

Adapting the Cognitive Walkthrough Method to Assess the Usability of a Knowledge Domain Visualization

Kenneth Allendoerfer¹, Serge Aluker², Gulshan Panjwani³, Jason Proctor⁴,
David Sturtz⁵, Mirjana Vukovic⁶, and Chaomei Chen⁷

College of Information Science and Technology
Drexel University

ABSTRACT

The usability of knowledge domain visualization (KDViz) tools can be assessed at several levels. Cognitive Walkthrough (CW) is a well-known usability inspection method that focuses on how easily users can learn software through exploration. Typical applications of CW follow structured tasks where user goals and action sequences that lead to achievement of the goals are well-defined. KDViz and other information visualization tools, however, are typically designed for users to explore data and user goals and actions are less well understood. In this paper, we describe how the traditional CW method may be adapted for assessing the usability of these systems. We apply the adapted version of CW to CiteSpace, a KDViz tool that uses bibliometric analyses to create visualizations of scientific literatures. We describe usability issues identified by the adapted CW and discuss how CiteSpace supported the completion of tasks, such as identifying research fronts, and the achievement of goals. Finally, we discuss improvements to the adapted CW and issues to be addressed before applying it to a wider range of KDViz tools.

CR Categories: H.5.2 [Information Systems]: User Interfaces – Ergonomics, Evaluation/Methodology; I.5.4 [Computing Methodologies]: Applications – Text Processing

Keywords: Cognitive Walkthrough, usability inspection methods, bibliographic networks

1 INTRODUCTION

The usability of a software tool can be evaluated at many levels. At the most basic user interface (UI) level, a tool can be evaluated on how easily users can interpret and manipulate UI elements like menus and pushbuttons. Do users understand how the element works and what it does? Are the menu items labeled with clear, unambiguous terms? Usability evaluations at this level are often associated with conformance to user interface standards and conventions [1].

At a more complex level, how easily a tool can be learned can be evaluated [9]. Ease of learning will be affected by the basic UI but also by the match between the system's organization and functionality and the users' knowledge and goals. Evaluations at

this level typically examine how successfully naïve users can use the tool to complete important and realistic tasks. The Cognitive Walkthrough (CW) method is a prominent technique for examining how easily systems can be learned by exploration [5], [7], [14]. CW is especially common when the system is designed for “walk up and use” applications, such as a public website, an information kiosk, or a library card catalog system.

Finally, at the user performance or outcome level, usability evaluations can examine how successfully a tool supports the users in the achievement of goals. Does the tool improve user performance as measured along dimensions like accuracy, speed, or quality? Does the tool help users achieve desirable outcomes, create good products, or make the right decision? Evaluations at this level typically involve empirical testing with users and comparison of their outcomes to current tools or processes or to outcomes generated by an independent experts [1].

1.1 Usability and Knowledge Domain Visualization (KDViz)

In this paper, we use an adapted form of CW to examine a knowledge domain visualization (KDViz) tool. At the basic UI level, does the tool provide UI elements that are easy for users to interpret and manipulate? At the ease of learning level, can naïve users successfully learn to use the tool through exploration? At the performance or outcome level, does the tool allow users to achieve their goals? That is, is the knowledge users gain by using the tool rich, high-quality, and useful?

1.2 CiteSpace

CiteSpace is a KDViz tool originally created for identifying intellectual turning points [3], [4]. CiteSpace creates co-citation networks among highly cited articles. It allows users to manipulate the resulting graphical network in many ways, such as by displaying different time periods and setting various thresholds. CiteSpace has undergone usability evaluations using heuristic evaluation and user testing [13]. Figure 1 shows the CiteSpace UI and the visualization of the social-networking literature discussed in this paper.

1.3 Cognitive Walkthrough (CW)

The CW method is a well-known usability inspection technique in which evaluators examine a system to identify UI problems [7], [11], [14]. Inspection methods are contrasted with empirical user testing in which members of the targeted user community serve as test subjects. Inspection methods are intended to be employed early in a development process and typically require fewer resources than user testing. Usability inspections can be conducted on software prototypes, screenshots, design diagrams, or even sketches on a whiteboard.

¹ email: kra25@drexel.edu; ² sma33@drexel.edu; ³ gkp24@drexel.edu; ⁴ jp338@drexel.edu; ⁵ dns24@drexel.edu; ⁶ mv39@drexel.edu; ⁷ chaomei.chen@cis.drexel.edu

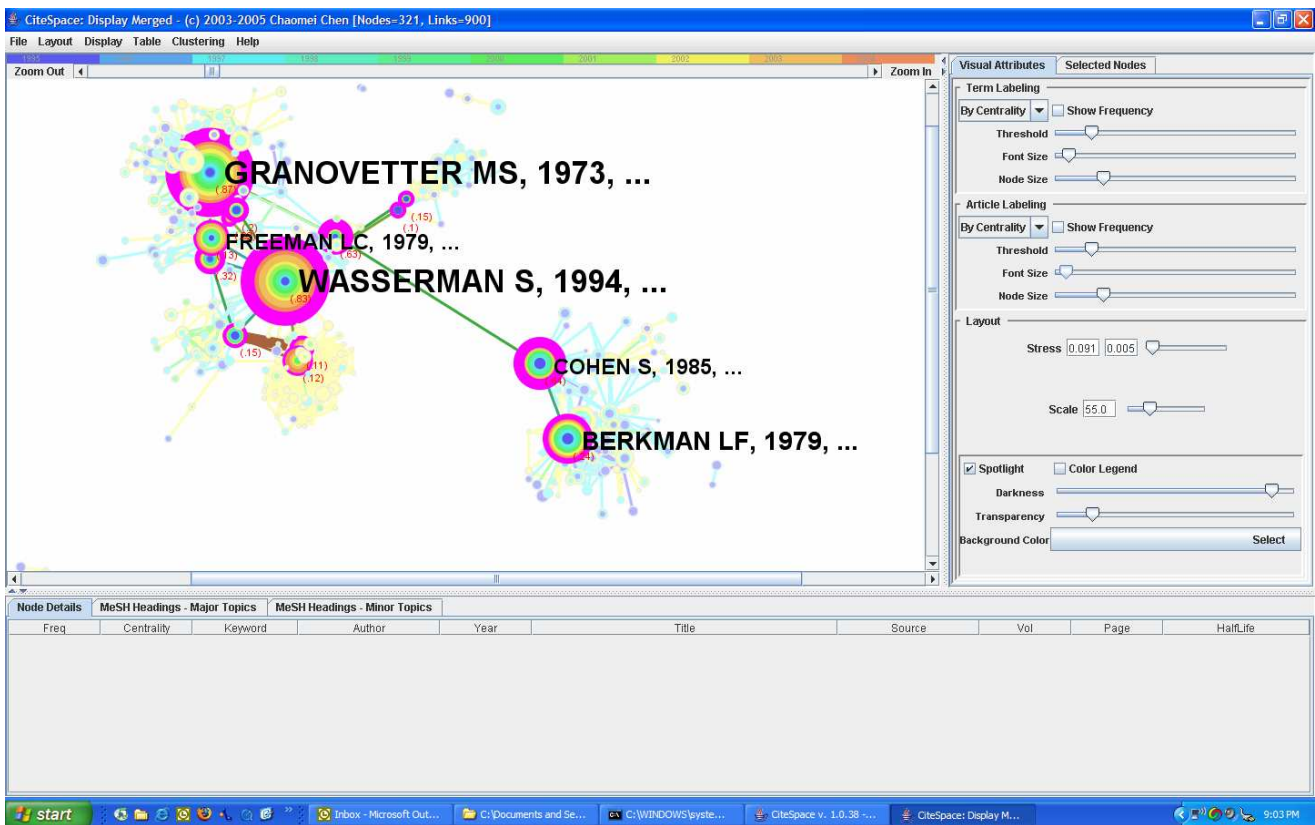


Figure 1. Screen shot of CiteSpace version 1.0.38 displaying a co-citation network of the social networking literature

Among usability inspection methods, CW is appealing from an engineering perspective because its creators have attempted to develop it for use by evaluators who are not HCI experts [7], [14]. Studies have shown that evaluators without HCI backgrounds have some difficulty applying CW but are able to use it to find valid usability problems, though perhaps different ones than HCI experts might find [6], [11]. CW is also appealing because it is methodologically similar to code walkthroughs and use-case models of system development, which allows it to fit with existing software engineering practices [14], [7].

From a research perspective, CW is appealing because it is grounded in theories of how people learn [9]. In particular, CW focuses on how users choose actions based on cues given by a system. Users select actions when the system provides information, such as a label, that overlaps with their current goal states. CW allows evaluators to identify cases where the system provides insufficient information to guide users toward the next correct action. This is especially important for systems that users are encountering for the first time. Other inspection methods like heuristic evaluation are very useful for identifying usability problems and improving products but are less interesting from a research perspective because theoretical considerations are not normally addressed [8].

1.4 Adapting Cognitive Walkthrough

CW is typically applied to systems in which users complete fairly structured tasks in service of well-defined goals. Several well-known CW studies have examined tasks such as forwarding calls in a voicemail system [14], creating and modifying documents in an multimedia authoring system [6], and

categorizing data using a survey analysis tool [11]. In each of these examples, action sequences that describe how to complete each task were prepared beforehand and provided to the evaluators during the CW session. Lewis and Wharton, two creators of CW, recommend that the chosen tasks be *important* to the system and *realistic* with regard to what users would actually use the system to accomplish [7]. They also recommend that the action sequences, which describe how to accomplish the tasks, be *correct* [7]. In CW, a correct sequence is not necessarily the optimal one in terms of clicks, keystrokes, or speed. Rather, a correct sequence is one that accurately represents the designer's intentions for how to complete the task. Several correct sequences may exist for a task and each can be evaluated, though normally only one or two are.

During a CW session, evaluators follow the action sequences and determine how successfully a given user could complete each action. The evaluators create a "success story" or "failure story" at each step. A failure story represents a mismatch between the designer's intentions, as expressed in the UI, and the user. Failure stories form a basis for improvements to the design.

The traditional CW method applies well to many aspects of KDViz tools. KDViz users engage in many structured tasks that lead to achievement of well-defined goals, such as loading datasets, searching online help, and configuring system parameters. Action sequences for completing these tasks have a clear progression and a small number of correct sequences can be generated. Any problems identified for these functions using CW probably can be addressed by consulting UI guidelines and best practices.

However, it is difficult to create action sequences for most open-ended (and interesting) KDViz tasks, such as identifying

connections between research areas. Action sequences in CW represent the designer's intentions for how to complete a task but when a domain is being visualized for the first time, how can designers express their intentions for how a particular visualization of a literature should be used? Even if a known sequence worked for other visualizations of other literatures, designers will not know beforehand how well the sequence will work for *this* visualization of *this* literature with *this* combination of parameters. In addition, depending on the statistical techniques used by the tool to create the visualization, two runs of a visualization on the same dataset may appear somewhat different, as can happen in CiteSpace.

In addition, there may be many possible sequences that lead to successful completion of the task. The correctness of an individual action cannot reliably be determined until after the task is successfully completed or not. A user who takes a seemingly incorrect action may actually be on a path toward success but on a different path than the designer anticipated. This problem exists for other interactive systems also but the number of possibly valid sequences is especially large for exploratory systems like KDViz.

Furthermore, when using KDViz tools, users may be engaged in multiple, evolving tasks simultaneously. To create scripted action sequences where users accomplish one task at a time would not be realistic and would not accurately reflect how the tool is intended to be used nor how users actually use it.

The best approach for creating action sequences for KDViz tools would be to draw from a collection of established sequences that had been found to lead to good outcomes across several visualized literatures, parameter configurations, users communities, and tools. At the time of this study, the research examining KDViz tools and CiteSpace in particular had not produced such a collection of sequences. In lieu of a strong basis for creating new action sequences, we chose to provide the evaluators with only broad goals and accompanying tasks and let them create their own sequences during the CW session.

What might be the effect be of eliminating action sequences from CW? Sears and Hess [11] found that the type of problem identified by CW evaluators was affected by the detail provided in the action sequences. When sequences contained fewer specifics, evaluators found more problems with identifying what action to take next. When sequences contained more details, evaluators found more problems with system feedback. Because KDViz tools are likely to be unfamiliar to many potential user communities, we felt that it was more useful at this point to identify cases where users would not know what to do rather than problems with feedback.

Lewis and Wharton [7] suggest that it may be beneficial to allow evaluators to participate in the creation of the action sequences they will later use. Jacobsen and John [6] asked two evaluators to independently create action sequences for the same system and then used their sequences to conduct a CW. Their evaluators developed different, equally correct action sequences for the same tasks. Their evaluators expressed concern that their self-developed sequences were not in line with the system designer's intentions. Because it is difficult to know what the designer's intentions would be for a particular instance of a visualization, we decided to let our evaluators select their own action sequences here.

2 METHOD

We applied CW to CiteSpace following the well-known chapter *The Cognitive Walkthrough: A Practitioner's Guide* [14] as closely as possible. Areas where we modified the method are described below.

2.1 Participants/Evaluators

The evaluators were six students in an upper-level graduate seminar in information visualization at Drexel University. All were familiar with major information visualization concepts and some concepts of HCI and bibliometrics. None of the evaluators had previous experience using CiteSpace to visualize a scientific literature and none had meaningful experience with the social-networking domain. One (Allendoerfer) had experience applying CW to other interactive systems.

The course professor (Chen), the creator of CiteSpace, served as a technical guide during the CW session. He observed the session and provided guidance on aspects of CiteSpace that were not being evaluated, such as importing the bibliographic data. He rarely stepped in during the CW, even when the evaluators struggled, and only when they asked for help.

2.2 Procedure

2.2.1 Fictive User

The first step in preparing a CW is to create a fictive user who represents a targeted user community [6]. The fictive user is a description of the experience and knowledge that the evaluators assume during the CW. It is the intent of CW that evaluators make decisions based on what the fictive user knows and not what the evaluators themselves know.

One potential user group for CiteSpace is graduate students doing research for class projects or presentations. In these cases, students need to quickly develop a high-level understanding of a literature. We were concerned about the inexperienced evaluators' ability to accurately make decisions as a fictive user who was very different from themselves. For this reason, we created a fictive user who was similar to the evaluators as a group but was not identical to any evaluator individually.

Because we were concerned that creating a fictive user similar to the evaluators could introduce other problems, we followed a recommendation made Jacobsen and John [6] to provide an especially rich fictive user description. In their view, making the fictive user's knowledge very explicit can help evaluators distance themselves from the fictive user. In addition, before beginning the CW, the evaluators were reminded to restrict themselves from offering information that they themselves knew but the fictive user did not. The rich fictive user description is shown in Table 1.

2.2.2 Goals and Tasks

The second step in preparing a CW is to choose tasks to be examined. To ground our task selection, we first developed an overall goal for the fictive user. The fictive user needed to prepare a class presentation and term paper on social networking. The fictive user's first sub-goal was to learn about important and active research areas, authors, and concepts in the social networking literature. The fictive user's second sub-goal was to learn which works from that literature that would be most useful to read.

To accomplish the goal and sub-goals, the fictive user undertakes one or more tasks. For the CW, we identified the tasks as

1. identify important clusters or research areas in the domain;
2. for the important clusters, identify critical authors, terms, and papers that serve to characterize or describe the cluster;

Table 1. Fictive User Rich Description

<p>Education Is a graduate student pursuing a master’s degree in information science. Is currently taking a graduate seminar on information visualization.</p>
<p>Relevant work experience Has 3-5 years experience designing information systems, primarily databases accessed through web browsers. This experience included graphical and user interface design work. This experience also included some programming and administration.</p>
<p>Experience with user interface design and usability assessment The user has taken two courses in human-computer interaction and has designed user interfaces as part of class projects and professional work</p>
<p>Operating systems and software packages used frequently (at least once a week) Microsoft Windows XP; Apple OS X; Microsoft Office (Word, Excel, PowerPoint); Microsoft Outlook; Microsoft Internet Explorer; Mozilla Firefox; Apple iTunes; Microsoft Media Player; Adobe Photoshop; Adobe Illustrator; Macromedia DreamWeaver</p>
<p>Experience with bibliometrics, co-citation analysis and related LIS topics Has been exposed to these topics in readings and coursework. Has conducted several bibliometric analyses as part of class projects.</p>
<p>Experience using CiteSpace or other tools for visualizing literatures/domains Has seen demonstrations of CiteSpace and several other information visualization tools during class but has not used one to create a visualization. Has not used CiteSpace or a similar tool to visualize a literature in support of a class or professional research project.</p>
<p>Experience with digital libraries and databases such as ISI Web of Science, LEXIS-NEXIS, Dialog, etc. Uses Web of Science and ACM Digital Library around weekly for research to support class projects</p>
<p>Experience with social network literature Very little experience with social networking literature. Has heard the term occasionally discussed during coursework and has read one short, general textbook chapter about the field. Has never taken a course, conducted a literature search, or written a term paper about social networking. Does not know names of important authors, papers, or journals in the field. Does not know any “hot topics” in social networking.</p>

3. from the important clusters, identify new and active ones that may constitute a research front or revolution in the domain; and
4. identify important connections between clusters.

We intentionally did not operationalize importance. The fictive user, doing research for a term paper, probably would not have well-defined criteria for what constitutes an important cluster or paper but might “know it when I see it.”

2.2.3 User Actions

The third step in preparing a CW is to create action sequences that reflect the designer’s intentions for completing the tasks. For the reasons discussed above, we did not script any action sequences in this CW. This required changes to other aspects of the CW method. In traditional CW, *before* each user action is taken, evaluators answer four questions [14]:

1. Will the user be trying to achieve the right effect?
2. Will the user know that the correct action is available?
3. Will the user know that the correct action will achieve the desired effect?
4. If the correct action is taken, will the user see that things are going ok?

In our adapted CW, because there were no scripted action sequences, we reworded the questions and the evaluators answered them *after* they selected each action. We modified the four CW questions as follows:

1. What effect was the user trying to achieve by selecting this action?
2. How did the user know that this action was available?
3. Did the selected action achieve the desired effect?
4. When the action was selected, could the user determine how things were going?

To assist with data collection, we created a list of user actions that many information visualization tools support. This allowed the evaluators to select from a list of standardized terms when recording actions. Many of these are derived from Shneiderman’s list of information visualization tasks [11]. Because the evaluators had little experience with CiteSpace but had experience with other information visualization systems, the list was not exhaustive or in any particular order. In fact, several of these actions are not available in CiteSpace but we did not know that when creating the list. The listed actions were:

- (Return to) Overview
- Zoom
- Filter
- Details on Demand
- Relate: Highlight connections between nodes of the network
- Return to previous settings (history/undo)
- Extract: Move selected nodes to another set for further analysis
- Categorize: Place coding information on selected nodes
- Other physical manipulations like pan, flip, rotate, send-to-back, move, compress
- Other appearance manipulations like change color, border, line style, fill-pattern, highlight

2.3 System and Dataset

The CW was conducted using CiteSpace version 1.0.38 [2]. The dataset was a collection of 3,379 bibliographic records retrieved from ISI Web of Science (WOS) and cross-referenced with PubMed. These records resulted from a search “social network analysis” for articles published in 1990-2004 in bibliographic fields such as title, abstract, and keywords. The resulting network contained 303 nodes with 846 links between nodes.

2.4 Data Collection

The evaluators completed each task using a projection of the CiteSpace UI visible to all evaluators. One evaluator (Allendoerfer) served as the “driver” and used the mouse and keyboard on a Windows XP laptop to complete the actions agreed upon by the evaluators. Evaluators were always free to suggest actions but the driver did not execute any action until the others agreed upon what to do, based on the fictive user’s knowledge, tasks, and goals. The projected CiteSpace screen and the audio portion of the evaluators’ discussions were videotaped.

Data were collected in three additional ways. First, one evaluator (Panjwani) served as recorder. After the evaluators chose an action, the recorder noted which action was chosen and recorded answers to the four questions following the consensus of the evaluators. To assist this process, the recorder used a data collection sheet containing the four questions and the list of possible user actions. Second, the evaluators took notes individually on their evolving understanding of the social networking literature, including the names of important articles, authors, and terms. Third, the developer of CiteSpace observed the CW session and took notes. The CW session took approximately two hours to complete.

At the end of the CW, the evaluators expressed their overall conclusions about the social-networking literature and came to a consensus on its important clusters, research areas, authors, and papers according to the four tasks.

3 RESULTS

3.1 Bugs and Usability Design Issues

The evaluators identified three software bugs that had some impact on the evaluators’ ability to complete tasks efficiently. In addition, the evaluators identified 10 areas where CiteSpace could be improved to make it easier to use, easier to learn by exploration, and better support achievement of goals. The design issues judged to have a high or medium impact on usability are listed in Table 2.

3.2 Completing Tasks and Achieving Goals

In this section, we discuss how the evaluators were able to complete the four tasks and achieve the fictive user’s goals. We do this by stepping through the evaluators’ actions and decisions at a high level. In future CWs, this discussion could form the basis of an action sequence.

Before launching the visualization, the evaluators systematically examined the functions on the main configuration window. They examined the contents of each menu and tried to determine what each option did. This sort of investigation is probably not what most users would do first and probably was an artifact of the evaluation environment.

Once they launched the visualization itself, the evaluators immediately identified several works as important: Granovetter 1973, Wasserman 1994, Cohen 1985, Berkman 1979, and Freeman 1979. The evaluators were also immediately able to identify that several clusters were present in the visualization, though what the clusters signified was not apparent. The evaluators set out to determine what the clusters represented.

The evaluators’ first strategy was to select all articles in a cluster and obtain details about them in the Node Details area. By finding commonalities among the articles, the evaluators reasoned they could determine the nature of the cluster. This seemed to be a reasonable and adaptive strategy.

However, due to the nature of the WOS and PubMed databases, this strategy quickly led to bad territory. The visualization was based on WOS data but only articles also indexed in PubMed contained title information. When the selected articles were brought into the Node Details area, only articles with a corresponding PubMed listing provided a title. Not surprisingly, the articles with PubMed listings tended to have a connection to the medical and public health literatures, such as examinations of the spread of HIV. For many minutes, until the quirks of the datasets and the interactions between them were explained by the CiteSpace designer, the evaluators assumed the small-world networks cluster was about medical applications of social networking. This would have been a serious error (see Table 2, Item 1) and highlights a potential problem facing KDViz systems in general.

The evaluators then adopted a new strategy by trying to display the terms associated with the clusters. They spent many minutes trying different options with the Term Labeling controls but were not successful in getting the terms to appear. The designer explained that on the initial configuration window, the terms needed to be enabled and the visualization re-created. Without this explanation, the evaluators may not have reached this conclusion in a reasonable timeframe (see Table 2, Item 2).

Once the terms were enabled and displayed, the evaluators were able to complete each of the four tasks. For Task 1, the evaluators identified six clusters and assigned the following names, based on the displayed terms (directions refer to Figure 1).

- Small World Networks – south of Wasserman 1994
- Individual Differences – northeast of Wasserman 1994
- Transmission Dynamics – southwest of Wasserman 1994
- Silicon Valley – north of Granovetter 1973
- C_i 1/Social Ties – southeast, including Berkman 1979 and Cohen 1985
- Problem Drinkers – far west (not shown)

The evaluators chose the small-world networks cluster to examine more closely for Tasks 2 and 3. As shown in Figure 2, this cluster is very distinct, large, and recent (yellow nodes indicate articles published around 2002). Within the small-world networks cluster, the evaluators identified important papers and authors using several methods. First, by clicking large nodes in the cluster, they identified as important papers: Albert 2002, Strogatz 2001, and Amaral 2000.



Figure 2. The small-world networks cluster

Table 2. Usability Design Issues

Description	Usability Impact
1. In the Node Details area, titles were displayed only if a PubMed listing for the article was available.	<p>High. This limitation of the data sources forced the evaluators to adopt a different strategy than they initially selected. The evaluators initial strategy was to select nodes (articles) within a cluster and read the corresponding bibliographic information (authors, titles). Because titles were not available for most articles in the dataset, the evaluators could not use titles to reliably determine what a cluster is about. Worse, they nearly drew a false conclusion that the small-world networks cluster was related to medical conditions because the only articles with available titles were related to medicine.</p> <p>To address this issue, the limitations of the individual datasets and potential interactions between them must be made clear through help, ToolTips, or other messages. In addition, labels should be enabled by default. If characterizing clusters is a primary task, the labels represent the best strategy within the CiteSpace for understanding clusters. Short of printing out the list of authors in a cluster and then searching for them in a separate database, there is no other reliable strategy. Had the labels been displayed by default, the evaluators may not have selected their initial strategy. A third option is to find additional sources for title or keyword data, acknowledging that these data would support a likely strategy.</p>
2. On the Visual Attributes tab, the Term Labeling controls were functional but terms were not enabled or present in the visualization. This was confusing and led to significant loss of time fiddling with these controls.	<p>High. The controls worked properly once the terms were enabled by using the initial configuration window and re-running the visualization. However, if the evaluators had not sought help from the designer, they might have never concluded that the terms were not enabled. To address the issue, the options for enabling terms could be moved to the same screen as the Term Labeling controls or the Term Labeling controls could be grayed out when terms are not enabled.</p>
3. The evaluators did not identify the capability to drag nodes until over 90 minutes into the session, though it could have been useful earlier. This is an example where CiteSpace did not support learning through exploration. The evaluators did not quickly discover a very useful function.	<p>High. The ability to drag nodes was not addressed in the Help nor was there an indication by clicking or hovering on the nodes that they could be dragged. Only by making mistakes when using other functions did the evaluators uncover the capability to drag nodes. Even then, they did not immediately recognize its function or utility. Once they began dragging nodes, however, the evaluators made significant progress in understanding the literature, especially establishing connections between works and clusters. To address the issue, indications of the drag option could be provided with ToolTips or a cursor change (e.g., four arrows).</p>
4. When viewing the visualization in monochrome, older articles appeared nearly white. Because the background is also white, these articles were overlooked.	<p>Medium. Users unintentionally selected white nodes when trying to click on the background. A viable workaround exists and the evaluators used it: use the color mode. To address the issue, the grayscale could be adjusted so that the lowest value still provides contrast with the white background. Alternately, the background color could be given a color other than white, gray, or black.</p>
5. In the Node Details window, the lines could not be sorted by field.	<p>Medium. The evaluators were trying to identify important authors and wanted to sort the list by author but this could not be accomplished. To address the issue, clicking on the column header could sort the list by that field, as is done in other applications.</p>

Second, the evaluators identified important authors by selecting all the articles in the cluster and looking for names that appeared many times. To do this, they used the marquee selection to select the entire cluster and display details in the Node Details area. Unlike their initial use of the Node Details area, this strategy worked because the WOS data contains author names. Instead, they looked through the author list manually and identified the following important authors: Albert, R., Newman, M., Pastorsatoras, R., and Amaral L.

To accomplish Task 3, the evaluators sought to identify articles that greatly influenced the formation of the small-world networks cluster. These articles, they reasoned, would be highly cited and slightly older than the majority of the cluster. This led to the identification of two papers: Watts 1998 and Barabasi 1999. These papers are connected to other papers throughout the cluster and serve as a bridge to Wasserman 1994. To test the hypothesis that these articles helped launch the cluster, the evaluators re-ran the visualization with only 1995-1998 selected. By doing so, the

small-world network cluster disappeared. The evaluators concluded that small-world networks constituted an active research area and that these articles had launched a revolution in the field.

The evaluators also realized that the link between the Ci_1/Social Ties cluster and the other central clusters did not exist before 1998. To complete Task 4, the evaluators sought to identify how this connection formed. They re-ran the visualization again, including the full time period, and dragged nodes until it was obvious that the connection followed this path: Berkman 1979 to Cohen 1985 to Wellman 1990 to Granovetter 1973 to the rest of the clusters, as shown in Figure 3. The evaluators were not able to establish what occurred after 1998 to make this link so strong but they hypothesized that the small-world network revolution, also launched in 1998, may have begun co-citing these papers and increasing the strength of the connection.

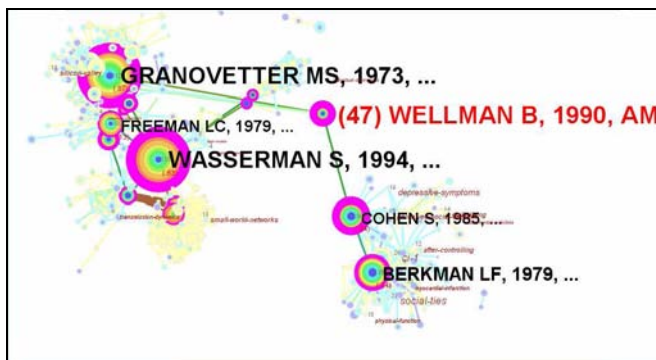


Figure 3. The connection between major clusters

3.2.1 Comparison to Expert

After completing the CW, the evaluators expressed their final conclusions according following the four tasks. We compared their conclusions to that of the CiteSpace designer who has expertise in the social-networking literature. He explained that indeed Watts 1998 and Barabasi 1999 were very important papers that helped launch the small-world networks community. He agreed that other useful works for the fictive user to read would be Wasserman 1994 and Granovetter 1973. The Ci_1/Social Ties cluster is an independent, mostly medically oriented community but who are now connected to the other clusters through a small number of bridging papers like Wellman 1990.

CiteSpace allowed evaluators with no knowledge of social networking to draw relatively sophisticated conclusions that are consistent with an expert in about two hours. This speaks to the ability of CiteSpace to support these tasks and goals. Once users have learned how to use the system, it may take considerably less time to accomplish similar tasks on a different topic. This comparison would be more instructive if the social-network domain expert were completely independent from the project.

4 DISCUSSION

4.1 Appropriateness of CW Method

We found that the traditional CW did not work well for CiteSpace and needed modification, given the current state of knowledge regarding tasks and action sequences for KDViz tools. Action sequences that lead to a good understanding of a domain are inherently tied to the visualized literature, the dataset, and the configurations that users select. Adding or removing a single paper from a dataset or changing the filtering options can dramatically change a visualization and different action sequences could be correct in each case. In this study, we chose not to script action sequences which required that we changed the wording and timing of the traditional CW questions.

With these modifications, we were able to use CW to identify three bugs and 10 usability design issues, several of which were high impact. We were also able to show how that naïve users can learn CiteSpace through exploration and it can support completion of these tasks and goals. However, as discussed above and in section 4.4, the evaluators required assistance from the CiteSpace designer in two cases, suggesting that more refinement is needed.

In completing one CW session without action sequences, we have, in essence, created an action sequence for a visualization of the social-networking literature that, while not necessarily correct, leads to a reasonable outcome. If we wished to run a CW of

CiteSpace using the same literature but using the traditional CW method, the action sequence created here could be used.

4.2 Suggested Improvements to CW for KDViz

We identified several improvements to make CW work more smoothly in future evaluations of KDViz tools. First, facilitating a CW is not a trivial task. Because we allowed evaluators to choose their action sequences, the session became very interactive and collaborative. This was a positive thing but once the evaluators' ideas began to flow, it became hard to rigorously follow the method and answer the CW questions. We recommend that the driver and recorder not also be evaluators. This way they can focus exclusively on ensuring the method is followed carefully and that the data are complete. In addition, the evaluators must be prepared to have their progress slowed by the recorder who must insist on answering each question with each action.

Second, CW was originally designed for use with early versions of systems. For dynamic, exploratory tasks like those examined here, screenshots and sketches will not provide the level of realism and functionality needed. However, it is tempting for evaluators, when conducting a CW using interactive software, to move quickly from one action to the next because the interactive software allows them to. They may lose sight of the questions or the fictive user. To preserve the realism and functionality of the software but also to slow the evaluators down and force them to consider each action, automated tools could be developed that capture user actions and prompt them to answer each question before the next action becomes available. Other studies of CW make a similar recommendation [6] and tools are available but for earlier versions of the CW method [10].

4.3 Supporting Cognitive Tasks

By examining the actions that the evaluators actually chose during the CW session rather than actions scripted beforehand, researchers can begin to see the cognitive tasks that need to be supported by KDViz tools. Because the CW method requires explanations of each user action and these explanations focus on cognitive concepts like knowledge, perception, and goals, CW provides developers with rich design rationales for functions they may consider adding. This CW highlighted the following tasks that should be supported, based on the actions the evaluators took during the session:

- Orienting to the tool (e.g., examining menu items)
- Orienting to the visualization (e.g., non-directed panning)
- Establishing a differential set of items for further examination based on salient features like size, proximity, and color
- Obtaining details on items from the differential
- Manipulating the visualization to highlight connections that may be hidden or obscured
- Hypothesis testing by changing configurations and options
- Revising the differential as understanding evolves
- Describing and organizing conclusions

4.4 Understanding the Underlying Datasets

The most serious problem we encountered during the CW was that an interaction between the databases underlying the visualization led the evaluators toward a serious error. The visualization was based on WOS data that did not include the article titles. To obtain the titles, CiteSpace cross-referenced the WOS listings with PubMed. As a result, only articles appearing in both databases showed title information.

The evaluators first adopted a strategy in which they tried to use titles to determine the nature of a cluster but this strategy could not be supported by the system. The evaluators did notice the large number of missing titles but any concern they felt was not enough to dissuade them from their strategy. Eventually, the designer of CiteSpace needed to intervene. Without his assistance, the evaluators likely would not have achieved their goal or would have reached wrong conclusions.

This problem is a potentially serious one for practical KDViz systems and merits further examination. The bibliographic databases upon which these tools are based have inherent limitations and differences (and will for the foreseeable future). If KDViz systems are intended for use by people unfamiliar with the underlying databases, users will likely not appreciate the capabilities or limitations of the data. When multiple databases are used in conjunction, as CiteSpace does, artifacts and interactions can compound.

In this case, the evaluators continued with a poor strategy because they did not understand how astray it was leading them and an alternative strategy (e.g., activating the terms) was not obvious. If users are not experienced using the underlying databases, the data are too hidden by layers of visualization, or better strategies are obscured, users may not recognize when they are at risk for making this type of error. Future KDViz tools could include functions to alert users when their chosen strategies are not well supported by the system. Alternately, tools could guide users toward strategies that the tool does support and that are likely to result in desirable outcomes.

4.5 Future Work

This study highlighted several areas for future work. First, there are numerous possible fictive users of CiteSpace other than graduate students. However, it remains unclear how well novice CW evaluators can put themselves into the mind of a fictive user who has very different knowledge or goals. One option is to use evaluators who are more similar to the fictive user. Alternately, a study of CW could compare evaluators playing fictive users who are very similar to themselves or very different to examine if the usability problems identified differed by both conditions.

Second, the CW method was created to be conducted early in a development process. In this study, CiteSpace had already been created and undergone many revisions. It would be informative to apply the adapted CW to a KDViz system that exists only as screenshots or prototypes. Would the method identify real usability issues? Could the ability of the system to support performance or outcomes still be assessed?

Third, it would be instructive to compare the evaluators' conclusions, and the effort needed to create them, to other KDViz tools. Does CiteSpace lead to better or worse conclusions given the same visualized domain? Does it lead users to equivalent conclusions but with less effort? In addition, KDViz tools could be compared to alternative methods for achieving the same goals, such as searching on the web or using the bibliographic databases without the visualization.

4.6 Implications for Other KDViz Systems

CW was originally created to serve a system development, practitioner-oriented community. As KDViz systems move from the research laboratory into more practical settings, evaluation methods that fit with existing engineering processes will be needed. We believe the adapted CW described in this paper is a method that could be used as part of an engineering process to examine the usability of a KDViz product. In addition to

identifying usability issues, the adapted CW examines how systems support completing tasks and achieving goals.

The most important lesson of applying CW to KDViz is to focus evaluations and design improvements on users, tasks, and goals. Preparing a CW requires that evaluators be explicit about each. Designers of KDViz systems would do well to prepare for a CW as part of their early design work, even if just as an exercise. The usability of the tool will improve if a designer can clearly articulate who the intended users are, what goals the users hope to achieve, and what tasks they might use to accomplish the goals. By conducting a CW, designers are forced to consider these aspects and that can only be beneficial.

REFERENCES

- [1] Kenneth R. Allendoerfer. An analysis of different methods for writing human factors requirements. *Proceedings of the 2005 Mini-Conference on Human Factors in Complex Sociotechnical Systems*. In press, 2005.
- [2] Chaomei Chen. *CiteSpace: Version 1.0.38* [Computer Software] <http://cluster.cis.drexel.edu/~cchen/citespace/>
- [3] Chaomei Chen. *Information Visualization: Beyond the Horizon*. Springer-Verlag, second edition, 2004.
- [4] Chaomei Chen. Searching for intellectual turning points: Progressive Knowledge Domain Visualization. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, 101 (Supplement. 1), 5303-5310, 2004.
- [5] Gilbert Cockton, Darryn Lavery, and Alan Woolrych. Inspection based evaluations. In *The Human-Computer Interaction Handbook*, Julie A. Jacko and Andrew Sears (Eds.), Erlbaum, Mahwah, NJ, 2002.
- [6] Niels E. Jacobsen and Bonnie John. Two case studies in using cognitive walkthrough for interface evaluation. *School of Computer Science Technical Report CMU-CS-00-132*. Carnegie Mellon University, 2000.
- [7] Clayton Lewis and Cathleen Wharton. Cognitive walkthroughs. In *Handbook of Human-Computer Interaction*, Martin G. Helander, Thomas K. Landauer, and Prasad V. Prabhu (Eds.), Elsevier, Amsterdam, 1997.
- [8] Jakob Nielsen and Robert L. Mack. *Usability Inspection Methods*. John Wiley & Sons, New York, 1994.
- [9] Peter G. Polson, Clayton Lewis, John Rieman, and Cathleen Wilson. Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies*, 36: 741-773, 1992.
- [10] John Rieman, Susan Davies, D. Charles Hair, Mary Esemplare, Peter Polson, and Clayton Lewis. An automated cognitive walkthrough. *Proceedings of the 1991 Conference on Human Factors in Computing Systems (CHI 91)*, 427-428, 1991.
- [11] Andrew Sears and David J. Hess. Cognitive Walkthroughs: Understanding the effect of task description detail on evaluator performance. *International Journal of Human-Computer Interaction*, 11: 185-200, 1999.
- [12] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. *Proceedings of the 1996 IEEE Symposium on Visual Languages*, 336-343, 1996.
- [13] Marie Synnestved and Chaomei Chen. Design and evaluation of tightly coupled perceptual-cognitive tasks in knowledge domain visualization. *Proceedings of the 11th International Conference on Human-Computer Interaction (HCI International 2005)*. In press, 2005.
- [14] Cathleen Wharton, John Rieman, Clayton Lewis, and Peter Polson. The cognitive walkthrough method: A practitioner's guide. In *Usability Inspection Methods*, Jakob Nielsen and Robert L. Mack (Eds.), John Wiley & Sons, New York, 1994.