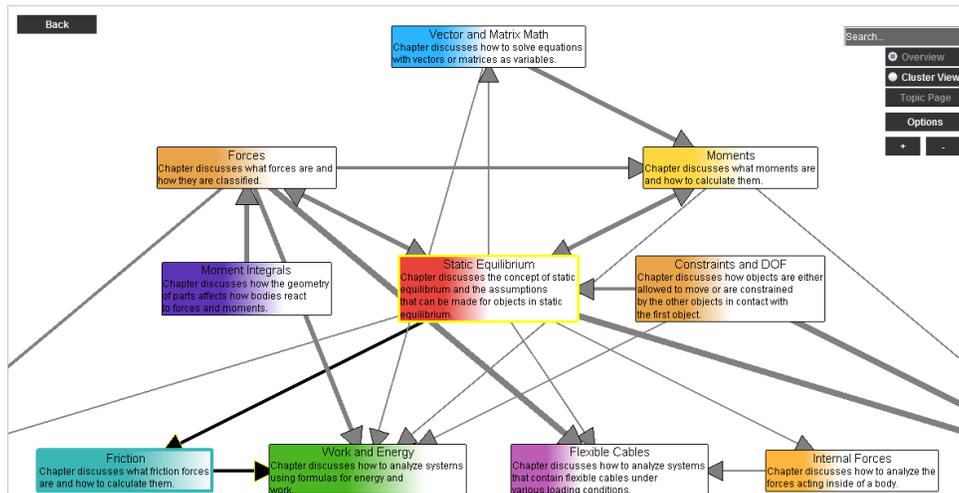


Figure 1: Adaptive Map Overview

The user at any level can pan by either clicking on cluster nodes to center them on the screen, or by clicking, holding and dragging the background. The user can also zoom in or out using scroll wheel, or by using the + and – buttons on the screen.

If the user zooms in to (or clicks on) any of the clusters in the overview, the cluster node will break apart into topic nodes and give the user more details on that cluster. Each cluster node contains several topics, where a topic was defined as the smallest independently teachable lesson. Information on how the topics and clusters were identified can be found in previous literature^[12]. At this level, the topics are represented by individual nodes in the concept map and the relationships between the topics are represented by links. A sample screenshot of the “Static Equilibrium” cluster is shown in Figure 2.

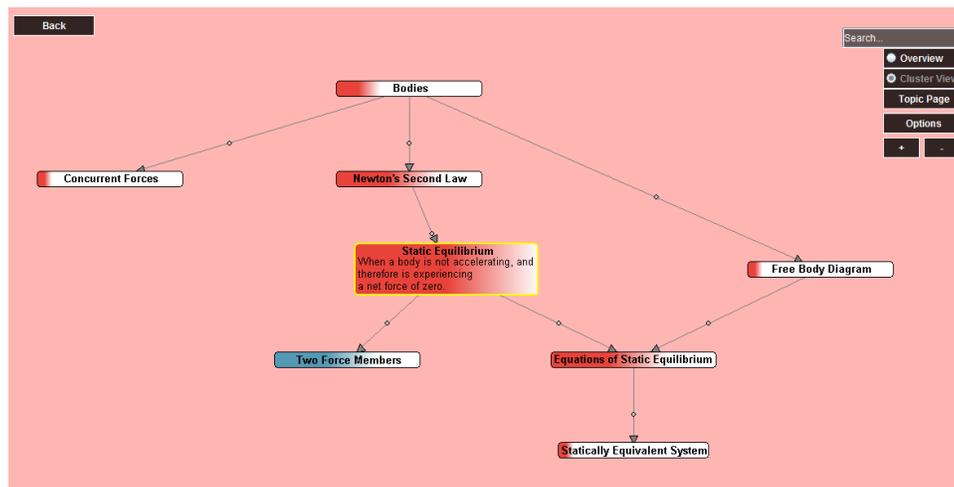


Figure 2: Adaptive Map Cluster View

In Figure 2, the user is focused on the static equilibrium topic, as indicated by the yellow border around the node and the description of the topic in the node. The background color and all the topics within the static equilibrium cluster are red to match the color of the static equilibrium cluster node in the overview. Any topic that is directly related to the focus topic from other clusters is also drawn in. In this case, the two force member topic is directly related to the static equilibrium topic, though the two force member topic is part of another cluster. These cross-cluster relationships are amalgamated into the links seen in the overview. More details on the topics or the relationships can be found by hovering over the nodes or link in these views.

If one zooms in further to a topic nodes, one will view the content page associated with that topic. These topic pages resemble the content in more traditional textbooks, with explanations, images and worked example problems. A screenshot of part of the “Static Equilibrium” Topic Page is shown in Figure 3.

STATIC EQUILIBRIUM



The structural members in this high rise building are in static equilibrium.



This truck, moving at a constant velocity, is in static equilibrium.

Objects in static equilibrium are objects that are not accelerating (either linear acceleration or angular acceleration). These objects may be stationary, such as a building or a bridge, or they may have a constant velocity, such as a car or truck moving at a constant speed on a strait patch of road.

Figure 3: Adaptive Map Topic Page

Through these three levels of zoom, learners can explore the topics and the relationship between topics contained in the adaptive map. The interface helps users maintain a sense of context within the overall concept map, and helps match the level of detail displayed to learner’s current level of focus (from a course wide overview, to a chapter overview, to a single topic focus).

2. Software Development Process

2.1 Developing the Software Goals

The software development process began with the formation of software design goals. These design goals were based on an extensive review of the educational literature relating to concept

maps and cognitive load theory. In addition computer science literature related to information visualization, particularly the information visualization literature pertaining to graphs, was consulted. The software design goals and citations of the related literature are listed below.

1. The tool will act as a digital content repository with a concept map based navigation scheme. The tool must be able to display detailed information as well as concept maps of the embedded information at several levels of detail and abstraction, and it must allow the user to easily navigate horizontally (from one topic to another) and vertically (from one level of detail to another).
2. Usability of the visualization as both a learning and navigation tool is the primary goal of the software. The information should be displayed in a way that does not cognitively overload ^[10, 11] the user.
3. The tool will automatically generate the visualizations based on metadata from the content developer. The content developer will develop content pages with metadata to inform the software what other topics a given idea is directly related to, what the nature of those relationships are, and to what groups the topic belongs. The software should interpret this information and determine how to best visualize the information automatically based on the expert-generated metadata.
4. The tool should be as modular as possible. Software and content should be kept separate so that future content can be developed without understanding the inner workings of the visualization software. This should lower the barrier to adoption for future content developers.
5. The software should provide means for the user to adjust the visual presentation settings. Since many of the ideal parameters for a concept map are learner dependent ^[13], the user should be able to change the settings to match his or her needs.
6. The concept map displays should provide good “symmetry” and “predictability.” The concept map displays should display node-link structures with similar relationships in similar ways. The displays should also be consistent temporally - if the user leaves a view and then later comes back to it, it should look the same as it did the first time. Good symmetry and predictability are shown to enhance comprehension of concept maps and therefore induce lower cognitive loads ^[13-15]. Also, small changes to the content should not radically alter the output. Both of these goals can be accomplished by having consistent and structured rules for how the concept map visual is built.
7. The concept map should minimize link lengths and link crossings. This will provide good proximity for directly linked nodes and reduce the mental effort needed to interpret the links ^[13, 16-17].
8. The concept map should provide good “continuation”. Concept maps with directional links that generally flow in a single direction are easier to understand than concept maps that have links with random or outward directions of flow ^[13]. This direction of flow should be vertically oriented for further ease of interpretation ^[2].

9. The nodes and links should utilize color and shape as indications of different features of the nodes and links. This should be kept fairly simple, however, and the degree of color and shape variation should be able to be adjusted by the user.
10. The tool should help the user maintain a sense of context while viewing details, as this is difficult to do within large scale concept maps ^[18, 19]. To accomplish this goal, smooth zooming and panning, multiple window setups, distorted views, or some combination of these methods will be employed.
11. Upon startup, the tool should display an overview of all the content available to the user. Introducing a visualization via an overview has been shown to enhance the sense of context for later displayed information ^[20].

2.2 Developing the Visualization Software

The software design team then began to develop the software to support these software design goals. As a first decision, the design team decided to develop the Adaptive Map tool as a Java applet, as this approach offered many advantages. First, the native cross-platform compatibility of Java allowed the team to distribute the program to students on all major platforms, without having to write much platform-specific code. Second, prior experiences with Java and the widespread availability of free web tutorials about programming with Java allowed all team members to participate in development process quickly without learning the details of more uncommonly used software. Finally, using a Java applet allowed the team to host the program on one central server rather than having users install and have to update software on their personal computers. This provided all users access via web browser and the centralized location allowed for simple and efficient updates to both the software and the content.

Once the team chose the programming language, the team selected a tool to use for the visual aspect of the program. The tool needed to be able to animate many objects at the same time in a smooth and efficient manner in order maintain a sense of context (software design goal #10) Of the possibilities that were considered, the “Zoomable Visual Transformation Machine” (ZVTM) toolkit ^[21] was chosen. This toolkit provided a simple programming interface that allowed the team to create and animate objects without much effort. It also provided the source code for all of its objects, which allowed for easy modification that better allowed the design team to address a variety of software design goals.

To properly implement design goals 3, 6, 7 and 8, which deal with graph layout, the development team researched available graphing software. Many free graph layout algorithms have already been developed and optimized by outside researchers, and it was deemed best to not try to figuratively “reinvent the wheel” for this particular task. The first graphing toolkit the team tested was the “Hierarchical Layout Plugin”, developed by Robert Sheridan for the Institute for Systems Biology ^[22]. This plugin worked well for getting a basic layout of nodes, but did not adequately minimize link crossings (software design goal #7). Graphviz ^[23] was ultimately selected as it provided a better graphing solution than the Hierarchical Layout plugin, with less

link crossings overall. Graphviz also provided the ability to pre-generate the graph layout solutions beforehand so that overall layout consistency could be maintained (software design goal #6). This created an extra step for the content developer when changing the content, but this drawback was balanced by the decreased load on user's machines. These pre-generated maps could be downloaded once upon loading the Adaptive Map and saved, rather than having user's machines continually regenerate the new maps as the user was exploring the concept maps.

Finally, it was decided that the content should be stored in the form of HTML and XML pages. HTML and XML were chosen as ways to encode the content pages and the concept map layout because of their simplicity. This ensured that content developers not familiar with programming would be able to translate their concept into a virtual book.

The HTML pages were designed to offer instruction on specific topics and provide relevant worked examples. Developing HTML is a fairly simple exercise and many tools are already available to help aid the construction of HTML pages. XML was used to encode the concept map structure, how all the topic pages were linked together. XML was chosen because of its simplicity, and because it allowed the content to be read by the Java visualization software while still keeping the content and software separate (software design goal #4).

2.2 Creating Content for the Adaptive Map

In order to develop the concept map itself, content developers would start by creating a course wide concept map of the subject area. There are many software packages that aid in creating concept maps, but the developers for this project simply used Microsoft PowerPoint. A process for creating large-scale concept maps is outlined in previous literature ^[12]. After a concept map is created it is converted into an XML document so that the software can interpret the concepts and links between concepts. This XML document captures the structure of the concept map (what the nodes are and what other nodes they are linked to), while leaving the physical layout of this concept map to be determined by the visualization software. To further aid with content management, a simple Java interface was developed to allow content developers to more easily create and manage the large XML documents needed to represent the large concept maps (Figure 4).

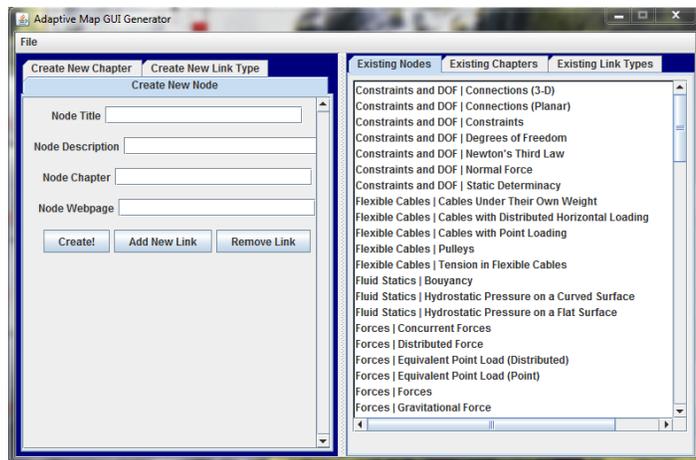


Figure 4: The XML Editor

After the main features of the program were in place, the team tested many optional features that the user could adjust (software design goal 5). Some users found the ability to drag different

nodes around useful, while others found that it got in the way. This made it a good candidate for a user option. Some users that dragged nodes found that the nodes could often get messier than they had expected. Thus, the team implemented a feature that returned all nodes to their original positions. Others found the size of the font to be too small, so the team added buttons that allows users to increase or decrease the size of the font. When the program transitioned from “Overview” mode to “Cluster View”, some users found themselves zooming out after each transition, while others enjoyed using the default zoom. A slider-bar was added to allow users to set a default post-transition zoom level. All of these optional features made the program more flexible, and allowed users to customize the Adaptive Map to meet their particular preferences.

3. Software Evaluation

The ultimate goal of the textbook is to promote conceptual understanding across a variety of topics in engineering statics. To evaluate this, the researchers are using data collection sessions in which engineering statics students use the Adaptive Map tool to study a prescribed topic area and then explain that topic area to the researchers. Transcripts of these sessions are being analyzed to get a measure of conceptual understanding. During these sessions, students are also self-reporting levels of cognitive load experiences while preparing for the explanations. Testing of this type is ongoing, but a way to more quickly identify design flaws for correction was desired.

In order for the Adaptive Map tool to be useful as a learning tool, it must first be an effective information visualization. Students must be able to draw meaning from the concrete representations of abstract concepts and relationships. In order to evaluate the Adaptive Map tool as an information visualization, a heuristic analysis of the visualization was performed. For this heuristic analysis, the researchers used Munzner’s four level information visualization framework ^[24].

3.1 Munzner’s Four Level Model for Visualization Design and Validation

Munzner’s model breaks the information visualization development process down into four stages, with each stage being dependent upon the previous stage ^[24]. To validate information visualizations using Munzner’s model, a developer needs to examine his or her design with respect to a number of threat’s to validity at each level. If the design fails to address the threats to validity at any of the four levels, the information visualization will not be an effective tool. Table 1 outlines Munzner’s four level model for visualization design and validation.

Table 1: Munzner’s Level Model for Visualization Design and Validation

Level	Primary Question	Threats to validity
Problem Level	Is the visualization attempting aid a legitimate user task?	The visualization is aiding a task the user does not need to perform. The visualization is aiding a task the user does not need help with.
Data Abstraction Level	Is the visualization presenting the appropriate data to perform the user task outlined in the problem level?	The data necessary to complete the user task is not presented. The data presented to the user is irrelevant to the user task.
Data Encoding Level	Do the visualization strategies and interaction techniques effectively communicate the data outlined in the data abstraction level?	The data is presented in a way that is difficult to understand. The data is presented in a way that makes it difficult for the user to navigate and explore the data in a meaningful way.
Algorithm Level	Do the software algorithms effectively implement the visualization strategies and interaction techniques outlined in the in the data encoding level?	The software algorithms cannot effectively execute the visualization strategies and interaction techniques.

3.2 Evaluating the Adaptive Map Using Munzner’s Framework

The Adaptive Map tool was examined through the lens of Munzner’s framework by both the developers and by a number of evaluators external to the project. The external evaluators consisted of 22 students enrolled in and information visualization class, these students were asked to evaluate the Adaptive Map tool according to Munzner’s model as part of a class assignment. Because most students in the class were not familiar with the content in engineering statics, two additional external participants who were familiar the statics content (one professional and one student) were brought into a class to interact with the tool to aid in the evaluation. An outline of the heuristic analysis and a summary of the suggestions from external evaluators is outlined below.

3.2.1 The Problem Level

At the problem level, users of the Adaptive Map tool were encouraged to use the tool as a replacement to their textbook. Common user tasks consist of looking up information to solve homework problems, studying for exams, and catching up with the information from missed classes. In all of these cases, the students are attempting to learn the content information in the textbook. Concept maps in particular aid in the formation of conceptual understanding^[8], a type of learning that usually lags behind the acquisition of procedural knowledge in mechanics courses such as engineering statics^[25–27].

In general, the external evaluators agreed that the task of learning the material during out of class hours was a valid task for statics students, the presumed users of the Adaptive Map tool. Because the tool is filling the role of an existing tool, that of a textbook, the user tasks were well defined and familiar to the users already. Additionally, several external evaluators agreed that gaining an understanding of the concepts in a course was more difficult than learning and using procedures.

3.2.2 The Data Abstraction Level

At the data abstraction level, the developer had to determine if the types of data being presented were relevant to the identified user tasks. The Adaptive Map presents two primary types of data to the user, concepts and relationships. Each of the concepts and relationships has many associated characteristics, but the core data is the concepts and relationships between those concepts. Concepts and relationships were specifically chosen because this is how people store knowledge in cognitive schemas in long term memory ^[28]. Concept maps were originally developed as a way to externalize these cognitive schemas in learners for evaluation ^[8], and the close correlation between the data presented and the way humans remember information speaks to the validity of the data presented. Overall, the external evaluators found that concepts and relationships were relevant to learning, but there was also a call for more data types that were not present.

The first data type that evaluators found was missing was a single linear ordering of the concepts. Though the directed nature of the links provided some guidance through the content, there was desire from a large number of evaluators for a linear ordering to the concepts. Without a “first page”, evaluators were left wondering where to start. This is an artificial data, stemming not from the way the mind works or from expert understanding but from experiences with previous textbooks. A number of different orders are valid for learning the topics, but as novices with a newfound choice on their hands (where to start and what to learn next), the evaluators were understandably uneasy. Preliminary results from the classroom based evaluation do not show similar feelings however, most likely because the course schedule provided the information on where to start and what to learn next. Since the linear ordering would interfere with the presentation of the relationships between ideas, the suggestion to add a linear ordering is not immediately being pursued. Developers have taken this into consideration however as a possible barrier to adoption.

The second data type that evaluators found was missing was user history data. The evaluators wanted a system to record what pages they visited, what had already been covered in class, and a system to record notes on the topics. Though some data was available through the back button, there was a call for more of this data to be recorded and available. Preliminary results from the classroom evaluation indicate that actual students desire to have this information as well. The desire for this information may stem from a performance orientation to learning ^[29], where students measure their success by checking off what they have done, what they understand, and

how they compare to their classmates. A system of tracking user history and user performance on integrated quizzes will be added to future versions of the adaptive map tool.

3.2.3 The Data Encoding Level

At the data encoding level, the primary visualization strategy was to use a graph, in the form of a concept map, to represent the data of concepts and relationships. To navigate and interact with the graph, a semantic zooming strategy was employed to manage cognitive load. Graphs are the most obvious and usually the best way to encode sets of data consisting of items and relationships between those items. The external evaluators generally agreed that the concept map structure was the best way to display the concept and relationships and that the semantic zooming technique was a valid way to manage cognitive load, but they did have two prominent suggestions to improve the data encoding design.

First, the evaluators had trouble learning what all the encodings meant (node color, node gradient, link thickness, etc.). There was some textual explanation of what these encodings represented, but the prevalence of this confusion indicated that the explanations were not sufficient. In order to address the issue, videos were created to introduce new users to the tool and to better explain what each of the encodings meant. Preliminary results from the classroom assessment corroborate this initial confusion with the data encodings, but after a few instances of using the tool, students did seem to be able to comprehend all the data encodings with ease.

The second recurring suggestion the evaluators offered was to improve the overview encoding. There were still concerns about cognitive overload due to the many overlapping links that were present in the overview. Preliminary results from the students in the classroom evaluation corroborate this claim, indicating that the overview was the most difficult view to make sense of. More design work will need to be done to determine an alternative method for representing the data in the overview.

3.2.3 Algorithm Level

The data encoding strategy is carried out via a Java applet supported by graph layout and visualization softwares. Evaluators found the software to be sufficiently fast and efficient in carrying out the design encoding strategy. The evaluators did find software bugs however, when refreshing the page and when clicking during a transition animation. These bugs could interfere with the encoding design under some conditions. These bugs are being addressed to improve the quality of the Adaptive Map tool.

4. Closure and Future Work

The software development process for the Adaptive Map tool ensured that design choices for this instructional technology stemmed from the literature on how people process information and learn (conceptual understanding, concept map, cognitive load theory and information

visualization literature in this case), not what the technology was capable of. By using a learning centered approach to design, the technology was served as a way to expand the design possibilities of the instructional tool.

Results from a heuristic analysis with external evaluators served as a way to identify and address issues with the instructional technology quickly and efficiently before the tool is tested in the classroom. This analysis has served as a precursor to the forthcoming in class evaluation with authentic users (students in a statics class).

Analysis of the effectiveness of using information visualization techniques to improve instructional technology is ongoing. Work needs to be done to more carefully examine the Adaptive Map tool's effect on conceptual understanding in students, and furthermore to examine how many of the design features of the adaptive map relate to student perceptions of the content and student learning. However, initial results indicate that there is a strong potential for collaboration between information visualization and education.

5. Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. NSF TUES-1044790. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors would also like to acknowledge Lauren Gibboney, Joseph Luke, James McIntyre, John Nein, and Joshua Rush for their work developing the Adaptive Map tool.

6. References

- [1] T. L. Russell, *The No Significant Difference Phenomenon*. North Carolina State University, 1999.
- [2] D. F. Dansereau, "Node-Link Mapping Principles for Visualizing Knowledge and Information," in *Knowledge and Information Visualization*, vol. 3426, S.-O. Tergan and T. Keller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 61–81.
- [3] G. W. Ellis, A. Rudnitsky, and B. Silverstein, "Using concept maps to enhance understanding in Engineering Education," *International Journal of Engineering Education*, vol. 20, pp. 1012–1021, 2004.
- [4] M. W. A. Salata, "Concept Maps as Organizers in an Introductory University Level Biology Course," PhD., University of Virginia, Charlottesville, VA, 1999.
- [5] R.-S. Shaw, "A Study of Learning Performance of E-learning Materials Design With Knowledge Maps," *Computers & Education*, vol. 54, no. 1, pp. 253–264, 2010.
- [6] M. Willerman and R. A. Mac Harg, "The Concept Map as an Advance Organizer," *J. Res. Sci. Teach.*, vol. 28, no. 8, pp. 705–711, Oct. 1991.
- [7] D. P. Ausubel, *Educational Psychology; a Cognitive View*. New York, NY: Holt, Rinehart and Winston, 1968.

- [8] J. D. Novak and A. J. Cañas, “The Theory Underlying Concept Maps and How to Construct and Use Them,” Technical Report Cmap Tools 2006-01 Rev 01-2008, 2008.
- [9] J. Blankenship and D. Dansereau, “The Effect of Animated Node-Link Displays on Information Recall,” *The J. of Expt. Educ.*, vol. 68, no. 4, pp. 293–308, 2000.
- [10] J. Sweller, “Cognitive Load During Problem Solving: Effects on Learning,” *Cognitive Science*, vol. 12, no. 2, pp. 257–285, Apr. 1988.
- [11] F. Paas, A. Renkl, and J. Sweller, “Cognitive Load Theory: Instructional Implications of the Interaction Between Information Structures and Cognitive Architecture,” *Instructional Science*, vol. 32, no. 1/2, pp. 1–8, Jan. 2004.
- [12] J. Moore, R. S. Pierce, and C. B. Williams, “Towards an ‘Adaptive Concept Map’: Creating an Expert-Generated Concept Map of an Engineering Statics Curriculum,” presented at the ASEE Annual Conference, San Antonio, TX, 2012.
- [13] D. A. Wiegmann, D. F. Dansereau, E. C. McCagg, K. L. Rewey, and U. Pitre, “Effects of Knowledge Map Characteristics on Information Processing,” *Contemporary Educational Psychology*, vol. 17, no. 2, pp. 136–155, Apr. 1992.
- [14] S. C. North, “Incremental Layout in DynaDAG,” in *Graph Drawing*, vol. 1027, F. J. Brandenburg, Ed. Berlin/Heidelberg: Springer-Verlag, 1996, pp. 409–418.
- [15] I. Herman, M. Delest, and G. Melançon, “Tree Visualisation and Navigation Clues for Information Visualisation,” *Computer Graphics Forum*, vol. 17, no. 2, pp. 153–165, Jun. 1998.
- [16] D. S. Wallace and S. C. West, “The Effect of Knowledge Maps That Incorporate Gestalt Principles on Learning,” *Journal of Experimental Education*, vol. 67, no. 1, pp. 5–16, 1998.
- [17] H. Purchase, “Which Aesthetic Has the Greatest Effect on Human Understanding?,” in *Graph Drawing*, vol. 1353, G. DiBattista, Ed. Berlin/Heidelberg: Springer-Verlag, 1997, pp. 248–261.
- [18] I. Herman, G. Melançon, and M. S. Marshall, “Graph Visualization and Navigation in Information Visualization: A Survey,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 1, pp. 24–43, 2000.
- [19] R. Spence, *Information Visualization: Design for Interaction*, 2nd ed. Harlow, England: Pearson, 2007.
- [20] C. North, “Information Visualization,” in *Handbook of Human Factors and Ergonomics*, 3rd ed., New York, NY: John Wiley & Sons Inc, 2005.
- [21] E. Pietriga, “A Toolkit For Addressing HCI Issues in Visual Language Environments,” in *2005 IEEE Symposium on Visual Languages and Human-Centric Computing*, 2005, pp. 145 – 152.
- [22] M. S. Cline, M. Smoot, E. Cerami, A. Kuchinsky, N. Landys, C. Workman, R. Christmas, I. Avila-Campilo, M. Creech, B. Gross, K. Hanspers, R. Isserlin, R. Kelley, S. Killcoyne, S. Lotia, S. Maere, J. Morris, K. Ono, V. Pavlovic, A. R. Pico, A. Vailaya, P.-L. Wang, A. Adler, B. R. Conklin, L. Hood, M. Kuiper, C. Sander, I. Schmulevich, B. Schwikowski, G. J. Warner, T. Ideker, and G. D. Bader, “Integration of biological networks and gene expression data using Cytoscape,” *Nature Protocols*, vol. 2, no. 10, pp. 2366–2382, 2007.
- [23] J. Ellson, E. Gansner, L. Koutsofios, S. C. North, and G. Woodhull, “Graphviz— Open Source Graph Drawing Tools,” in *Graph Drawing*, P. Mutzel, M. Jünger, and S. Leipert, Eds. Springer Berlin Heidelberg, 2002, pp. 483–484.
- [24] T. Munzner, “A Nested Model for Visualization Design and Validation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 921–928, Nov. 2009.
- [25] J. Clement, “Students’ Preconceptions in Introductory Mechanics,” *American Journal of Physics*, vol. 50, no. 1, pp. 66–71, 1982.
- [26] L. C. McDermott, “Research on conceptual understanding in mechanics,” *Physics Today*, vol. 37, no. 7, p. 24, 1984.

- [27] P. S. Steif, "An Articulation of the Concepts and Skills Which Underlie Engineering Statics," in *Proceedings of the 34th Annual Frontiers in Education Conference*, Savannah, GA, 2004, p. F1F-5.
- [28] J. E. Ormrod, *Human Learning*, 5th ed. Upper Saddle River, N.J.: Pearson/Merrill Prentice Hall, 2008.
- [29] C. Ames and J. Archer, "Achievement goals in the classroom: Students' learning strategies and motivation processes," *Journal of Educational Psychology*, vol. 80, no. 3, pp. 260-267, 1988.