CoCoNutTrix: Collaborative Retrofitting for Information Visualization

Petra Isenberg and Sheelagh Carpendale ▪ *University of Calgary*

Anastasia Bezerianos ▪ *École Centrale Paris*

Nathalie Henry ▪ *Microsoft Research*

Jean-Daniel Fekete ▪ *INRIA*

CoCoNutTrix extends the NodeTrix social-network-analysis tool to enable multiuser interaction in collaborative environments. A user study verifies the low-cost retrofitting's effectiveness and highlights implications for practitioners.

In human-computer interaction, much of the literature on designing and evaluating colocated collaboration revolves around dedicated technology in the form of touch-sensitive displays, input devices, or software. Each of these has advantages for certain collaboration environments and situations. Adapting an application to colocated collaboration might appear to require using specialized hardware and reimplementing the application, for example, to

- scale to specific presentation spaces such as large high-resolution wall or tabletop displays,
- employ head-mounted displays or CAVEs (Cave Automatic Virtual Environments), or
- react to other forms of input such as direct touch, gloves, or pens.

Combining these approaches—taking large information visualization and analysis systems and reimplementing them to fully incorporate our current understanding of computer-supported cooperative work (CSCW)—is exhausting, time-consuming, and expensive.

We are exploring how to create collaborative information analysis environments cost-effectively in terms of the required hardware and time. We are motivated by the potential benefits of colocated collaboration around data. Sharing a single information display might enable new types of interaction between analysts and enrich existing collaborations. Colleagues can discuss and negotiate data interpretations during collaboration rather than after, they can share expertise and data analysis skills, and peer learning and peer teaching are encouraged.

To create a low-cost collaborative environment, you could use multiple off-the-shelf projectors simply pointed at a blank wall to create a large display, coupled with technical solutions that replace single mouse or keyboard input streams with multiple input devices (for example, JInput; https://jinput.dev.java.net), as Figure 1 shows. However, it is not clear to what extent such a simple approach supports collaborative information analysis, what the requirements and challenges are in practice, or whether a low-cost collaborative environment will support the representations and tasks typically involved in information analysis.

As a first step to answering these questions, we retrofitted a version of NodeTrix,[1] a single-user graph visualization environment based on the InfoVis toolkit, to support multiple independent mice. We call the resulting low-cost environment

0272-1716/09/$26.00 © 2009 IEEE

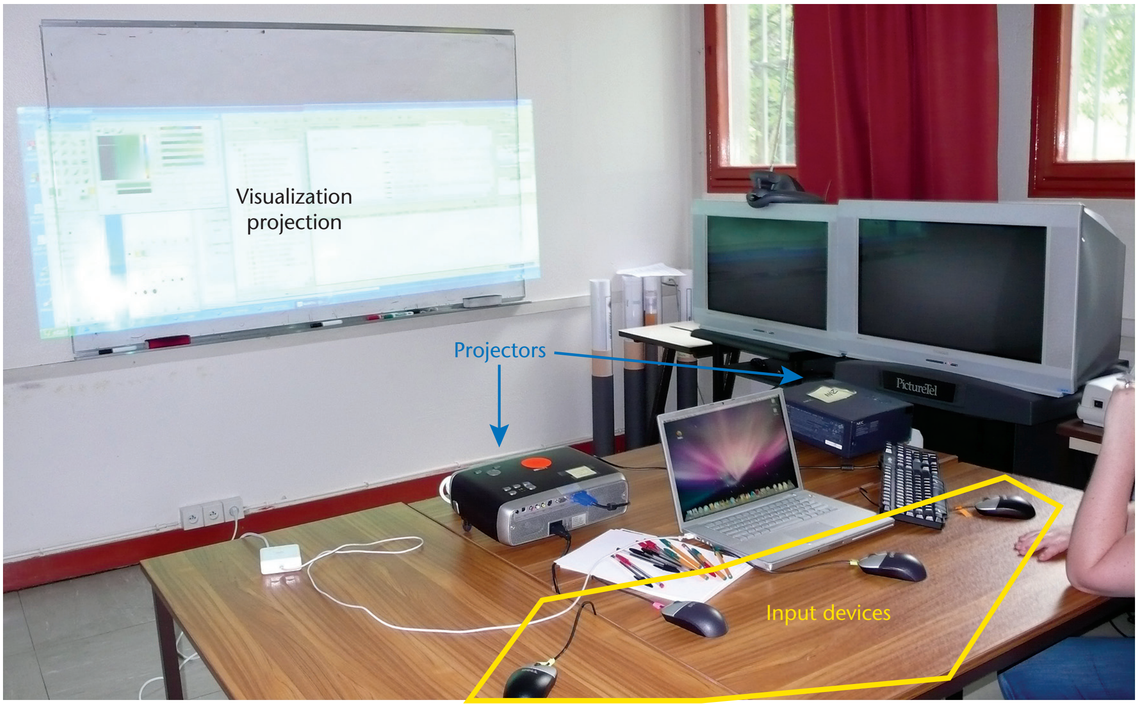

Figure 1. A low-cost setup for colocated collaborative data analysis using four mice, two projectors, and a wall for projection. Our goal was to determine whether such a setup could support effective collaboration.

CoCoNutTrix (Colocated Collaborative NodeTrix). Then, we assessed how analysts viewed CoCoNutTrix and whether it effectively supported collaborative data analysis among domain experts using real data sets for social-network analysis. Our goal is to refine and expand our knowledge about retrofitting and hence designing colocated collaborative-visualization systems.

From NodeTrix to CoCoNutTrix

NodeTrix combines a node-link representation and an adjacency matrix-based representation of a social network in a single view.[1] Analysts can view all data entities as nodes and all internode relationships as links. Or, they can view all data entities as labels in matrix rows and columns and their relationships as the matrix cells. Most important, they can combine the two representations, with part of the data presented in either node-link or matrix form. Analysts can interactively control whether a particular entity in the data appears in either of these representations. For instance, they can group node-link data entities to form a matrix or can select a data entity and drag it into or out of any given matrix. This hybrid visualization offers the benefits of both representations and is conducive to visual data exploration. Figure 2 shows a visualization that displays each research lab in a computer science department as a matrix and connects the matrices by links representing coauthorship relations.

Why We Chose NodeTrix

To explore retrofitting for collaborative information visualization, we wanted to begin with a tool that seemed a promising candidate in its existing state. So, we first looked at the published considerations for information visualization design for colocated collaboration.[2]

Using these design considerations from the literature, we found a promising candidate in NodeTrix, which supports

- *Free categorization of items*. Analysts can group nodes into a matrix using a lasso gesture and can dissolve a matrix with a single click. They can add or remove matrix nodes with drag-and-drop. So, work on a given item can be done independently from work on others. This could support concurrent work.
- *Free workspace organization*. Analysts can freely reposition data items. This lets them work on the task in different areas of the display.
- *Individual viewing preferences*. By making local changes in the representation, analysts can adapt parts of it to their own preferences.
- *Fluid interaction*. The number of changes of input modality, the manipulation of interface widgets, and dialogs are minimized, which can improve coordination of activities in a group.
- *Focus on mouse interaction*. Analysts use the mouse to perform almost all actions, which makes NodeTrix open to retrofitting for multiple inputs. Only three tasks require a keyboard: typing labels, triggering redo or undo, and activating a graph re-layout.
- *Minimal global changes*. NodeTrix includes only two main choices for global changes (undo/redo

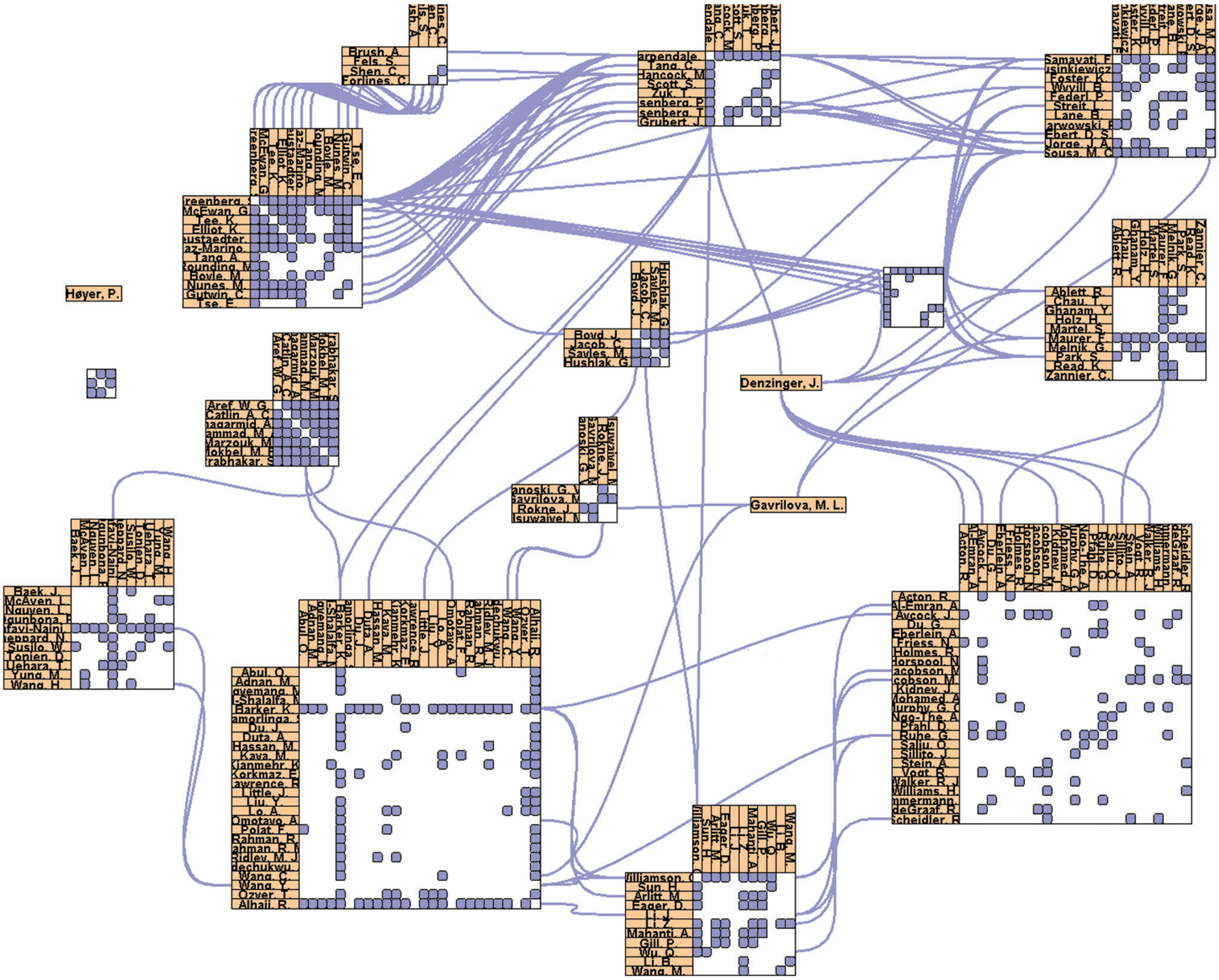

Figure 2. A NodeTrix visualization integrating node-link and matrix visualizations. This image shows the coauthorship network of a university computer science department, in which research labs have been grouped into matrices.

and graph re-layout). This reduces the likelihood of accidental changes that affect all users, thus potentially leading to less interruption of the group work.

In addition, several practical aspects made NodeTrix a good candidate for our work. Experts have used it successfully for social-network analysis, and it has proven useful in single-user work.[1] We also had access to the underlying source code and could make necessary adjustments to introduce concurrent inputs.

However, NodeTrix does not specifically support some of the design considerations from the literature. It does not support communicating findings or discoveries, solving interaction conflicts, preserving a graphical history, or maintaining individuals' awareness of each other's efforts. So, although NodeTrix presents a promising starting point, it is not clear whether it will help group members collaborate effectively.

So, for our observational study, we were interested in five questions. Does our software enable communication between analysts? Do interaction conflicts occur that hinder collaboration? Can group members stay aware of each others' work? Are group insights achieved? What is the qualitative collaborative-analysis experience, and what does it reveal about the system?

Implementation Details

When developing CoCoNutTrix, we kept our reimplementation choices to a minimum. Whenever possible, we left things as they were, because we aimed to study whether a minimal retrofit would accrue collaboration benefits.

General collaboration support. One challenge in redesigning software for collaboration is to minimize global changes to avoid interrupting group work. Yet many information visualization systems, including NodeTrix, offer numerous parameters for changing visualization output. For CoCoNutTrix, we turned off menu bars and control panels and chose default values for all visual features such as link width, color, or label size that were appropriate for our task and data set.

Because the main current operating systems do not support having multiple windows in focus at the same time, we provided a full-screen visualization environment that allowed no accidental resizing, repositioning, or change of focus of application windows. Because we had already turned off the control panels, we achieved this by giving all available screen space to the rendered visualization. In applications requiring multiple windows, widgets, or dialogs, these would likely have to be reimplemented.

Adding multiple inputs. In NodeTrix, because the mouse is the most common interaction mode, we gave each collaborator a mouse. Because keyboards are used for only the three relatively rare tasks we mentioned earlier and take up much physical space on the table, we provided one shared keyboard.

To capture independent input from any attached mouse, we used the JInput library and added a GlassPane—a transparent panel—on top of the application to render the additional mouse cursors and dispatch modified mouse events to the application. We derived a new mouse event class that carried individual mouse IDs in addition to the traditional mouse event data. These IDs were necessary so that CoCoNutTrix could react to user-specific input. For example, we added user-specific data structures to keep track of which items were being drawn or dragged by which mouse. For instance, to capture the lasso gesture, we needed to save a mouse path for each user.

In keeping with the spirit of changing as little as possible and because researchers have suggested that social protocols are often a successful conflict resolution method,[3] we left conflict resolution to the participants.

For a closer look at the challenges of adding multiple inputs, see the “Retrofitting for Collaborative Information Visualization” sidebar.

Changing representation and interaction. We made three changes to visual representation and interaction.

First, we provided additional visual feedback. To differentiate the available mice, we enlarged each cursor and assigned it and its mouse the same unique color. Clicking or dragging with a mouse created a similarly colored glow on each affected node or matrix. To achieve this effect, we extended the rendering code for nodes and matrices and rendered a colored semitransparent rectangle on top of them.

Second, we changed keyboard input for matrix labels. Previously, users created labels by selecting a matrix and typing the desired text. In a multiuser case, several matrices might be in focus. So, it is unclear to which one a label should be added once a user starts typing. To circumvent this problem, we created a new label object, representing the label text. CoCoNutTrix added this object to the visualization after a user finished entering text. The user could then drop it on a matrix to label the matrix.

Finally, we transferred functionality to the mice that was previously available through menu selection and control panel interaction. To allow zooming in on and out of rendered matrices, we mapped the resizing action to the mouse wheel, a simple fix to address the mouse focus problem (that is, when several mouse cursors are holding objects, it’s not clear on which object the system should focus). CoCoNutTrix uses *interactor* objects to implement all the interactions, which decouples the interaction from the visualization rendering and from the application logic. This feature, part of the InfoVis toolkit, made retrofitting easier.

Retrofitting cost. Estimating this cost is difficult because the estimation relies on the developers’ knowledge of the underlying code and the number of places to edit. The InfoVis toolkit contains approximately 750 classes and 65,000 lines of code (65 KLOC); NodeTrix contains approximately 50 classes and 10 KLOC. To retrofit NodeTrix, we created 10 classes and wrote fewer than 1 KLOC. We extended only the classes in charge of the interaction and created classes to detect and draw the multiple mice. The retrofitting was conducted mainly by one researcher, who was expert in Java but new to the application and toolkit. The retrofitting took one full week, with help from the main developers of NodeTrix and the InfoVis toolkit.

Study

We studied groups of four experts performing social-network analysis using data from their own organization. Our participants were experts in the data, not in social-network analysis. To ensure that our collaboration setup was effective in different realistic settings, we conducted the study in three organizations, using each organization’s existing technical facilities. Organization A was an educational institution; B and C were research organizations.

Retrofitting for Collaborative Information Visualization

Connecting several mice, keyboards, or other input devices to one desktop computer can be problematic, owing to support issues at four levels: operating systems, libraries, graphical toolkits, and applications.

Some operating systems such as Windows explicitly limit support for multiple mice and keyboards owing to security issues. Others (including Linux, most Unix flavors, and Mac OS) allow the management of extraneous input devices, but with a different level of support than for standard input devices. For example, these systems do not provide cursor feedback for extraneous positional devices, so applications or window-manager extensions must provide this capability.

Low-level libraries for access to USB devices or game devices allow the reading of input devices in system-dependent ways. Recent years have seen some progress in standardizing access to these libraries with projects such as JInput for Java (https://jinput.dev.java.net). These libraries raise issues because the window manager applies many hidden operations to the standard input devices (for example, acceleration management for relative-positional devices, and key mappings for keyboard devices). Emulating these operations through external libraries is difficult or impossible, except when the libraries are integrated with the window systems (for example, the X Input Extension; http://en.wikipedia.org/wiki/DirectInput).

Graphical toolkits such as Swing for Java or Qt for C++ support GUI components (widgets) and input management. Like most toolkits, they manage only a limited set of input devices through typed events. Even for well-supported devices such as the mouse, they usually do not support more than one device predictably. Only recently have researchers tried to support multiple input devices at this level.[1–3]

Applications such as MMM[4] (Multi-device Multi-user Multi-editor) that support colocated collaboration have been built from scratch owing to the lack of toolkit and library support. However, the developers of newer generations of colocated applications have been either trying to build toolkits or relying on special toolkits to simplify these applications' designs.

Some researchers have reported on retrofitting single-user applications for collaborative use. However, only a few have specifically studied this in relation to colocated information visualization and considered the implications of offering multiple independent inputs.

Clifton Forlines and his colleagues describe collaborative retrofitting of Jmol for molecular visualization[5] and Google Earth.[6] They adapted both tools for a multiuser, multidisplay environment. Their research describes how they adapted the visualization for viewing and interacting in a colocated scenario, using different views on different display configurations. They solved concurrent-input problems through a single-user floor control policy that lets only one person interact with a display at a time.

Comparing distributed and colocated information visualization, Gloria Mark and Alfred Kobsa studied collaborative use of existing information visualization tools.[7] They found that group performance increased with system transparency. Collaborative retrofitting for this study was minimal. Although participants in the colocated setting used a large shared display, they shared a single input.

Some graphical toolkits managing scene graphs[8] or information visualization[9,10] use the *interactor* abstraction to implement modular interaction techniques. They decouple display management and interaction, simplifying the retrofitting for multiple inputs. Moreover, they support a layering mechanism on which to draw additional cursors and add highlighting without interfering with the standard display management. One such toolkit is the InfoVis Toolkit,[9] which was the basis for our NodeTrix visualization environment and its extension for collaborative visualization, CoCoNutTrix (see the main article).

References

1. J.P. Hourcade, B.B. Bederson, and A. Druin, "Building KidPad: An Application for Children's Collaborative Storytelling," *Software: Practice & Experience*, vol. 34, no. 9, 2004, pp. 895–914.
2. P. Dragicevic and J.-D. Fekete, "Support for Input Adaptability in the ICON Toolkit," *Proc. 6th Int'l Conf. Multimodal Interfaces* (ICMI 04), ACM Press, 2004, pp. 212–219.
3. S. Huot et al., "The MaggLite Post-WIMP Toolkit: Draw It, Connect It and Run It," *Proc. 17th Ann. ACM Symp. User Interface Software and Technology* (UIST 04), ACM Press, 2004, pp. 257–266.
4. E.A. Bier, S. Freeman, and K. Pier, "MMM: The Multi-device Multi-user Multi-editor," *Proc. SIGCHI Conf. Human Factors in Computing Systems* (CHI 92), ACM Press, 1992, pp. 645–646.
5. C. Forlines and R. Lilien, "Adapting a Single-User, Single-Display Molecular Visualization Application for Use in a Multi-user, Multi-display Environment," *Proc. Working Conf. Advanced Visual Interfaces* (AVI 08), ACM Press, 2008, pp. 367–371.
6. C. Forlines et al., "Multi-user, Multi-display Interaction with a Single-User, Single-Display Geospatial Application," *Proc. 19th Ann. ACM Symp. User Interface Software and Technology* (UIST 06), ACM Press, 2006, pp. 273–276.
7. G. Mark and A. Kobsa, "The Effects of Collaboration and System Transparency on CIVE Usage: An Empirical Study and Model," *Presence*, vol. 14, no. 1, 2005, pp. 60–80.
8. B.B. Bederson, J. Meyer, and L. Good, "Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java," *Proc. 13th Ann. ACM Symp. User Interface Software and Technology* (UIST 00), ACM Press, 2000, pp. 171–180.
9. J.-D. Fekete, "The InfoVis Toolkit," *Proc. IEEE Symp. Information Visualization* (InfoVis 04), IEEE CS Press, 2004, pp. 167–174.
10. J. Heer, S.K. Card, and J.A. Landay, "Prefuse: A Toolkit for Interactive Information Visualization," *Proc. SIGCHI Conf. Human Factors in Computing Systems* (CHI 05), ACM Press, 2005, pp. 421–430.

Table 1. The physical study setup in the three organizations.

Organization	Screen size	Resolution	Projector setup	Viewers' distance from the display	Figure
A	1.46 × 1.1 m	2,048 × 1,536	4 projectors in a 2 × 2 setup	1 m	3a
B	4 × 1.5 m	2,560 × 1,024	2 projectors in a row	1.5 m	3b
C	2 × 0.8 m	2,560 × 960	2 projectors in a row	2 m	1

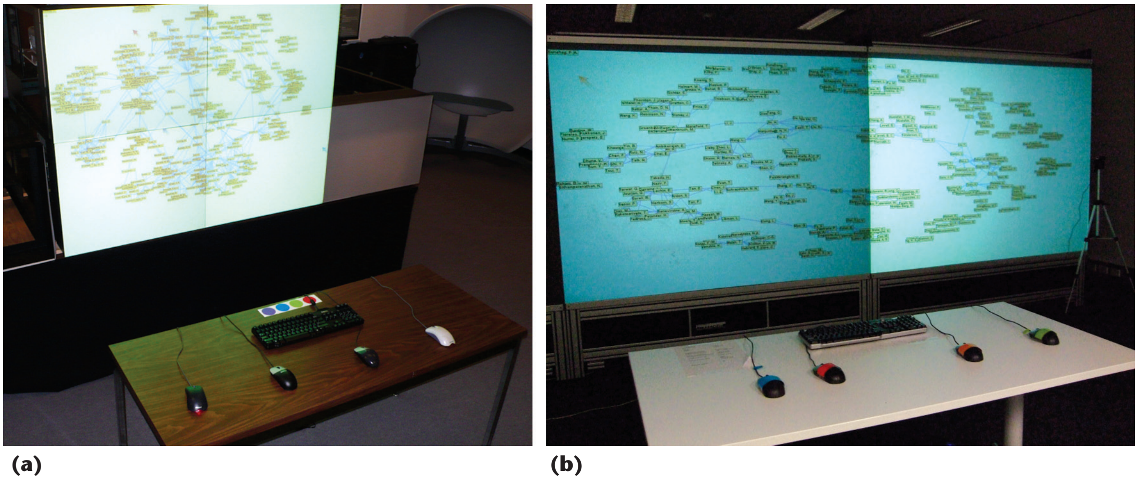

Figure 3. The physical setups for (a) organization A and (b) organization B. (Figure 1 shows the setup at organization C.) To study our system in different realistic settings, we used each organization's available display and computer resources.

Social-Network Data

The three organizations have an interest in determining how their internal research groups collaborate and how effective these collaborations are. So, we decided to use research collaboration social networks as data for our study. Given that research publications are a good indication of collaboration, we used each organization's coauthorship network as a data set. Authors in the data set became network nodes, and coauthorship relationships became links. Each institution had many authors (more than 800 in each of the three), making the analysis difficult to complete in less than one hour. We wanted to ensure that we could conduct a complete experimental session in approximately 1.5 hours, thus making it easier to recruit knowledgeable experts with limited available time. So, we filtered out authors with a low number of publications. This resulted in 423 remaining authors for organization A, 327 for B, and 430 for C.

Participants

All 44 participants (14 female) had been with their organization for at least six months and were experts in part or all of the social network we asked them to analyze. Their positions included senior professors and researchers, group and project leaders, administration, human-resources personnel, technical personnel, and a few graduate students. We divided the participants into 11 groups, each with four participants—four groups each from organizations A and C, and three groups from B. To ensure a realistic, comfortable collaborative setting, participants were either work collaborators or friends. All participants except one reported being familiar with their group.

Apparatus

Resources in the organizations differed slightly, but we tried to keep the settings as similar as possible. The same visualization software ran on a dual-core 3 GHz CPU, with 2 Gbytes of RAM, running Windows Vista. Table 1 gives the setups' physical details. Figure 3 shows the setups at organizations A and B; Figure 1 shows the setup at C.

Task

We presented the participants with a visual representation of a social network about which they had intimate knowledge in terms of actors (researchers), their roles and positions in the organization, and their working relations. We asked them to create a representative view of the researchers in their organization that could later be printed as a poster. We gave them a single shared network representation using a force-directed layout (LinLog[4]). They then had to identify and name the different communities, defining their own criteria. Such an open-ended task of identifying communities and examining their connections is common in social-network analysis.[5] (For more on social-network analysis, see the related sidebar.)

Procedure

We first asked the participants to complete a brief questionnaire eliciting their background, their

Social-Network Analysis

Any collection of persons or organizations connected by relations is a social network. In the last decade, social-networking applications' popularity has dramatically increased. Intelligence agencies use social-network-analysis tools to monitor terrorist networks, epidemiologists use them to study transmission networks and to detect and contain disease outbreaks, and company managers and research institutes use them to examine communication flow between their employees and the strength of their employees' collaboration. We focus on exploratory visual analysis of social networks.

With the increasing popularity of social networking and progress of Internet technologies, many systems have emerged to visualize and analyze social networks. (For an overview, see www.insna.org/software/index.html.) The most common representations are *node-link diagrams* and *matrix-based representations*. Node-link diagrams are commonly used to understand the network's global structure, whereas matrices can improve readability for detailed community analysis.[1]

Trial demonstrations of social-network-analysis software have given us empirical evidence of spontaneous analysis sessions of colocated colleagues coming together over a small shared display to make sense of, discuss, and explore their data. Jeffrey Heer and Danah Boyd reported similar observations when studying Vizster, a visualization tool for online social networks in a public setting.[2] Social-network analysis can benefit highly from collaborative analysis through the combination of knowledge, expertise, and skills as well as the combined cognitive power of several analysts tackling larger networks together.

References

1. N. Henry, J.-D. Fekete, and M.J. McGuffin, "NodeTrix: A Hybrid Visualization of Social Networks," *IEEE Trans. Visualization and Computer Graphics*, vol. 13, no. 6, 2007, pp. 1302–1309.
2. J. Heer and D. Boyd, "Vizster: Visualizing Online Social Networks," *Proc. IEEE Symp. Information Visualizaton* (InfoVis 05), IEEE CS Press, 2005, pp. 33–40.

familiarity with the rest of the group and with the data set, and their experience in using social-network software. We then introduced them to CoCoNutTrix and let them experiment with it for 15 to 20 minutes on a training data set. Once they felt comfortable using the system, they performed the main task of organizing and labeling their organization's coauthorship social network. The task ended either when they finished it or when they reached the 40-minute mark. After a short break, the entire group took part in a semistructured interview eliciting their opinions on the task and the system. An experimenter was present throughout the study to answer any questions.

Data Collection and Analysis

Besides the pretrial questionnaire, observations, and interview, we collected other data for later analysis. We videorecorded all sessions from two distinct locations, focusing on both the participants and the screen. We also stored detailed system logs for each session. Finally, a note taker made detailed observations of participants' use of the system and social interaction. We then combined all our data and created affinity diagrams to reveal patterns in the data.

Results

We group our results according to the mechanics of collaboration[6]—that is, the low-level actions and interactions that a collaborative system must support for group members to complete a task in a shared manner. We include findings relating to our understanding of effective collaborative data analysis. Similarly to Carl Gutwin and Saul Greenberg,[6] we consider a collaboration to have been effective when the participants successfully completed the activities and no major errors or conflicts arose.

Explicit Communication

In face-to-face settings such as ours, most explicit communication is verbal and is the main means to establish a common understanding of the task at hand.

Observations. We observed frequent verbal communication: in nine groups, lively communication arose around the data's content. We observed two types of explicit communication. *Running commentary* was common when participants wanted to quickly inform others of an action performed or planned without intending to start a conversation. *Direct discussions* directly contributed to social knowledge building. Groups exchanged rationale and argumentation regarding actor placement or grouping choices. Group members would agree, disagree, and negotiate, building a shared understanding of the analyzed network.

Because participants were not directly interacting with the display, our system needed to facilitate deictic references and gesturing for communication. Participants performed deictic references by not only pointing with their hands at the display and making verbal references but also gesturing and pointing indirectly with their cursors. Moreover, they repurposed the system for their communication needs—for example, by enlarging an object to attract attention. During phases of joint visual attention, participants commonly moved the cursors to the joint focus area to show that they were focusing on a specific information item under discussion.

Requests for improvement. Participants requested additional features only to support deictic references. Three groups asked for a visual feature, such as a user-controlled glow or animation, that could explicitly draw the group's attention to a particular cursor.

Summary. Our system adequately supported intentional verbal communication, facilitated mostly through the face-to-face setting. Participants creatively used the visual representation to perform deictic referencing, with few participants asking for better support. One goal of collaborative information visualization tools is to let groups come to a common understanding of the data. Through our observations of instances of explicit communication, we are quite confident that CoCoNutTrix met this goal.

Consequential Communication, Monitoring, and Group Awareness

Collaborators and artifacts in physical collaborative settings can unintentionally transmit information—for example, through hand movement or the rustling of papers. Such consequential communication is also important in digital collaborative tasks because it is the primary mechanism for creating group awareness of what is going on, who is working on what, and where others are in the workspace.

Observations. We observed four main visual features through which the representation mediated consequential communication and enhanced group awareness.

The first feature was color coding. The combination of uniquely colored cursors and matching coloring of selected artifacts provided a single, explicit awareness mechanism. This color coding indirectly indicated to participants areas of the display and specific artifacts on which others were focusing.

The second was labeling. Participants labeled communities to indicate that they had been analyzed or needed further work, implicitly informing the group of the work to be done. For example, in nine groups, participants named a community only when they felt it was reasonably finalized. The other two groups assigned unknown or unfinalized communities a predefined default name (for example, "unknown 1").

The third was location. Participants implicitly communicated their decisions regarding communities by placing them at predefined areas of the display. Two groups placed finalized communities on the display's periphery, while two other groups used a predefined area of the screen for "unknown" or "draft" communities. Although in most cases this placement was initially unintentional, it often became an explicit work practice (for example, "I am putting unknowns to the right").

The last feature was scale. Six groups shrank matrices representing finished groups to communicate that they should not be edited further.

Participants generally reported having been aware of group work with the visualization. Interestingly, several participants stopped interacting for moments at a time and raptly watched the representation. When asked about this behavior in the interview, they reported they did it to gain an overview of what had changed in the data set, what the group strategy was, and what areas they could work on next.

Requests for improvement. A known issue regarding awareness is that users easily lose their mouse cursors on large displays.[7] Participants in six groups reported losing their cursors occasionally, even though we had increased cursor size to four times that of the standard Windows desktop and given each cursor a distinct, bright color.

During the interview, five groups also asked for more-explicit ways to label and annotate their work to ensure that decisions would not get lost in the process (for example, changing communities' colors to indicate they are finalized and giving matrices specific labels such as "do not merge!").

Only participants in four groups requested a feature for viewing the group's interaction history, to see each other's actions and a specific network area's history.

Summary. Although our participants were able to collaborate on the retrofitted setup, half of them felt the colored cursors did not provide enough awareness of other users' actions. Some participants also requested annotation functionality to mark the state of communities. However, most felt that although they missed the ability to perform detailed actions, they were globally aware of the group process and progress. Perhaps participants did not frequently request an interaction history owing to the study's task and length. We generally saw the participants using the visualization itself as the medium to indirectly capture, represent, and communicate the group understanding and knowledge of the communities in the data set.

Action Coordination, Assistance, and Protection

An important part of effective, fluid collaboration is how collaborators mediate their actions and share common workspace resources. People organize their actions to avoid conflict with others and efficiently complete their task.

Observations. Our participants clearly organized their actions to not conflict with others. They achieved this by either explicitly dividing the task and workspace through verbal communication or observing where others were working. Collaborators worked predominantly individually or in pairs in different areas of the workspace, moving fluidly between closely and loosely coupled work styles. In every group, when questions arose or global changes had to be negotiated, the participants came together and evaluated a solution, performing coordinated actions on the workspace. Coordinated actions were also common when participants helped each other out. Participants would either request such peer aid (for example, "Could you remove X from that community while I ...") or volunteer it when observing others' actions (for example, "Let me do that").

In groupware systems, accidental conflicts of concurrent input can be disruptive; researchers have suggested special control mechanisms.[6,8] Because we chose not to provide any conflict control mechanism, we logged potential sources of interaction conflicts to validate this choice. These included two or more participants grabbing the same node or matrix or one participant trying to lasso an item on which another participant was working. Such conflicts were rare. Ten groups logged only two conflicts, with concurrent dragging being the most common (four instances overall). One group had seven such conflicts, caused mostly by two people interacting with the same matrix concurrently. When the participants discussed this in the interview, none of them perceived the logged conflicts as problems. We also observed conflicts dealing with inadvertent dropping of elements in matrices or a participant editing matrices after others considered it finished. All these conflicts were solved socially, and some groups even established rules (for example, "Ask before editing a reduced-size matrix" and "If you see labels, do not touch them"). When interviewed, participants felt these conflicts were easily solved and did not interfere much with the task.

Requests for improvement. When asked whether they would have wished for a mechanism to lock control or indicate ownership of items, all but one group responded negatively.

Summary. Our participants coordinated their actions very fluidly. We feel that our choice to not include specific protection mechanisms was further justified because the participants resolved conflicts socially and mistakes could be easily reverted through local or global undo.

Analysis Strategy and Group Insight

One indicator of successful collaborative visualization is the establishment of an effective strategy leading to group insight. Group insight is difficult to measure but can be visible in interactions between participants, interactions with the visual representation, or interview comments such as "we found out that ... ."

Observations. Although our environment provided no explicit planning support, most participants verbally negotiated their strategies. Nine groups started the task with a short group-exploration phase in which participants identified initial obvious clusters. The establishment of an analysis strategy seemed to evolve naturally from conversation and participants observing each others' actions.

When asked, all 11 groups reported gaining insight from working with the data set. They reported several confirmatory or surprising findings, such as close collaboration patterns between research groups previously thought unconnected. They also reported findings about their close working environment—for example, "I had no idea that many people were collaborating in our lab; I even learned things about my own team!" Peer teaching and learning of these insights occurred often in groups with an imbalance of shared knowledge. In one group, for example, a participant helped identify the initial communities and taught others about parts of the data set they were unfamiliar with, so that the work could then commence in parallel.

Summary. Participants smoothly established an analysis strategy and did not request additional features for activity planning. Observations and comments showed that our tool helped the group gain insight, helped participants teach each other facts about the data, and supported knowledge building.

We see this as an important part of a successful collaborative data analysis environment.

Work Preference

As an indication of successful collaboration, we asked participants whether they preferred conducting this analysis task as a group rather than individually.

Observations. Forty participants preferred group work; four preferred to do the task alone. Three of the latter were among the most-knowledgeable members of their group and felt they could have done a reasonable job on their own, although they admitted it might be slower. The fourth had a completely different opinion than the rest of her group about what criteria to use in forming communities.

The participants who preferred group work gave these reasons:

- shared knowledge (27 participants),
- the fun of collaboration (25),
- the shared process of forming consensus (6),
- the opportunity to brainstorm (4),
- efficiency (4), and
- shared working styles (1).

One participant commented that "doing it with three people was fun; doing it by myself would be work." In addition, nine groups reported feeling happy with the result of their analysis and the communities they had identified.

Requests for Improvement. Most participants stated that additional time and meta-information would have helped resolve questions about unknown people and improve the analysis's visual presentation.

Summary. Groups were generally very happy with their collaboration and the result of their work. We take this as an indication that the retrofitting was successful for this setting and task and could effectively support collaborative data analysis as perceived by these participants.

Reaction to Low-Cost Environment Choices

Observations on the environment's usability can further reveal the retrofitted tool's effectiveness.

Observations. One strength of the CoCoNutTrix visualization was its intuitive interaction. All participants at some point interacted with the information items, and all participants moved their mice simultaneously for extended periods of time. Participants were comfortable interacting anywhere on the screen; the different sizes of the screens at each organization did not affect this observation. Typically, one user in each group acted as a dedicated scribe, using the keyboard to input the agreed-upon labels for the communities. Groups rarely used features that would have created global view changes (undo, redo, and graph re-layout); when they did, it was generally after negotiating and obtaining group approval. Five groups never used these functions, two groups used them six times, and the remaining groups used them two to three times. Participants commented that our low-cost setup of mouse input and large screens supported their group work well.

> ***One participant commented that "doing it with three people was fun; doing it by myself would be work."***

Requests for improvement. Three groups expressed the need for a second keyboard to avoid interrupting others' work processes by asking for a label or handover of the keyboard. Overall, there were 15 requests for functionality that was originally part of NodeTrix but was removed during retrofitting. These requests were mostly for visual features mentioned earlier (such as highlighting or more metadata) or for additional interactions on matrices (such as sorting). Participants reported they did not feel the sitting configuration influenced their collaboration. However, some participants would have preferred a slightly curved seating arrangement so that they could talk to each other more easily. Organization C's participants, who had to deal with a larger network on a slightly smaller display, would have preferred a larger display or functionality to "push nodes to get more space." So, the participants in that organization perceived the ratio between the display and network size as a threshold condition for comfortable use.

Summary. Although participants requested additional functionality for the system and physical setup, they generally reported being well supported in their global task. The lack of interaction capability and metadata affected their work efficiency, but the work quality was not generally compromised. We see this as proof that our discounted interface was a good compromise for this task.

The requested additional visual and interaction features are difficult problems to solve when multiple users interact with the system. Selection actions can induce input conflicts, and parameterizing actions require consensus because they affect the entire representation. That is why we removed them in our retrofitting, but further research is necessary to reduce global changes in visualizations or make them less disruptive. Although the actual sitting position did not seem to interfere with the collaboration, we found that the display size was important. Finding the optimal screen size for visualization tasks requires further research.

Assessing the Results

To summarize our findings, we return to our initial questions regarding utilization of our retrofitted collaborative software.

The visualization evolved and became an archive of the participants' process.

Does CoCoNutTrix Enable Communication?

We observed frequent interaction between analysts, with the data, and with the visualization. Analysts slipped in and out of interaction with the full group and with varying subgroups as work progressed. This confirms previous CSCW studies on information visualization in other settings that observed frequent switching between loosely and closely coupled work.[9,10] Active data interpretation, discussion, and negotiation occurred throughout the collaboration while participants interacted on all areas of the display. This finding is important because information visualization analysis requires seeing and interacting with all parts of the representation to explore all available data and avoid misleading or incomplete data analysis.

Do Interaction Conflicts Hinder Collaboration?

The occurrence of few interaction conflicts between participants echoed previous findings that people naturally avoid interfering with each other by spatially separating their actions in the workspace.[3] Moreover, because almost no participants requested any additional control mechanism features, our decision to leave them out was further justified.

Are Group Members Aware of Each Other's Work?

The visualization mediated the awareness of decisions made about the data and helped group members build on each others' work. Factors such as labeling helped the group coordinate which data aspects had been decided on and which were still in flux. Yet, participants asked for several additional awareness features; this is a promising direction for further collaborative-visualization research.

Are Group Insights Achieved?

The visualization's hybrid nature helped in facilitating, and hence observing, group insight because it captured the evolving construction of knowledge in the group. Participants did not view a matrix as simply a different representation of a group of researchers in the data set. A matrix expressed a particular research group and, together with a label, became the result or artifact of choices made by one or several participants during the collaboration. This artifact was then visible to others and facilitated the emergence of a common understanding of the data in the group. So, the visualization evolved and became an archive of the participants' process—what work was completed or needed discussion—and of the participants' insight—the interpretations and meaning they together had given to specific information in the data set. Gerry Stahl has made similar observations for collaborative communication and learning in online communities.[11]

What Does Qualitative Feedback Reveal?

Participants positively received both the physical environments (the large back-projected display and sitting arrangements) and the use of multiple mice for interaction. Together with other positive responses and feedback regarding the system's usability, we feel confident that we sufficiently retrofitted NodeTrix to enable effective collaboration.

Implications for Other Information Visualization Systems

Our results have implications for other information visualization researchers or designers considering how to adapt their own single-user applications to colocated collaborative work. Because our study used a qualitative observational methodology, our considerations should be seen as grounded hypotheses requiring further evaluation. Our methodology can serve as a guideline to assess how well other retrofitted tools support collaboration.

We believe that the changes we made to allow for multiple inputs are generally possible in other information visualization systems. Getting differentiable user IDs might be difficult on most multitouch technologies (a notable exception being the DiamondTouch display), so some retrofitting techniques for these technologies might have to work around the lack of user IDs. The most difficult changes, however, pertain to the effects of multiple people concurrently interacting with a system. In the next sections, we highlight the sets of common information visualization features that might be most affected by the introduction of synchronous inputs. These features thus might be the most important to consider when retrofitting information visualization tools.

Global Controls

When multiple people collaborate, global controls can raise problems; this is particularly important for information visualizations in which collaborators share views of a data set. For example, if a shared visualization is too large to fit on the screen, a single user's pan modifies the view for all the collaborators. A number of information visualization features are commonly implemented as global controls, and you might have to reconsider their design or use when introducing multiple concurrent inputs. Such features include filtering, navigation (for example, pan and zoom), transformations (for example, changes between different representations and changes of data encodings), or view changes (for example, projection and rotation). The amount and importance of these features will influence a retrofit's difficulty.

When retrofitting global changes, you can follow several strategies. The simplest solutions are to remove interface features that permit global changes, implement control policies, or simply let collaborators deal with potential conflicts themselves. We have had good experiences with a combination of the first and last options. Another solution requires developing novel interaction techniques in the system to replace those causing problems. For example, you could restrict zooming, panning, or filtering to influence only a local scope. For instance, with a node-link diagram, you could exploit the graph's topological information to replace panning by bringing a node's neighbors into the view. Similarly, to avoid panning as much as possible, you can use lens techniques to shrink areas of less interest and introduce new visual features.

Introducing these more elaborate interaction features might be particularly important for certain types of visualization systems. For example, in visualizations that cover the whole display, are space-filling, or include 3D views, global changes are often central to the tasks. For instance, a space-filling treemap relies on a zoom-and-filter operation, and a 3D visualization such as ConeTrees relies on global view changes for information to be understood and read in its entirety. In this case, the visualizations might need a more intensive redesign to allow exploration of multiple foci or coordinated views.

Undo and Reversible Actions

Being able to undo an action is important in almost any computer program and in collaborative information visualization systems. An undo feature must have an associated thread of actions. This makes undo for synchronous collaboration a difficult issue in retrofitting because several people could affect a single information item successively and because an undo's effects might be harder to coordinate.

You could simply keep a system's current undo capabilities (as we did in CoCoNutTrix), but then an undo will be global and will undo the last action, no matter which collaborator issued it. A user-specific undo might seem semantically more meaningful, but conflicts could arise when one collaborator tries to undo an action on an object that another collaborator subsequently modified. On the basis of our study, we believe that participants would prefer this personal undo; however, the system would have to incorporate more meaningful ways to deal with undo conflicts. In CoCoNutTrix, participants dealt with the problem by reversing their actions manually in the interface (by removing a node from a matrix or placing one back in). Adding local undo functionality to a system might be a viable alternative.

Windows and Dialogs

When multiple people synchronously interact with a visualization system, the connections between interface and information display items must be coordinated differently. So, you need to reimplement multiple windows, widgets, or dialogs to allow more than one concurrent interaction. This retrofit can be difficult but is crucial to allow for synchronous data exploration and manipulation. Its difficulty depends mostly on the underlying windowing toolkit. This applies particularly to multiple-coordinated-view systems or other information visualization tools requiring many dialogs and widgets. The difficulty in retrofitting dialog boxes lies in making the right connection from an interface item to the data that will be affected. For example, to which data item should you apply a color change if different people are simultaneously highlighting multiple items? One possibility is to track an item-plus-widget selection through input IDs. However, this would effectively hinder collaborators from teaming up and applying changes together—one highlighting the items, the other operating a dialog box.

In CoCoNutTrix, we made a more fundamental decision. We removed all dialogs from the interface and made only a subset of features available through direct mouse control (left click, right click, scroll wheel, and gestures). This effectively made all interaction local to the information items. We made a fundamental choice between two options: provide all possible features of the original tool, which might cause more interaction conflicts, or reduce the number of features and minimize them for the task at hand.

Anyone retrofitting an information visualization tool must make a similar choice. What task will the retrofitted tool be used for? Should the retrofitted tool be as powerful as the original? Or will a group use it to answer more-specific questions with specific tasks that are better solved by a group than an individual?

In CoCoNutTrix, participants solved the task well with our minimal set, so our choice was justified. However, they missed some of the original features, so in a second redesign phase we would reconsider this design choice. Also, for a different task or data set, we might have to do another retrofit and consider remapping interface gestures and inputs to other parameter changes, leading to more retrofitting effort.

Awareness Features

A retrofitted tool must support awareness of what the participants have looked at, what they have analyzed, and which data items they have made decisions about. In our case, this was facilitated mostly through the hybrid nature of the visualization. So, we hypothesize that information visualizations in which group members can give the data meaning by either transforming data items into different representations (as in our case) or annotating and marking them (for example, through spatial positioning or graphical markers) will not require much additional functionality.

Many other 2D network and graph visualizations can likely be easily retrofitted in regard to awareness features. Most of these tools already allow free spatial repositioning, which helps annotate or mark data by changing the data's position. Coupling this repositioning with user-specified visual clustering could help capture group insight and support group coordination and communication. Other systems should consider adding data annotation. This could simply include adding colorful labels, highlighting, or the ability to reposition or reorganize information items, if information is not already encoded this way.

Our results indicate that retrofitting existing information visualization software to include discount multi-input interaction is doable, given appropriate open source software. The resulting low-cost collaborative environments can be reasonably functional and well worth assembling. However, the retrofitted software's overall success depends on the interactive features of the original tool.

To refine our results and to be able to make further recommendations for low-cost retrofitting, we must study how other types of visualizations fare in a retrofitted scenario. We also must study how they are used in real-life situations where the analysis's outcome has a big impact on participants' everyday work. Our research and existing guidelines for collaborative information visualization can be a useful starting point. Future studies could refine this knowledge.

References

1. N. Henry, J.-D. Fekete, and M.J. McGuffin, "NodeTrix: A Hybrid Visualization of Social Networks," *IEEE Trans. Visualization and Computer Graphics*, vol. 13, no. 6, 2007, pp. 1302–1309.
2. P. Isenberg and S. Carpendale, "Interactive Tree Comparison for Co-located Collaborative Information Visualization," *IEEE Trans. Visualization and Computer Graphics*, vol. 12, no. 5, 2007, pp. 1232–1239.
3. E. Tse et al., "Avoiding Interference: How People Use Spatial Separation and Partitioning in SDG Workspaces," *Proc. 2004 ACM Conf. Computer Supported Cooperative Work* (CSCW 04), ACM Press, 2004, pp. 252–261.
4. A. Noack, "Energy-Based Clustering of Graphs with Nonuniform Degrees," *Proc. 13th Int'l Symp. Graph Drawing* (GD 05), Springer, 2005, pp. 309–320.
5. S. Wassermann and K. Faust, *Social Network Analysis*, Cambridge Univ. Press, 1994.
6. C. Gutwin and S. Greenberg, "The Mechanics of Collaboration: Developing Low Cost Usability Evaluation Methods for Shared Workspaces," *Proc. IEEE 9th Int'l Workshops Enabling Technologies: Infrastructure for Collaborative Enterprises* (WETICE 00), IEEE CS Press, 2000, pp. 98–103.
7. P. Baudisch, E. Cutrell, and G. Robertson, "High-Density Cursor: A Visualization Technique That Helps Users Keep Track of Fast-Moving Mouse Cursors," *Proc. Interact '03*, IOS Press, 2003, pp. 236–243.
8. M.R. Morris et al., "Beyond Social Protocols: Multi-user Coordination Policies for Co-located Groupware," *Proc. 2004 ACM Conf. Computer Supported Cooperative Work* (CSCW 04), ACM Press, 2004, pp. 262–265.

9. P. Isenberg, A. Tang, and S. Carpendale, “An Exploratory Study of Visual Information Analysis,” *Proc. 26th Ann. SIGCHI Conf. Human Factors in Computing Systems* (CHI 08), ACM Press, 2008, pp. 1217–1226.
10. A. Tang et al., “Collaborative Coupling over Tabletop Displays,” *Proc. SIGCHI Conf. Human Factors in Computing Systems* (CHI 06), ACM Press, 2006, pp. 1181–1190.
11. G. Stahl, *Group Cognition*, MIT Press, 2006.

Petra Isenberg *is a PhD candidate at the University of Calgary whose research is on collaborative information visualization in colocated environments. Her research interests include information visualization, visual analytics, and computer-supported cooperative work. Isenberg has a Dipl.-Ing. (MSc) in computational visualistics from the University of Magdeburg. Contact her at petra.isenberg@ucalgary.ca.*

Anastasia Bezerianos *is an assistant professor in the MAS (applied mathematics) Laboratory and the SAP Business Objects Academic Chair at École Centrale Paris. Her research interests include design and evaluation of interaction and visualization techniques for large-scale displays for single and multiple users. Bezerianos has a PhD in computer science from the University of Toronto. Contact her at anastasia.bezerianos@ecp.fr.*

Nathalie Henry *is a researcher in Microsoft Research’s VIBE (Visualization and Interaction for Business and Entertainment) group. Her interests lie in social networks, network visualization, information visualization, and human-computer interaction in general. Henry has a PhD in computer science from Université Paris-Sud and the University of Sydney, focusing on matrix-based representations for visualizing social networks. Contact her at nath@microsoft.com.*

Sheelagh Carpendale *holds the Canada Research Chair: Information Visualization and the NSERC/iCORE/SMART Industrial Research Chair: Interactive Technologies and directs the Innovations in Visualization group at the University of Calgary. Her research in information visualization, collaborative visualization, and large interactive displays draws on her dual background in computer science and the visual arts. Contact her at sheelagh@ucalgary.ca.*

Jean-Daniel Fekete *is a senior research scientist at INRIA, the French national institute for research in computer science and control. He leads the Aviz team, which focuses on visual analytics. His research topics include network visualization, evaluation of information visualization systems, toolkits for user interfaces, and information visualization. Fekete has a PhD in computer science from Université Paris-Sud. Contact him at jean-daniel.fekete@inria.fr.*

Silver Bullet Security Podcast
In-depth interviews with security gurus. Hosted by Gary McGraw.
www.computer.org/security/podcasts
Sponsored by SECURITY&PRIVACY
cigital