

# Collaborative Information Linking: Bridging Knowledge Gaps between Users by Linking across Applications

Manuela Waldner and Dieter Schmalstieg \*

Institute for Computer Graphics and Vision  
Graz University of Technology, Austria

## ABSTRACT

Information exploration processes are often conducted in teams of experts, family members, or colleagues. These teams have to retrieve information from different sources, verify it, and finally compare and discuss their findings to find consensus. Today, support for these collaborative processes is limited and users often end up sharing either a single PC with one user taking control or using separate workstations, where support for tight collaboration is limited. In this paper, we present collaborative information linking which visually connects information across private and shared application windows to bridge knowledge gaps between users. We present the technical infrastructure for multi-user interaction and personalized meta-visualizations on large multi-projector displays, and demonstrate how personalized visual links connect information across existing applications modified in a minimally invasive manner. An observational experiment showed that information linking helps individuals to deal with large display space and teams to switch between individual information retrieval and joint verification and discussion.

**Keywords:** Single-display groupware, collaborative information work, information linking

**Index Terms:** H.5.3 [Information Interfaces and Presentation (e.g., HCI)]: Group and Organization Interfaces—Computer-supported cooperative work

## 1 INTRODUCTION

Information extraction and analysis is often accomplished in a collaborative context. Examples range from families planning a holiday trip to medical experts discussing the treatments of their patients. What these examples have in common is the need to extract relevant information from multiple sources – from web sites and maps to patient records and biomedical visualizations – and to share these pieces of information with the collaborators to find consensus. In practice, these two steps are often separated. Each team member accomplishes the information extraction task individually, and contributions are discussed in dedicated meetings. This separation stems from multiple deficiencies in conventional computing systems: First, input is usually restricted to a single contributor. Therefore, specialized groupware applications are required to support co-located synchronous interaction. Second, information is rarely contained within one single application. Especially if experts from different fields collaborate, information is presented by different role-specific applications. If multi-user support is encapsulated by specialized groupware, synchronous collaboration incorporating multiple of such applications is not fluidly possible. Finally, screen space limitations impede the synchronous display of multiple information sources and representations.

\*e-mail: { waldner | schmalstieg } @icg.tugraz.at



Figure 1: Color-coded sets of visual links connect individual user selections across protected and shared application windows.

In this paper, we present the technical infrastructure as well as visualization and interaction techniques to visually coordinate information exploration across multiple applications on single-display setups. Existing single-user applications – ranging from specialized information visualization software to a conventional web browser – can be employed as private or shared information resources. Personalized meta-visualizations visually highlight elements associated with a user’s current activity across multiple applications (Figure 1). Collaboration-supportive interaction techniques and tools help users to keep interference from collaborators low and to manage their findings. In summary, our proposed system consists of the following components:

- A collaboration-transparent environment supports concurrent multi-pointer interaction and construction of large-scale projected displays in a standard desktop environment.
- Collaborative information linking affords visual coordination by user-specific visual links across applications [32]: Multiple single-user applications can contribute to a shared information analysis task by providing user-specific visual feedback and a shared visual context on public application windows.

We will report on design considerations for collaborative information analysis, the collaboration-transparent environment and visual linking infrastructure, and finally on an observational evaluation of collaborative information linking on a shared single-display workspace.

## 2 RELATED WORK

In the field of computer-supported cooperative work (CSCW), numerous groupware applications have been presented for collaborative information search, browsing, and analysis. Examples for

co-located, synchronous information work range from collaborative picture galleries [18], web search [1, 30], tree comparisons [13], text analysis [14], and graph visualization [12], to more general hierarchical data visualization in multiple coordinated views [28]. In these examples, collaboration support is guaranteed by using special-purpose collaboration-aware software tools. Mostly, these groupware applications are self-contained developments. In contrast, Isenberg *et al.* [12] showed that existing visualization software can be “retro-fitted” for multi-user interaction – for instance by rendering multiple color-coded mouse cursors within the application. Similarly, Forlines *et al.* [6] presented a wrapper for Google Earth to support multi-user input, viewport synchronization across multiple instances, and annotations, without changing the application’s core implementation. Yet, applications themselves are modified for collaboration-awareness and therefore cannot be combined with other groupware applications for a fluid information exploration process.

We draw our motivation from an early vision in the field of CSCW: Lauwers and Lantz [16] suggested to use existing single-user applications in collaboration-aware windowing systems instead of special-purpose groupware applications. For co-located collaboration in conventional windowing systems, Hutterer and Thomas [11] presented multi-pointer X (MPX), a modification of the X Window System for multi-pointer interaction. While MPX enables collaboration-transparency on the windowing system level, it does not provide any further collaboration features, such as personalized visual feedback, role-based access regulations, or interaction histories.

The system presented in this paper relies on MPX [11] for multi-pointer support on windowing system level. To accommodate for the missing collaboration features, we provide user-specific meta-visualizations illustrating how selected information in private application windows relates to information contained in shared windows. Visualizations are rendered as desktop overlay by the window manager. Single-user applications contributing to the information analysis process are extended by a minimally invasive approach. Applications neither need to be multi-pointer aware nor render user-specific information for themselves.

Most groupware systems for co-located, synchronous information work rely on single-display groupware (SDG) [24]: they provide a shared display and multi-input support. Typical display form factors of SDG systems are tabletop displays (*e.g.*, [18, 13, 14, 28, 30]) or projected wall displays (*e.g.*, [12]). A common problem of SDG systems is the interference of individual work. In particular, distractions caused by changes to the spatial display layout [5] and cursor movements by other users [33] have been reported. Tse *et al.* [29] report that users tend to work on separate areas of the workspace in conventional single-display workspaces and can coordinate their activities with little verbal explanation. Therefore, SDG systems often use a large, high-resolution display to allow multiple users to establish their personal territories and to visualize a lot of information simultaneously. For instance, WebSurface [30] and CoCoNutTrix [12] both use multiple projectors to create a large high-resolution display. Common problems of large, shared displays are the inability to track other users’ cursors [35], gesturing to distant display locations [23, 12], and locating items of interest on a display scattered with information [25]. Guiding the attention to a user’s cursor on a large display has been achieved by introducing a “spotlight” metaphor [15]. WinCuts [26] allow for visual comparison of private task-related window regions on a shared display. In WeSpace [35], relation between imagery of different laptop displays can be compared on a shared display by linking or overlaying the images. However, task-related WinCuts and landmarks for linking have to be selected manually. VisLink [3] and visual links across applications [32] visually connect related elements (*e.g.*, text or map locations) in distinct visualization

views and application windows, respectively, to facilitate location of relevant items on large displays and to guide attention to remote display locations. However, both techniques only provide one set of links at a time and therefore do not support collaborative work styles.

In this paper, we use visual links across applications [32] for visualizing how information in personal application windows relates to shared information. For this purpose, we extended the system to use multiple, user-specific sets of visual links across applications as collaboration-supporting visual assistance and introduce interaction techniques to protect application windows and to store findings. In contrast to previous collaboration techniques for large-scale displays, our technique visually connects related items between different users’ application windows without manual registration of associated landmarks.

### 3 DESIGN CRITERIA

In the following, we list a number of design considerations we established for our co-located collaborative information workspace. The basic motivation was to allow interaction across a rich selection of applications while supporting important coordination aspects of collaborative work (as summarized in the *mechanics of collaboration* [8]):

**Single-user application support.** A single special-purpose groupware application is often not sufficient to solve a complex collaborative task. Users often prefer the usage of existing, powerful, and well-known single-user applications to specific collaboration-aware applications [16, 7]. Looking at the process of sense-making [9], we see that the actual data analysis process using some form of visual encoding is only a sub-phase of the entire sense-making process. To support the overall process, a system for co-located collaboration should support information retrieval from multiple sources, analysis, as well as software for note-taking and structuring of findings [10]. If users with different background, preferences, and roles collaborate, they may rely on their specialized software and their role-specific visual encodings [34, 4], as well as dedicated shared applications or visualization views to establish common ground [2].

**Sufficient screen space.** A large, high-resolution workspace is required if a lot of information in personalized visual encodings is displayed concurrently [13]. A sufficiently large screen real estate is also crucial for users to establish their personal territories and to coordinate their activities [29]. With multi-user input support, users can work independently on private areas of the workspace and fluently switch to tight collaboration on public display spaces.

**Visual coordination.** To ease localization of relevant items scattered across multiple application windows and representations, visual cues can help to lead the user’s gaze to items relevant for her current task. Visual cues should be personalized so the user can easily identify which items are associated with her current activity and what other users are currently investigating. Linking items across representations might also help users when switching between loose and tight collaboration [13]. For a multi-application workspace this implies the necessity to synchronize operations and selections in different applications and to provide a consistent visual highlighting mechanism.

**Protecting workspaces.** Concurrent interaction on single-display groupware can lead to interferences [5, 33] – especially when working in a loosely coupled fashion [27] or using indirect input devices [20]. Besides social protocols, potential ways to decrease interference is to provide protected spaces (*e.g.*, dedicated private displays or screen areas) or explicit locking mechanisms for workspace items.

**Storing and sharing.** Users should be enabled to record their data for later review and merging of findings [2, 22, 10]. If multiple applications are employed, there should be a central tool which is able to capture “snapshots” of the environment or save extracted

findings in a generic form.

In the following sections, we introduce our technical infrastructure for synchronous, co-located collaborative information work based on these design considerations.

## 4 COLLABORATION-TRANSPARENT ENVIRONMENT

To create a collaboration-transparent environment, we build on recent advances of the X Window System for Linux. In particular, we use Multi-Pointer X (MPX) [11] and the Compiz window manager<sup>1</sup> as foundations for a single-machine implementation.

### 4.1 Multi-user interaction and identification

MPX [11] provides multiple independent mouse pointers and keyboard foci on windowing system level for the X Window System. Although MPX is already available in the most recent X Window System release, application support is still scarce. Most window managers do not reliably render or handle multiple pointers. We thus extended the Compiz window manager with basic multi-pointer support for our purposes. Compiz provides a plug-in infrastructure for extension of its functionality without changing the core implementation. We implemented a plug-in that queries the XInput2 event loop (a new X event loop for multi-pointer events) and renders each mouse pointer in distinct colors.

MPX allows multiple users to interact concurrently with distinct applications but it cannot resolve situations where multiple pointers attempt to access the same application concurrently, if the application itself is not multi-pointer aware. As long as conventional GUI-toolkits do not provide multi-pointer event notification, making existing applications multi-pointer aware requires substantial low-level modifications to the applications themselves. Our multi-pointer plug-in therefore provides an interface to query user identities based on (single-pointer) input events received in the application, such as key strokes or button presses (*c.f.*, Figure 2). The plug-in establishes an interaction history for received input events (*i.e.*, mouse motion, button press, and key press and corresponding time stamp). Applications connect to the window manager via a network interface (using *Ice*<sup>2</sup>), which matches event reports by applications with the plug-in’s interaction history and returns the ID of the pointing device. In our collaborative information linking infrastructure, user identity management is encapsulated by a central management application, so individual applications only report user interaction – irrespective of how many users are actually interacting (more details in 5.2).

### 4.2 Large displays

Large, high-resolution displays, as shown in Figure 1, can be constructed from multiple overlapping projections. We create seamless imagery with casually aligned projectors using a camera-assisted offline calibration step and geometric compensation of individual projector images in a Compiz window manager plug-in [31]. Warping and blending on the window manager level is transparent to all legacy applications and to the collaborative information linking infrastructure, and allows for real-time interaction.

## 5 INFRASTRUCTURE FOR COLLABORATIVE INFORMATION LINKING

Collaborative information linking provides personalized visual links across applications [32]: each user is provided with a distinct set of visual links connecting related elements across applications (see Figures 2 and 3). We will first briefly discuss the basic infrastructure of visual links across applications from [32], then introduce the extensions for multi-user links management, workspace protection, and interaction techniques and tools facilitating collaborative information analysis with conventional single-user applications.

### 5.1 Visual Links across Applications

Visual links are created upon user *selection*: depending on the application, the user marks a word, clicks on a chart, or enters a text in a search field. Selections are distributed as string to the individual applications, which report occurrences in displayed content. Mapping of the selection string to visual content is accomplished by the individual applications: they can map it to single words in a text, a geographic location on a map, a row in tabular data, or labeled elements in an image (*c.f.*, Figure 3).

The single-user implementation of visual links consists of three system components: A central management application – implemented as Tomcat Java application server – coordinates registered applications and user selections within these applications. Registered applications communicate via HTTP with the management application to report and receive user selections, as well as to report visual changes of content (*e.g.*, if the user has scrolled a page or zoomed into a map). When receiving a selection from the manager, applications need to map the selection string to occurrences in their content and report associated bounding rectangles. Finally, collected selection rectangles of the applications are sent to the window manager via Ice network interfaces, where rendering of the individual links is accomplished as an OpenGL desktop overlay.

From the window the selection has been reported (*i.e.*, the window the user is currently interacting with), one link for each window containing at least one occurrence of the selection (*target window*) emerges. At the intersection point of the link with a target window, links to the bounding rectangles of all window’s occurrences expand (see Figure 3). As visual links across applications were designed for large displays, window overlaps were not considered in the design.

### 5.2 Multi-User Links Management

Collaborative information linking extends this infrastructure by coordinating multiple concurrent selections by different users. Each user in the environment is uniquely associated with a pointing device and is assigned a distinct color, also used for cursor rendering. If an application reports a user selection, the management application first consults the window manager’s interaction history to

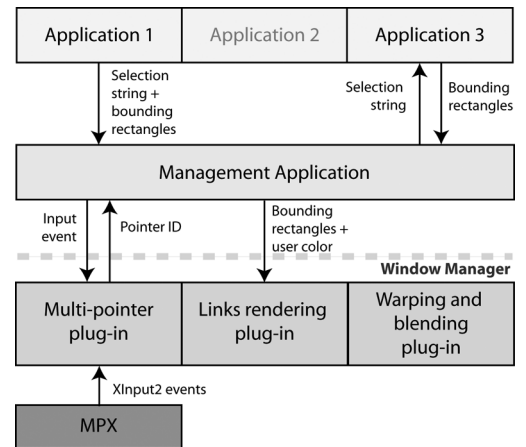


Figure 2: Collaborative information linking infrastructure: Application 1 reports a user selection and delivers the corresponding bounding rectangles of occurrences within the application. The management application consults the multi-pointer plug-in to determine the identity of the user triggering the selection and forwards the selection string to all applications accessible for this user (Application 3). After all applications have reported the bounding rectangles of selection occurrences, they are sent to the rendering plug-in of the window manager.

<sup>1</sup><http://www.compiz.org>

<sup>2</sup><http://www.zeroc.com/>

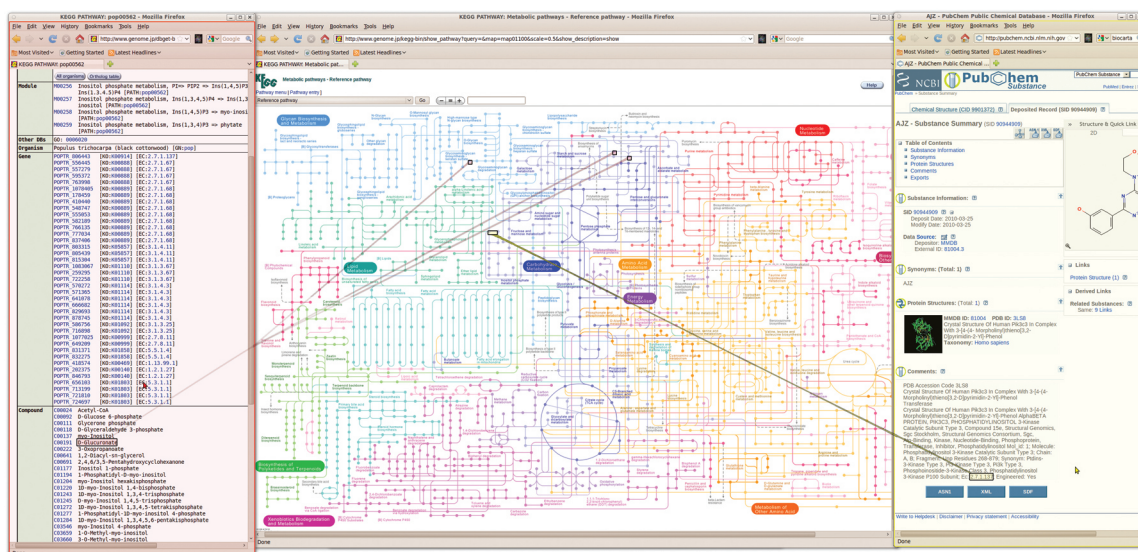


Figure 3: Screenshot of a shared workspace: color-coded sets of visual links connect text selected in two protected browser windows with matching occurrences in a shared overview chart.

determine the most recently active pointer within the application window's boundary. The manager then reports the selection to all applications accessible for the user associated with the determined pointing device. After all applications reported occurrences of the selection in their displayed content, visual links are routed and rendered in the user's associated color. Figure 2 illustrates these steps.

The management application stores the application windows a user is currently linking to. If an application reports a content change, the manager checks for users currently linking to the respective application and re-calculates their links.

Visual links are rendered in the user's associated color and emerge from the window the user selection has been reported (see Figure 3). Links are rendered in high opacity after the user has initiated a selection but fade to a low alpha value after a few seconds. The rectangular highlight regions around the reported selection occurrences remain at full opacity. This way, readability of content is improved. By moving the mouse over one of her selection highlights, the user can set back the links to their initial alpha value. For fading personal sets of links to a lower alpha level manually or switching them off entirely, users can employ simple keyboard shortcuts.

This infrastructure assures that applications communicating with the visual links infrastructure do not need to be collaboration-aware themselves. Instead, a minimally invasive approach by extending the application through an add-on or wrapping the application is sufficient. As depicted in Figure 2, applications need to report selection strings to the management application whenever a user selected an item, and map incoming selection strings to window locations (more details can be found in [32]). Multi-pointer support and user-specific visualizations are accomplished on the windowing system level – completely transparent for the applications and visually consistent across the entire desktop.

From a technical point of view, our system supports an arbitrary number of users. In practice, display space restrictions and the thereby resulting input conflicts, as well as visual clutter due to multiple sets of visual links, will be limiting factors.

For a more illustrative walk-through, we will explain the system's capabilities by a scenario: *Alice and Bob are conducting an open-ended, explorative investigation of the economic situation in different countries on a single-display setting, as shown in Fig-*

*ure 1. As information resource, they rely on content provided in the internet and start their exploration by a simple web search in two separated browser windows. In between their browsers, they place a map application. They register all three windows to the management application by pressing a button embedded in the browsers and map application, respectively. While Alice reads a newspaper article, she successively marks multiple countries mentioned in the text and presses a button to invoke visual links. By following the link to the map, she can quickly check the geographic location of these countries. In meanwhile, Bob opens some tabular data containing economic statistics. The country Alice is currently investigating is immediately highlighted in the table and linked to her article in red (her cursor color) as the page is loaded. So Bob starts investigating this country's statistics first.*

### 5.3 Protected and shared workspaces

In contrast to *input* conflicts in single-display groupware, where users compete for concurrent interaction on a single workspace item or access items in other users' personal territories, information linking does not provide the opportunity for social coordination, as visual links are created automatically. As a result, newly created sets of visual links may unintentionally intrude into other users' personal territories and interrupt their current work. Application windows therefore can be protected from other users' links by a simple locking mechanism. The user can protect any window in the workspace which is not already locked by another user by moving the mouse cursor to the desired window and pressing a shortcut. With the same procedure, a protected window is released. Protected windows are visualized by color-coded boundaries in the users' assigned colors (see lateral browser windows in Figure 3).

Window protection is coordinated in the window manager. Window (un)lock events are propagated to the visual links management application, which updates access restrictions between users and applications. In Figure 2, Application 2 has been protected by another user, so the manager does not forward the user's selection to this application.

*As Bob discovers a country showing some interesting economic statistics, he marks the country's name in the browser and selects the link button. Yellow visual links to the centered map, and the article Alice is currently reading, appear. As Alice first wants to*

*read some articles individually before discussing her findings with Bob, she locks her browser window to prevent his links from being rendered in her window.*

#### 5.4 Selection hijacking

Users can test whether others' selections occur in their private workspace by temporarily releasing window protection or by "hijacking" a specific collaborator's selection. To hijack a selection, the user moves the cursor over one of the collaborator's selection rectangles on a shared window. The user's links are then temporarily replaced with the collaborator's selection, while the previous highlight regions remain visible to explicitly show the relationship between the two selections. Hijacking is active as long as the user keeps the cursor within the selection region, with a minimum display time of one second.

*Bob asks Alice whether the country he has currently selected is mentioned in her article, which is protected. Alice moves the pointer to the country's selection rectangle on the shared map. Her current links fade out and are substituted by links emerging from Bob's selection rectangle on the map. As a link to her (protected) article appears, she reads the text passage mentioning the country out aloud. Then she moves her pointer back to her article window, her previous links are restored, and she continues reading.*

#### 5.5 One-shot linking

As applying the necessary extensions to existing applications for communication with the information linking infrastructure is not always possible or feasible, we provide a light-weight version of information linking for arbitrary, unmodified applications. One-shot linking temporarily connects a text item selected in an unregistered application with correspondences found in registered applications. The user selects the desired text and presses a shortcut to invoke linking, while hovering with the mouse above the selection. The management application then consults the operating system's selection buffer for the selection text and reports the selection string to all (accessible) registered applications. As we can only access the selection string and not the text location on the display using this method, links emerge from the current mouse pointer location instead of the actual bounding rectangle of the string occurrence. Also, we cannot perceive changes in window content of unregistered applications and therefore show these "one-shot links" only temporarily. After a few seconds, the previous set of links is reloaded.

*Alice recalls that she received an e-mail related to her current investigation a few weeks ago. She opens a conventional e-mail client and marks a country mentioned in the message text. Then, she presses the keyboard shortcut for one-shot linking and a link connecting her mouse cursor with the map application appears.*

#### 5.6 Selection storage and management

We provide a "bookmark list" as a central storage tool, where users can store their current selections for later investigation. In addition, users can use it as a global search tool by entering arbitrary selection strings and testing whether this selection is available in any registered application window. The bookmark list displays each user's current selection string together with a bookmark button. By pressing this button, a new bookmark appears as button labeled with the selected text. Linking the bookmarked selection is achieved by simply pressing this button. This way, users can quickly store findings as selection strings and test them later on new window content. By selecting a bookmark, links to the currently visible application content are created.

This is a different bookmarking concept than conventional web browsers provide, which store an application state or content rather than a user's search string. Our implemented selection bookmarking does not allow users to restore the applications' states at the time

of bookmarking their selection. To provide such an environment-wide bookmarking, registered applications additionally would have to provide context information with every user selection report – such as their current URL, document, or visualization view – and the possibility to restore these states upon bookmark selection. Alternatively, a system-wide time-centric information management approach (similar to [21]) could restore the entire system's previous (visual) state by storing screen-shots of the window contents at the time of bookmarking. However, restoring a previous system state can lead to interference for collaborators. Thus, privacy mechanisms – such as restricting reconstitution to unprotected application windows – would be required. In addition, the visual presentation of the windows' previous states has to be well-designed, in particular if the window layout has changed in meantime.

Alternatively to the bookmark list, users can manage their findings in their preferred text editors, spreadsheet applications, or presentation tools. Irrespective whether these tools have been integrated into the linking infrastructure, they can employ one-shot linking to re-establish their visual connections to registered application windows.

*After linking the country from her e-mail client, Alice opens the bookmark list to mark the country for later investigation. The country's name is automatically listed as potential bookmark, so she just presses the button next to the selection string to create the bookmark. Later, they both discuss Alice's findings and step through all of her bookmarked countries by subsequently pressing the labeled buttons in the bookmark list. By switching off window protection, they can check occurrences of the countries in all open application windows.*

### 6 EVALUATION

We conducted an observational experiment to initially assess the benefits and limitations of collaborative information linking. The main goals of the experiment were: 1) Observing how users coordinate their activities with conventional single-user applications and multi-pointer support on a shared display wall, 2) assessing how collaborative information linking helped (or impeded) users to coordinate their work, and 3) how individual findings were managed, communicated, and prepared. We decided to gather observational evidence as an initial evaluation step, as there is no clear baseline condition for a formal, comparative experiment. We believe evaluating collaborative information linking against a system without personalized visual coordination would not have been a fair comparison.

Eighteen participants (16 males, 2 females, aged 23 to 38) recruited from a local university worked in pairs on an information analysis task.

#### 6.1 Experimental setup

The experiment was conducted on a front-projected display driven by six XGA projectors, resulting in a resolution of 3072 x 1536 pixels. Projectors were operated by a commodity PC running Linux Ubuntu 9.10 with a dual-head graphics card and two graphics splitters<sup>3</sup>. Each participant had a mouse and keyboard pair to operate the display.

The task was to analyze migration from Africa to Europe by evaluating different sources of information. As information resources, we used content freely available on the internet, which was only slightly adapted to reduce the task complexity. Participants were presented a table showing numbers of migrants from African countries to European OECD countries in 2006<sup>4</sup>, a map, and several

<sup>3</sup><http://www.matrox.com/graphics/en/products/gxm/th2go/>

<sup>4</sup>simplified from <http://stats.oecd.org>



articles on the topic<sup>5</sup>. They were asked to answer three questions on African migration to Europe, where each of the questions were divided into focused and open-ended sub-questions. A few questions could be answered by evaluating solely the table (for instance: “From which African countries do most of the European migrants come from and where are they going?”), while others required support of the map and background information from the articles (for instance: “Which African countries are used as hub for migration to Europe? Where are the typical routes?”).

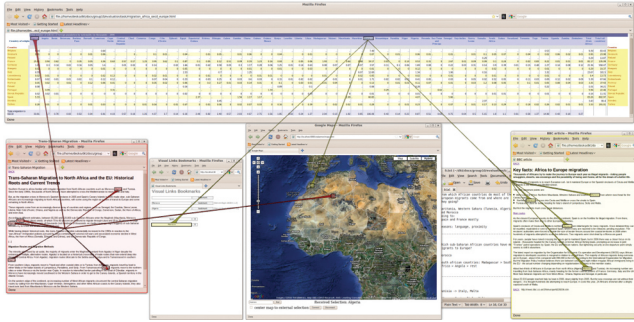


Figure 4: Screenshot of the shared display in the experiment.

We used the Firefox web browser<sup>6</sup>, which was extended through an add-on to communicate with the information linking infrastructure, and a maps mash-up application using the Google maps API<sup>7</sup> as application resources. In addition, a bookmark window and a simple, unmodified text editor (gedit) were available to organize findings. The table and the map were horizontally centered on the shared display. The articles were presented in two browser windows – one for each participant – and located on the respective closest display locations to the users. The two browser windows were protected from the other user’s links. The bookmark window was placed on the left, and the text editor on the right display half. Figure 4 shows the window arrangement.

Prior to the task, participants had an approximately 10 minutes warm-up period where they could get familiar with the system and the collaborative information linking technique. They were instructed how to make selections, switch off their links, and how to protect and release windows. Task time was limited to 30 minutes.

Groups could decide on their own how to organize their findings and were free whether to use visual links. We employed system logging, video-taped the session, handed out questionnaires, and conducted a semi-structured interview at the end of the experiment.

## 6.2 Results

We will report on observations during the experiment and video analysis of the sessions, as well as selected questionnaire results (7-point Likert scale), interview feedback, and logging numbers. For each pair, we obtained about thirty minutes of video material and two pages of field notes taken during the experiment. Based on the field notes and video material, we established coding categories for information linking, such as whether participants were linking for individual information retrieval or whether bookmarks were selected for individual information query or for collaborative discussion.

**Collaborative information linking.** The median number of selections triggered for collaborative information linking was 32 per

group. Information linking usage varied with the group’s configuration and work style (c.f., Figure 5): We observed that users working in a mixed-focus collaboration style (i.e., often switching between individual and tightly coupled work) used collaborative information linking most frequently.

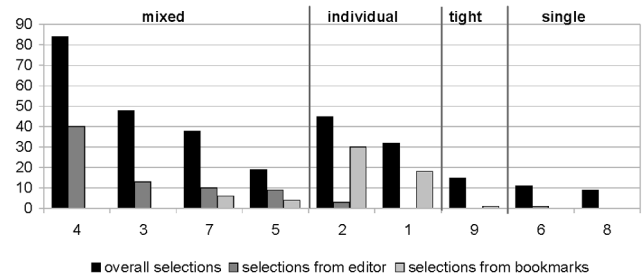


Figure 5: Recorded activities for each group sorted by work styles and frequency of information linking: mixed-focus collaboration, individual task-solving, tight collaboration, and a single interacting person (view coupling [27]) were observed work styles. The overall number of selections for information linking, and selections from text editor and bookmark list, respectively, are listed.

When working individually, information linking was mainly employed to locate occurrences of a certain country and to quickly look up its geographic location and associated migration numbers in the table. In the post-test questionnaire, the median rating for having visual links *across* application windows on a 7-point Likert scale was 6. Having links *within* application windows was rated 5. Linking was not actively used when reading articles or scanning the table individually. Links often remained faded during these activities, but were rarely completely disabled.

In seven groups, we could observe situations when users silently watched their partner’s activities while linking. Being able to see links by the partner had a median rating of 6. Users largely agreed that individual links were easy to identify (median score 5). The main distinguishing aspect was the unique color-coding (6), while the direction of the links played a minor role (4). One user explained he did not actively distinguish his links from his partner’s but rather followed his own links from the current window to the relevant target windows – color did not matter for him.

The median score for *I was often distracted by my partner’s actions* was 2.5. In the interview, users mentioned that interference and distraction was mainly caused by changes to the window layout caused by the other person, as well as concurrent interaction on shared application windows. Only one user mentioned that he was “a bit annoyed” by his partner’s links. The ability to protect windows from other users’ links was rated fairly low (median score of 4.5). It has to be noted that the browser windows containing the articles were protected when the experiment started. Only two users switched window protection off during the task, but no user locked one of the initially unprotected windows.

When working in a tightly coupled fashion, one participant usually contributed linking while the other set of links was either faded or switched off. In three groups, one user summarized findings by linking selective countries to map and table. Linking was also employed to clarify unknown items, e.g., the exact location of a country, especially if the country’s label was not visible on the map due to a coarse zoom level.

**Storing and sharing findings.** The bookmark list was actively used by four groups (c.f., Figure 5). However, usage of the bookmark list was usually only temporary – all but one group used the text editor to gather and structure all findings. Users complained that the bookmark list was too restrictive. While it seemed to be sufficient to gather first findings (as also indicated by the higher us-

<sup>5</sup>shortened from <http://news.bbc.co.uk/>, <http://www.migrationinformation.org/>, and <http://www.wikipedia.org/>

<sup>6</sup><http://www.mozilla.com/firefox/>

<sup>7</sup><http://code.google.com/apis/maps/>

age by groups working mainly individually), users stated that crucial collaboration features, like sorting or being able to add meta-information, were missing.

Both, bookmark list and text editor, were used to “replay” previous findings by stepping through the individual items to invoke linking. In four groups, the list in the text editor or bookmark list – created either by a single user or by both participants – was replayed by one participant to link to the map and the table. The group then discussed and verified the individual items and decided on the most suitable candidates collaboratively. Conversely, several group members consulted the list created in a collaborative session to identify relevant locations in their articles for further reading.

**Interaction with single-user applications.** Although we observed that most groups could resolve input conflicts on shared application windows by social protocols, the majority of users reported that occasional input conflicts were very annoying. Conflicts were observed when panning the map or selecting text by dragging the mouse while the other user was moving the cursor across the window or when attempting to drag concurrently. Apart from input conflicts, three groups complained about having to share a single bookmark or text editor window. One user mentioned that “*sharing windows is fine as long as you are mainly viewing. Like on the map or the table. But it’s exhausting if both have to interact*”. In practice, they agreed at some point to switch storage window ownership and moved it to the other side of the display or were alternately adding items on the original window location. This was perceived as awkward, as one user put it: “*windows on the left side of the display felt like [user B’s] area*”. Obviously, users silently agreed on window ownership based on the initial window layout. This is reflected in logging data, where 96% of registered interaction with the bookmark list (*i.e.*, bookmarking or selecting a bookmark for linking) was contributed by the user sitting on the left side of the display. Similarly, 75% of linking activities from the text editor were recorded by the user sitting on the right.

Some users reported problems when accessing distant shared content. For instance, one user argued that the numbers in the table were hard to read for him. However, he “*did not dare to pull the table down as I did not want to disturb [user B]*”. Indeed, three users reported that window re-stacking or moving by the other user caused distraction for them when working individually.

## 7 DISCUSSION AND FUTURE DIRECTIONS

The most extensive usage of collaborative information linking was observed for groups working in a mixed-focus collaboration style. Usually, these groups started by a joint task clarification and division step, followed by individual reading and gathering of information, and concluded with jointly discussing, clarifying, and verifying their findings. Similar collaboration patterns have been identified in previous research on collaborative information work (*e.g.*, [19, 17]).

As observed by previous work, facilitating multiple concurrent input devices and sufficient physical space leads to increased amount of individual work (*c.f.*, [29, 30]). Indeed, we saw a lot of individual work, primarily during information retrieval phase. Individual information retrieval was typically supported by visual links to find related information in shared application windows. Since previous research reported that cursor movements were perceived as distracting on a shared display [33], we were surprised to find that users hardly felt distracted by the other user’s visual links. However, our experiment was limited to a pair of users. A more thorough evaluation of visual coordination techniques and the resulting visual clutter for a larger number of users is subject to future work.

Users generally seemed to easily coordinate their activities during periods of independent work. Similar to previous observations on territoriality in single-display groupware (*e.g.*, [29, 27]), we saw

that users mainly interacted with windows located on their physically closer side of the display without explicitly negotiating personal areas. We did, however, observe occasional input conflicts when users attempted to access detailed information in a shared application window. This was the main cause of distraction mentioned by the users of the experiment. Although system-imposed access restriction is often considered as too restrictive [7, 5], well-designed automatic protocols have recently been demonstrated to be preferred over pure social policies by users [20]. We will investigate whether such protocols can help to mitigate distraction caused by unsynchronized input on single-user applications.

Collaborative information linking reduced the need to interact with shared window content – in some situations. It seems to be useful for mere overview applications, where users are rarely required to interact – such as the map application in our experiment, where the need to zoom into the map was often mitigated. However, it could not help users to read highlighted, but small, textual labels (as in the migration table) located at a far distance. Dragging shared windows closer was disliked by the users for the distraction this would have caused for their collaborator. In the future, we will investigate combinations of information linking and portal techniques (similar to WinCuts [26]) to provide easy duplication of relevant workspace areas for closer inspection with minimal distraction for the other users.

When transitioning from individual information retrieval to joint discussion, users often employed the information storage facilities: they “replayed” selected findings gathered in the bookmark list or text editor to their partner for focused discussion and verification. Mark and Kobsa [17] discovered a similar “retracing” step when users validated answers collaboratively. Conversely, some users replayed findings discussed in the team to get an entry point for new information sources when returning to individual information retrieval. Replaying selections also seemed to support users in deictic references. In contrast to previous observations reporting difficulties in gesturing on large displays (*e.g.*, [23, 12]), we did not observe any communication problems when pointing to a specific display location using visual links.

The tools we provided for information storage were described as too inflexible. Users had a strong desire to properly collect and structure their findings when working in a mixed-focus collaboration style. Based on the observations in our experiment, we found that a centralized information storage tool should provide similar features as powerful application-specific interaction histories [10], but should additionally enable the users to merge multiple personalized instances and to easily invoke personalized visual highlighting to diverse applications. As the information provided in the experiment was filtered, a simple selection storage mechanism was sufficient to revisit previous information occurrences. We expect that with a growing amount of information, mechanisms to restore previous application states (as discussed in 5.6) will be required.

We consider this observational experiment as starting point for further research in the field of collaborative information linking. In the future, long-term observations and more focused empirical evaluations will be required to fully assess the limitations and benefits of collaboration-supportive visualizations for information work.

## 8 CONCLUSIONS

We have presented collaborative information linking to provide visual multi-user coordination for information work in collaboration-transparent environments. Based on a system supporting construction of large-scale projector-based displays and multi-pointer interaction, we provide a shared visual context by linking user-specific selections to multiple application windows. In addition, we support the arrangement of protected and shared workspace areas and a bookmark list for storing selections. An early experiment indicates that information linking supports teams in operating a large, shared

display in several ways: Information linking helps individual users to locate information on a large-scale display with minimal distraction for their collaborators; it facilitates the transition between individual work and group discussions by replaying stored selections; and it eases the usage of single-user application windows on a shared workspace, as relevant items are highlighted without zooming content or moving windows into personal display space.

We will continue by addressing the future research directions discovered in our experiment, namely: providing a more powerful storage tool supporting users in structuring and restoring findings, combining linking and portal techniques, and conducting more focused experiments.

## ACKNOWLEDGEMENTS

The authors want to acknowledge Werner Puff for his contributions to the visual links framework. Thanks to Alexander Lex and Marc Streit, as well as all users of our experiment, for valuable discussions. This work was funded by the Austrian Research Promotion Agency (FFG) BRIDGE 822716 and 385567.

## REFERENCES

- [1] S. Amershi and M. R. Morris. Cosearch: a system for co-located collaborative web search. In *Proc. CHI 2008*, pages 1647–1656. ACM, 2008.
- [2] M. C. Chuah and S. F. Roth. Visualizing common ground. In *Proc. IV 2003*, pages 365–372. IEEE Computer Society, 2003.
- [3] C. Collins and S. Carpendale. Vislink: Revealing relationships amongst visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13:1192–1199, November 2007.
- [4] G. Convertino, C. H. Ganoe, W. A. Schafer, B. Yost, and J. M. Carroll. A multiple view approach to support common ground in distributed and synchronous geo-collaboration. In *Proc. CMV 2005*, pages 121–132. IEEE Computer Society, 2005.
- [5] C. A. Ellis, S. J. Gibbs, and G. Rein. Groupware: some issues and experiences. *Commun. ACM*, 34(1):39–58, 1991.
- [6] C. Forlines, A. Esenther, C. Shen, D. Wigdor, and K. Ryall. Multi-user, multi-display interaction with a single-user, single-display geospatial application. In *Proc. UIST 2006*, pages 273–276. ACM, 2006.
- [7] S. Greenberg. Sharing views and interactions with single-user applications. In *Proc. COCS 1990*, pages 227–237. ACM, 1990.
- [8] C. Gutwin and S. Greenberg. The mechanics of collaboration: Developing low cost usability evaluation methods for shared workspaces. In *Proc. WETICE 2000*, pages 98–103. IEEE Computer Society, 2000.
- [9] J. Heer and M. Agrawala. Design considerations for collaborative visual analytics. *Information Visualization*, 7(1):49–62, 2008.
- [10] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 14:1189–1196, November 2008.
- [11] P. Hutterer and B. H. Thomas. Groupware Support in the Windowing System. In *Proc. AUIC 2007*, pages 39–46, 2007.
- [12] P. Isenberg, A. Bezerianos, N. Henry, S. Carpendale, and J.-D. Fekete. Coconuttrix: Collaborative retrofitting for information visualization. *Computer Graphics and Applications: Special Issue on Collaborative Visualization*, 29(5):44–57, September/October 2009.
- [13] P. Isenberg and S. Carpendale. Interactive tree comparison for co-located collaborative information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1232–1239, November 2007.
- [14] P. Isenberg and D. Fisher. Collaborative brushing and linking for co-located visual analytics of document collections. In *Proc. EuroVis 2009*, volume 28, pages 1031–1038, June 2009.
- [15] A. Khan, J. Matejka, G. Fitzmaurice, and G. Kurtenbach. Spotlight: directing users’ attention on large displays. In *Proc. CHI 2005*, pages 791–798. ACM, 2005.
- [16] J. C. Lauwers and K. A. Lantz. Collaboration awareness in support of collaboration transparency: requirements for the next generation of shared window systems. In *Proc. CHI 1990*, pages 303–311. ACM, 1990.
- [17] G. Mark and A. Kobsa. The effects of collaboration and system transparency on cive usage: an empirical study and model. *Presence: Teleoper. Virtual Environ.*, 14(1):60–80, 2005.
- [18] M. R. Morris, A. Paepcke, and T. Winograd. Teamsearch: Comparing techniques for co-present collaborative search of digital media. In *Proc. TABLETOP 2006*, pages 97–104. IEEE Computer Society, 2006.
- [19] K. S. Park, A. Kapoor, and J. Leigh. Lessons learned from employing multiple perspectives in a collaborative virtual environment for visualizing scientific data. In *Proc. CVE 2000*, pages 73–82. ACM, 2000.
- [20] D. Pinelle, M. Barjawi, M. Nacenta, and R. Mandryk. An evaluation of coordination techniques for protecting objects and territories in tabletop groupware. In *Proc. CHI 2009*, pages 2129–2138. ACM, 2009.
- [21] J. Rekimoto. Time-machine computing: a time-centric approach for the information environment. In *Proc. UIST 1999*, pages 45–54. ACM, 1999.
- [22] A. Robinson. Collaborative synthesis of visual analytic results. In *Proc. VAST 2008*, pages 67–74, oct. 2008.
- [23] Y. Rogers and S. E. Lindley. Collaborating around vertical and horizontal large interactive displays: which way is best? *Interacting with Computers*, 16(6):1133–1152, 2004.
- [24] J. Stewart, B. B. Bederson, and A. Druin. Single display groupware: a model for co-present collaboration. In *Proc. CHI 1999*, pages 286–293. ACM, 1999.
- [25] K. Swaminathan and S. Sato. Interaction design for large displays. *Interactions*, 4(1):15–24, 1997.
- [26] D. S. Tan, B. Meyers, and M. Czerwinski. Wincuts: manipulating arbitrary window regions for more effective use of screen space. In *CHI ’04: CHI ’04 extended abstracts on Human factors in computing systems*, pages 1525–1528, New York, NY, USA, 2004. ACM.
- [27] A. Tang, M. Tory, B. Po, P. Neumann, and S. Carpendale. Collaborative coupling over tabletop displays. In *Proc. CHI 2006*, pages 1181–1190. ACM, 2006.
- [28] M. Tobiasz, P. Isenberg, and S. Carpendale. Lark: Coordinating co-located collaboration with information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1065–1072, 2009.
- [29] E. Tse, J. Histon, S. D. Scott, and S. Greenberg. Avoiding interference: how people use spatial separation and partitioning in sdg workspaces. In *Proc. CSCW 2004*, pages 252–261. ACM, 2004.
- [30] P. Tuddenham, I. Davies, and P. Robinson. Websurface: an interface for co-located collaborative information gathering. In *Proc. ITS 2009*, pages 181–188. ACM, 2009.
- [31] M. Waldner, C. Pirchheim, and D. Schmalstieg. Multi projector displays using a 3d compositing window manager. In *Proc. IPT/EDT 2008*, pages 1–4. ACM, 2008.
- [32] M. Waldner, W. Puff, A. Lex, M. Streit, and D. Schmalstieg. Visual links across applications. In *Proc. GI 2010*, pages 129–136. Canadian Information Processing Society, 2010.
- [33] J. R. Wallace, S. D. Scott, T. Stutz, T. Enns, and K. Inkpen. Investigating teamwork and taskwork in single- and multi-display groupware systems. *Personal Ubiquitous Comput.*, 13(8):569–581, 2009.
- [34] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *AVI ’00: Proceedings of the working conference on Advanced visual interfaces*, pages 110–119, New York, NY, USA, 2000. ACM.
- [35] D. Wigdor, H. Jiang, C. Forlines, M. Borkin, and C. Shen. Wespace: the design development and deployment of a walk-up and share multi-surface visual collaboration system. In *Proc. CHI 2009*, pages 1237–1246. ACM, 2009.