

Lark: Coordinating Co-located Collaboration with Information Visualization

Matthew Tobiasz, Petra Isenberg, and Sheelagh Carpendale

Abstract—Large multi-touch displays are expanding the possibilities of multiple-coordinated views by allowing multiple people to interact with data in concert or independently. We present Lark, a system that facilitates the coordination of interactions with information visualizations on shared digital workspaces. We focus on supporting this coordination according to four main criteria: scoped interaction, temporal flexibility, spatial flexibility, and changing collaboration styles. These are achieved by integrating a representation of the information visualization pipeline into the shared workspace, thus explicitly indicating coordination points on data, representation, presentation, and view levels. This integrated meta-visualization supports both the awareness of how views are linked and the freedom to work in concert or independently. Lark incorporates these four main criteria into a coherent visualization collaboration interaction environment by providing direct visual and algorithmic support for the coordination of data analysis actions over shared large displays.

Index Terms—Information visualization, Meta-visualization, Collaboration, Coordination, Co-located work, Workspace awareness.

1 INTRODUCTION

Real-world information continues to grow in size and complexity, creating ever increasing challenges for information workers. Both information visualization and collaborative team work have been suggested as important factors in addressing these information complexity challenges [17]. Information visualization has the potential to provide different ways of examining and exploring the data. Collaborative data analysis scenarios can combine the analytic power of multiple individuals, with the possibility of including varying types and levels of expertise, potentially leading to increased quality of solutions and discoveries. However, while considerable research is being conducted in both information visualization and computer supported cooperative work, comparatively less research examines the interplay between them. This is especially true for co-located collaborative scenarios. It is this problem we address: how to best support co-located collaboration among information workers who are making use of information visualizations in their analysis process.

Large multi-touch wall and tabletop displays offer new opportunities to support face-to-face collaboration, discussion, interpretation, and analysis around information displays. These displays have the potential to expand the possibilities of desktop-based data analysis environments as large displays allow multiple people to stand comfortably around a shared workspace with sufficient room for individual and group work. Multi-touch capabilities offer the opportunity for team members to manipulate both shared and individual instances of data representations concurrently, while large multi-touch displays offer new opportunities for information visualization collaboration support. Fully realizing this potential requires careful consideration of the design of information visualization workspaces, representations, and interaction techniques. One important aspect of designing for concurrent interaction with information visualizations is the coordination of collaborators' interactions with individual views of the data. Since this coordination can take place at several different stages of the visualization pipeline—data, representations, presentation, or view level [6]—making the stage at which coordination is taking place visually explicit may help provide awareness, which collaborators can use to flexibly coordinate their activities. We present Lark, an information visualization system for hierarchical data that supports this type of coordi-

nation by integrating a representation of the information visualization pipeline into the shared workspace, thus indicating coordination points for data, representation, presentation, and view levels. This visualized pipeline supports both the awareness of how views are linked and the freedom to work in concert or independently. Our goal is to provide a multiple view information visualization system in which the views are linked and coordinated by a meta-visualization offering a mechanism for integrating individual and team work, helping team members to switch between degrees of cohesion in their work, and allowing them to build on each others' findings.

2 RELATED WORK

Early work in Computer-Supported Cooperative Work (CSCW) has indicated that many group activities, such as brainstorming or planning, involve phases of *mixed-focus collaboration* in which group members transition from loosely coupled, parallel work to closely coupled, group work [5]. We also know that these transitions require the coordination of group members' activities. Mixed-focus collaboration has recently been shown to apply to our work scenario: synchronous co-located collaboration with information visualizations over shared displays [9, 16]. In Tang *et al.* [16] different types of lenses and filters were studied to understand different types of group cohesion. It was noted that a tradeoff is necessary between providing only a single or multiple independent instances of data views. With a single shared representation individuals' abilities to work independently may be compromised, yet using separate copied views may prevent many group collaborative dynamics from emerging.

Several collaborative information visualization and analysis systems have dealt with the problem of trading off individual and joint work with visual data representations and offered different solutions for coordinating among both types of work. Brennan *et al.* [1] proposed a distributed collaborative visual analytics framework that represents individual viewpoints as distinct from shared viewpoints. Individual views could be explicitly shared with others and were algorithmically merged to aid group analysis. In Keel's distributed analysis system [11], computational agents were used to identify when an individual had uncovered potential relationships between information items in his/her workspace; this insight was then automatically relayed to the larger group of collaborators. We share these goals of providing explicit logical and graphical support for sharing information and translating among different views. Yet, their execution is quite different as individual perspectives are emphasized and spontaneous interactions are not easily possible. We want to support the coordination of views and interactions so that team members can follow a momentary insight, glance at another's views, and transition quickly and effortlessly between different views of the data. This goal has been examined in two

• Matthew Tobiasz, Petra Isenberg, and Sheelagh Carpendale are with the University of Calgary. E-mail: {tobiasz|petra.isenberg|sheelagh}@ucalgary.ca.

Manuscript received 31 March 2009; accepted 27 July 2009; posted online 11 October 2009; mailed on 5 October 2009.

For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

previous systems for co-located data analysis [7, 8]. In a co-located collaborative tree comparison system [7], group members could create multiple view instances and interact with these as separate entities. Yet, coordination among views was limited to a tree comparison operation and did not allow for group members to easily coordinate data annotations or other data modifications. In Cambiera [8], group members were provided with coordinated visualizations of their search results through text document collections. Through “collaborative brushing and linking” individual search results and text document representations were linked and joint interactions and search overlap were explicitly visualized. Lark takes a different approach to the coordination of activities through explicit coordination points that use a representation of the information visualization pipeline.

Our implementation of coordination points relates to Heer and Agrawala’s idea of using the information visualization reference model as “entry points for collaborative activity” [6]. In earlier work, Wood *et al.* [20] used the idea of coordinating collaboration around the visualization pipeline in a distributed system. This work is conceptually close to ours but the implementation was limited to an architectural design whereas our approach focuses on making different coordination points in the pipeline visually accessible and understandable. We base our work on Chi and Riedl’s [3] operator interaction framework as a conceptual model for visualization operations. One of the main problems described in Chi and Riedl’s work is the *gulf of execution* or the difference between intended and possible interactions in a system, explaining the *view vs. value* property as a fundamental classification for system operations. This is a critical distinction if people are to be allowed to distinguish between view and value operations when multiple views and multiple group members’ interactions need to be coordinated. For example, interactions such as filters can apply to one or many visual representations (views) or propagate to the underlying data (value). A value operation would have a global scope and extend to all views attached to a specific data source, independent of which collaborator issued the operation. A view operation could have a more local scope, affecting only specific views of the data. The connections between our use of the visualization pipeline and the model proposed by Chi and Riedl are described in more detail in Section 4.

3 LARK: COLLABORATION CONCEPT

By incorporating visual information about how and at what level views are linked, Lark extends existing approaches to coordinated multiple views [18] of the same data in a single information workspace. We are motivated by the need for the coordination of interactions originating from multiple people synchronously working in a shared information visualization workspace. This work scenario shares the general requirements of coordinated multiple view systems: we want to map actions on objects in one visualization to actions in another [13]. However, our multi-user scenario requires that the coordinated multiple view concept be extended to better support concurrent synchronous interactions. The Lark coordination and collaboration concept focuses on four design criteria:

Changing Collaboration Styles: Since much evidence has been gathered about the importance of supporting team members in switching between tightly and loosely coupled work [5], a primary goal in Lark is to integrate this collaboration style support within a multiple coordinated view system. While collaboration styles are frequently discussed as parallel and joint, it has been shown that these are end points of a continuum which exhibits many variations in the degree of cohesion between team members [16]. For instance, collaborative work with information visualizations could transition through these phases:

- A:** Joint examination of one shared view, everyone interacting and discussing findings.
- B:** Joint examination of one shared view, only one person is interacting, the others are observing.
- C:** Parallel exploration using different views, people closely communicating about their findings.
- D:** Parallel exploration using different views, little communication occurs.

The need for coordination of view interactions with the data changes for the different cohesion styles inherent in Scenarios A–D. We want to support work styles as they gradually transition from parallel work (as in D) to joint examinations (as in B). During parallel work phases (such as D) interactions by each group member should stay separate so that actions such as filters or view changes remain local. During the transition towards C and B coordinating views between team members may benefit from the integration and relation of joint analysis results.

Scoped Interaction: To provide for these different types of coordination we want to support flexible definition of interaction scope. Scoped interaction is important in collaboration so that team workers can avoid interfering with each other’s tasks. Thus, the goal is to keep concurrent interaction individually scoped, allowing collaborators to choose how information will be linked and how changes to data, representation, or presentation will be propagated to different views. Through a visual representation of the information visualization pipeline in our system, collaborators can visually specify which other views their interactions will affect.

Temporal Flexibility: Based on recent evidence that temporal flexibility among information analysis tasks is common practice among team workers [9], our goal is to provide concurrent interaction within our system and to require no specific temporal flow of activities, allowing team members to follow their own unique analysis approaches.

Spatial Flexibility: Since changing collaboration cohesion in large workspaces is commonly accompanied by changing team member locations, our goal in Lark is to incorporate information views that can be individually placed, scaled, and organized throughout the workspace. This flexible approach to workspace organization can allow team members to establish their own work areas [14] and coordinate their actions [12].

4 REALIZING LARK

Lark is a collaborative information visualization environment in which a meta-visualization shows the links and relationships among multiple coordinated views. The views used to illustrate this discussion, present various tree layouts.

In order to facilitate the ease of collaboration coordination we chose to provide visual awareness of how team workers’ actions relate. We do so by integrating a meta-visualization [19] of the interconnections between all views that are currently displayed. This meta-visualization is based on the information visualization pipeline. Our design goals for this meta-visualization are:

- to create an integrated meta-visualization [19] within the visualization workspace,
- to make relationships between views explicit,
- to represent propagating interactions,
- to clarify the distinction between view or value operations,
- to keep visuals minimal, and
- to embed necessary interactions within the meta-visuals.

We start by describing the visual elements as manifested in Lark. Figure 1 shows an example of the systems interface: a single data set, a linked view meta-visualization with four views. Note that in this and the following figures, we organized the data views so that the meta-visualization is clearly visible for illustrative purposes.

4.1 Representing Lark’s Primary Components

4.1.1 View Representation

Each view in the system is contained within an individual view-pane, which is mobile and resizable, and can be placed anywhere in the workspace. This is an important feature for both individual and shared work. During individual work, team members may want to keep views they are working on in close proximity or within their personal territories [14]. Transitioning to more closely coupled work may require relocation and resizing of views so that they can more easily be seen by multiple collaborators. Being able to spatially re-orient interface items has also been shown to support ease of reading and collaborative communication practices [12].

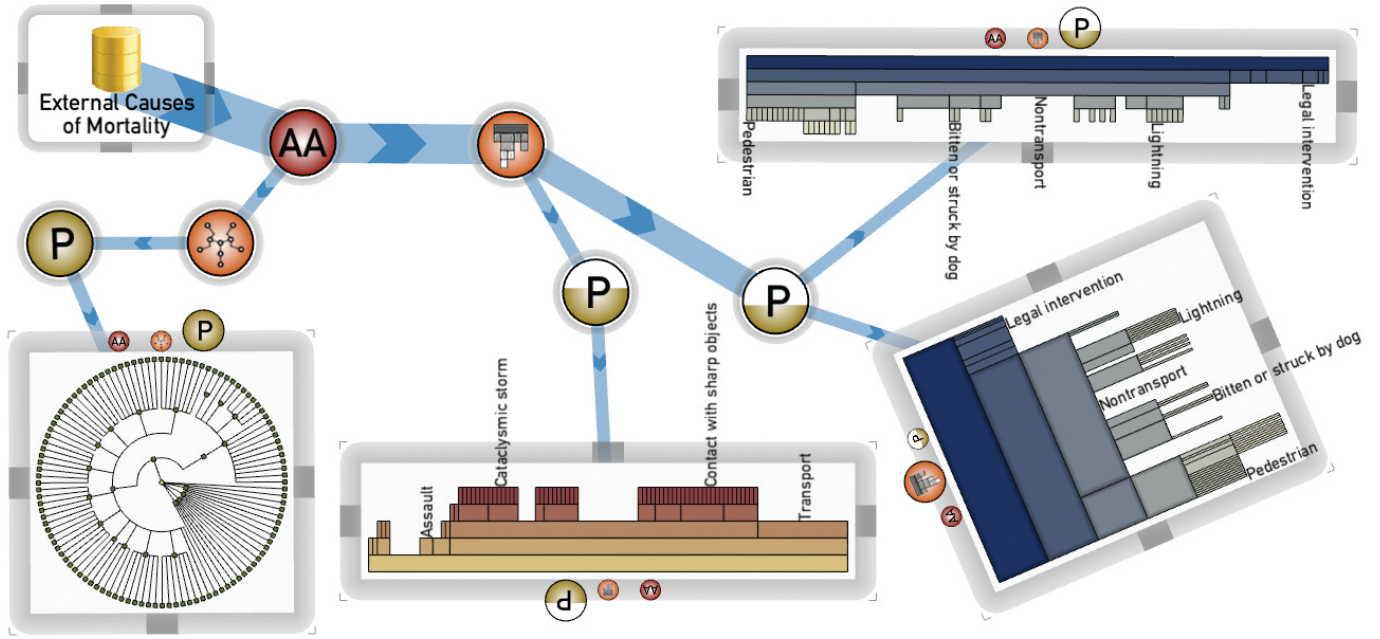


Fig. 1. Lark's collaborative visualization environment: single data set “External Causes of Mortality”, four views, plus the meta-visualization.

The visualizations contained in the view-panes, are the focus of the teamwork that is to be supported. Figure 2 shows a common data set drawn within individual view-panes and visualized with different layout algorithms. Data interactions are supported through gestures made directly on the data visualization. The view-panes are framed with a semi-transparent grey border, which provides resizing, translation, and RNT (integrated rotate and translate) [12] operations. The border's width is set to allow for direct-touch interaction. Resizing is initiated from any of the four corners, translation from the darker grey regions of the border, and RNT throughout the rest of the frame. The three icons at the top of the frame are radio buttons, with the current selection indicated as the larger of the three. Through interactions with these buttons, data interactions are scoped and linked.

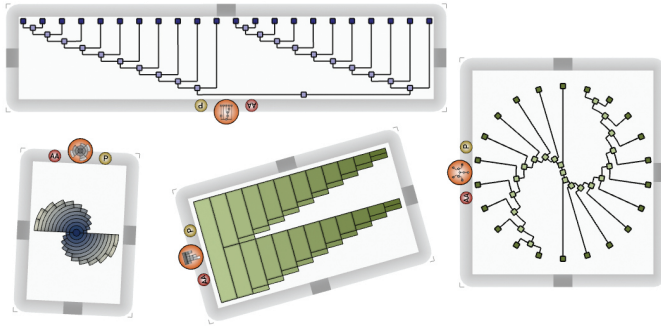


Fig. 2. Individual view-panes, visualizing the same data set with different layouts. View panes are mobile and resizable.

4.1.2 Pipeline Representation

Our version of the visualization pipeline is similar to the data state model from Chi and Riedl [3] where “each node represents a certain data state, and each edge represents an operator transforming the data from one state to the next.” In contrast to the pipeline used by Chi and Riedl, we make a specific distinction between the spatial layout and presentation stage, following the model introduced in [2]. Our conceptual pipeline and its visual representation is shown in Figure 3. It contains three specific coordination points for possible interaction: analytical abstraction, spatial layout, and presentation, which are discussed

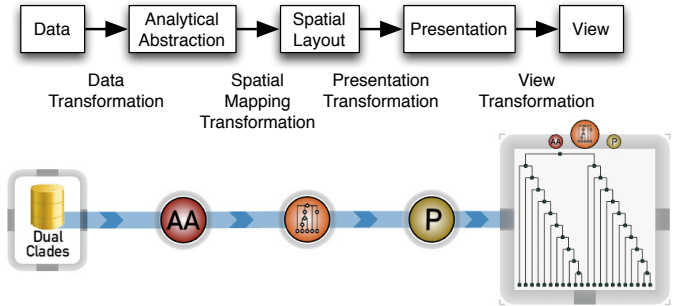


Fig. 3. The conceptual visualization pipeline (shown above) and its integrated meta-visualization (shown below) as seen in Lark.

in detail in Section 4.1.3. These are essentially *collaboration coordination points* (CCPs) and each is represented as a circular pipeline state icon, as seen in the lower half of Figure 3.

4.1.3 Collaboration Coordination Points Representation

The collaboration coordination points arise from the pipeline concept and are made visually explicit, allowing the development of a collaboration coordination tree that specifies which views are linked and at which level of the pipeline. In this meta-visualization, the number of views that are linked at any point in the collaboration coordination tree are indicated by the thickness of the tree edge. The collaboration coordination points themselves are indicated by a circle and labeled with an icon which declares the relation to the pipeline. The first pipeline phase contains *analytical abstractions*, labeled “AA” in the icon, which contains fundamental data operations such as removal of non-pertinent aspects of the data. The next phase takes the data in its “ready-examine-state” and creates a *spatial layout*. In Lark, a small layout icon indicates this phase. The third phase, *presentation*, labeled “P”, includes all temporary visual transformations, such as colour transformations. The final phase is the *view* itself, which contains the summation of the choices made in the previous phases, the ability to interact with the view, and the ability to indicate which pipeline phase the interaction should affect.

4.2 Visual Collaboration Coordination

Lark organizes and provides interaction with multiple visualizations of multiple data sets by illustrating the structure of the underlying visualization pipeline that was used to generate these visualizations. By making the visualization pipeline explicit, the relationships between individual views of a data set are emphasized. The visual structure of these relationships also provides awareness information to collaborators and shows how their interactions relate to one another.

4.2.1 View Generation

The system starts by showing available data sources, each labelled and in its own view-pane (Figure 4(a)). Creating a view of the data uses a *touch-drag-release* technique. *Touching* the data source and then *dragging* away generates a semi-transparent view with default parameters for factors such as colour scales and layout type. The newly created view gains opacity as it travels with the touch-point away from the data source (Figure 4(b)), to be *released* in its intended work location (Figure 4(c)). Note in Figure 4(b), this new view is linked to its data source via a visualization of the pipeline with coordination points for analytical abstraction (AA), visual representation (layout), and presentation (P). Note also that the view-pane's three interaction scoping buttons are initialized to the default (P), which keeps interaction local. This is a first branch of the underlying visualization pipeline which visualizes how views are coordinated (Figure 4(c)).

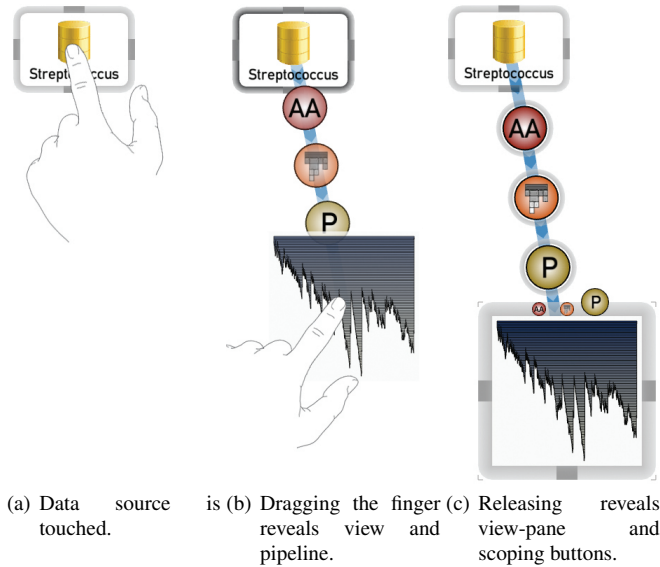


Fig. 4. Touch-drag-release creates a new view from the data source.

4.2.2 Pipeline Creation and Branching

The visual representation of the pipeline in Lark is shown in Figure 1. Here the pipeline is structured as a *tree*, where the root is the initial data state and leaves are individual view states. All leaves have an equal depth of four in this example.

Collaborators can dynamically create new visualization pipelines and branches from existing pipeline states through the *touch-drag-release* interaction technique. A new branch of the pipeline tree can be created from any collaboration coordination point. To create a new pipeline branch, the CCP icon is touched and the touch point is dragged to the intended view-pane location. As the distance between the CCP and touch point increases, the newly created branch proportionally gains opacity and is alpha blended into the workplace. Releasing the touch anywhere outside of the CCP icon boundary confirms the creation of the pipeline branch. A quick animated transition adds interface decorators to the components of the newly created pipeline and completes the fade-in from transparent to opaque. All CCP operators

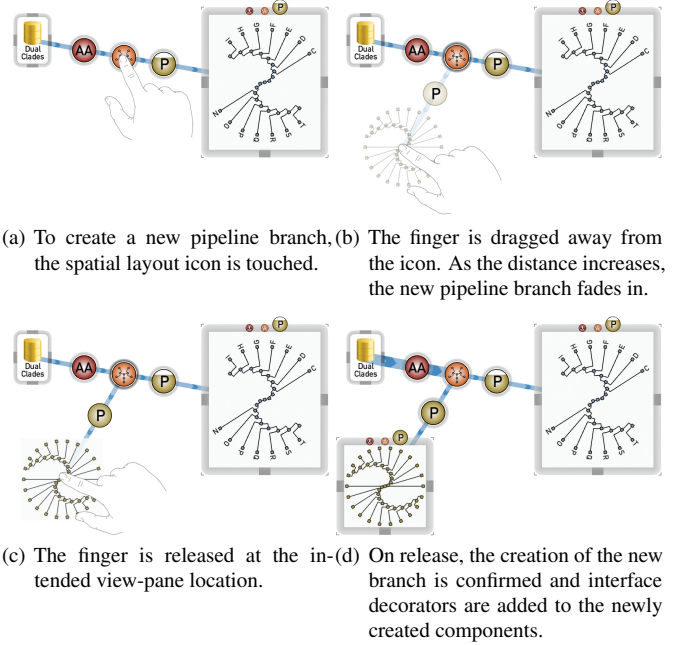


Fig. 5. Interaction technique for creating a new pipeline branch off an existing spatial layout collaboration coordination point.

of the newly created view and its pipeline link are initialized to default values. This pipeline creation interaction technique is illustrated in Figure 5. The operation is cancelled by relinquishing the touch anywhere within the bounds of the CCP icon. While the touch point is inside the CCP icon, the new pipeline is completely transparent, and so, cancellation occurs in a seamless and non-disruptive fashion.

4.3 Lark Interactions

Lark contains an integrated meta-visualization and a data visualization, and both are interactive elements. Interactions with the meta-visualization exclusively handle the creation of new views which contain data visualizations. Through use of the meta-visualization one can choose exactly where in the pipeline a particular view should be linked.

4.3.1 Setting Interaction Scope

All interactions with a data-visualization in Lark happen directly in the view. The interaction scope is explicitly set through three CCP icons bordering the top of a view-pane (Figure 6). Note that these use the same representation as the CCP icons from the pipeline, linking the data set and the view-pane. The icon for the currently selected scope is displayed at a larger size. View transformations are always independent; this means that operations such as changing rotation, translation, and size of a view do not affect other views. Hence, the view-pane does not include an icon for view transformations.

In Figure 6, P is currently selected indicating that all interactions will happen at the presentation level. If two view panes are linked at a presentation CCP, any operation performed on one of these views will receive the same operation and will change their display. Thus, via use of the view-pane CCP icons and locally scoped interactions in the view, one can explicitly coordinate how one's local interaction affects other views.

Setting the interaction scope can be quite crucial for how operations affect the final view. We will illustrate this with the example of filtering. Similar to Chi and Riedl's work [3], the distinction between view and value operations is particularly crucial for us. For instance, we make a distinction according to where filtering occurs in the pipeline: filtering as a data transformation at the analytical abstraction CCP removes the filtered parts of the data before it reaches the layout (what

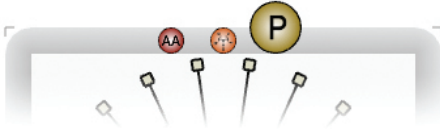


Fig. 6. The CCP icons bordering the top of a view-pane.

Chi and Riedl call *value-filtering*), whereas filtering at the presentation CCP is *view-filtering* [3]. Value filtering occurs before the spatial mapping stage, and hence, will influence the way the data is laid out in space. View filtering will still remove the selected aspect of the data but the basic spatial mapping is left unaffected with a gap where the data has been removed. This distinction is illustrated in Figure 7. While filtering in both cases uses the same gestures and has the same general result of removing data, where it is applied in the pipeline matters and leads to dramatically different effects.

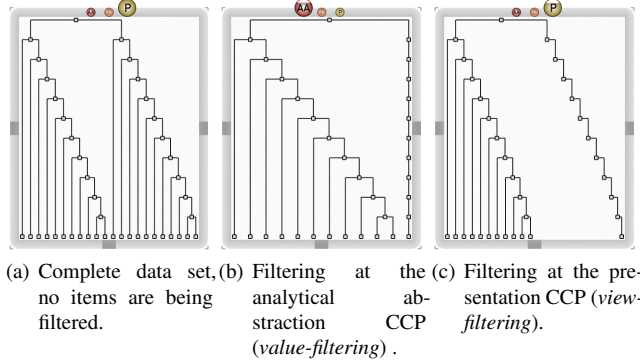


Fig. 7. The outcome of filtering operations applied to different points in Lark's pipeline.

We explored making different sets of gestures that would affect different parts of the pipeline. However, informal user feedback suggested keeping the gesture set simpler and using the CCP icons on the view-pane to make the scoping—either analytical abstraction (AA), layout, or presentation (P)—explicit.

4.3.2 Coordinated Interactions

Let us look at one concrete example. In Figure 1, we see four view-panes attached to the same data source. Each view shares a common analytical abstraction (AA) step, the three icicle plots on the right share the same layout step, and two of these share the same presentation step. By touching the P icon on the icicle plot on the top right we set this view to receive interactions affecting presentation. Any interaction on this view will automatically affect the other icicle plot that shares the same presentation CCP in the pipeline. Presentation operations include filtering, colourization, vertex/edge annotations, and selective labeling. Here we can see that both icicle plots connected at P share the same colour and labels and have been view-filtered to show the same subset of vertices. The three icicle plots share the same layout CCP. When the layout icon on the view-pane of one of these three views is selected, a gesture can be used to switch the layout of all three views. Available tree layouts in our system are currently: cladogram, radial cladogram, radial-space filling, and icicle plot. The radial cladogram is only connected to the three other views at AA and hence has independent layout and presentation parameters. Should one select the AA icon, a performed filtering operation would affect all four views, changing the spatial arrangement of the unfiltered subset.

We implemented several awareness features to help establish the coordination of interactions more explicitly. When a view-pane icon is touched and an interaction scope is selected, the branch connecting this view-pane with others that will be affected, is highlighted. Similarly, at the start of an interaction gesture, the branch highlights to

indicate that an interaction is about to occur and which other views it will affect. This helps to maintain common ground among collaborators by explicitly indicating the shared objects in the workspace as also discussed in [4]. In addition, the pipeline CCP icons are indicative of previous operations on the views connecting them. Both P and AA icons show the percentage of data that has been filtered (see Figure 8), while the layout icon shows which tree layout is currently selected (see Figure 9). This gives a high-level overview of what's happening at that particular CCP.



Fig. 8. Icon meta-visualization of the analytical abstraction CCP icon. The amount of colour fill indicates the percentage of filtered items at this state. This same encoding is used in the presentation CCP icon.

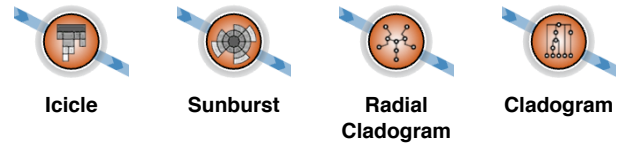


Fig. 9. Icon meta-visualization of the spatial layout CCP showing the four types of spatial layouts available in Lark.

4.4 Pipeline Cloning

The concept of pipeline cloning arose from observation of two different information task scenarios. One occurs during individual work and one occurs between collaborators. The independent scenario involves reaching a point in a data analysis sequence where one would like to compare a variation of one's current view to the current view. The collaboration scenario involves a team member being interested in a particular view and deciding to commence work starting from that view. Both of these scenarios lead to the need for an independent—not linked—but identical view. Creating new pipeline branches off a CCP icon is sometimes insufficient, as newly created CCPs are initialized to an unaltered default state. We therefore, included a cloning feature. Cloning a section of the pipeline makes a deep copy of each of the pipeline states, beginning with the selected state and moving down the pipeline to include the view state. This allows one to explore alternative configurations of the pipeline, based on previous modifications. Note that a new branch differs from a clone in that each of the pipeline states of a new branch are initialized to some default value, while a clone is a duplication of the selected pipeline.

A similar interaction technique to the one used in creating a new branch is employed in cloning. Instead of interacting with the CCP icon on the pipeline visualization (Figure 5), clones are created by interacting with the CCP icons (Figure 6) at the top of the view-pane. Pressing on a view-pane CCP icon and employing the same touch-drag-release to outside of the view-pane boundary creates a new clone of a section of the pipeline. This sequence is illustrated in Figure 10. The section of the pipeline to be cloned is chosen from whichever CCP icon it is operated from. As the touch is dragged further away from the view window, the opacity of the cloned branch proportionally increases. Confirmation of the operation is made by releasing the touch anywhere outside of the view-pane's boundary. Cancellation of the operation is done by releasing anywhere inside the view-pane.

5 IMPLEMENTATION OF LARK

Lark was designed for use on a large, multi-touch tabletop display. Our development system consisted of a touch-sensitive DVIT Board from SMART Technologies, capable of capturing two concurrent and

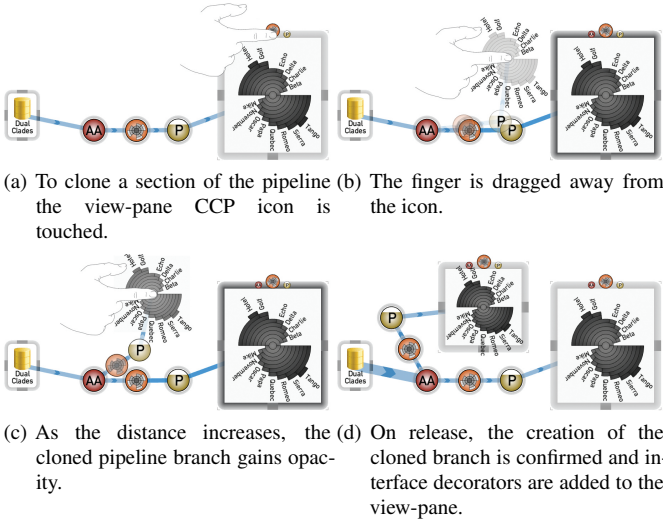


Fig. 10. Interaction technique for cloning an existing pipeline branch.

independent inputs that allow two people to work synchronously with objects in the workspace. The tabletop’s display area measures 1.5×1 meters, at a resolution of $2,800 \times 2,100$ pixels (≈ 5.9 megapixels), provided by four rear-mounted projectors in a 2×2 configuration. These projectors are driven by two NVIDIA GeForce 7900 graphics cards connected via SLI, running under Microsoft Windows XP on a single-core 3.0 GHz processor with 2 GB of RAM.

The visualization pipeline is the structural model for Lark, both within the visual workspace as well as throughout the underlying software design. Lark’s software architecture is comprised of a series of libraries—Elm and SnowMonkey—that were developed in order to support this *pipeline-centric* design, along with the Large Display Framework [10] and OpenGL. Figure 11 illustrates how these components are interconnected to form Lark’s system architecture stack.

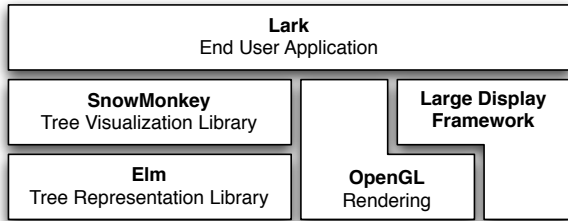


Fig. 11. Lark’s system architecture stack.

Elm is a lean, fast standalone library for representing hierarchically structured data; written in C++ it provides a generic API through the use of templates. Although existing software libraries for data hierarchies are available, such as the Boost Graph Library [15], we needed an uncompromising and consistent manifestation of the larger pipeline-centric design throughout the different tiers of the system’s API, beginning with the lowest tier. Elm implements the initial *Data* state of our visualization pipeline, as shown in Figure 12. Its only purpose is to provide a structure of the raw data which can be accessed from higher levels of the pipeline stack.

SnowMonkey builds on Elm, implementing the *data transformation* to *presentation* state section of the pipeline (Figure 12). SnowMonkey’s main purpose is to keep different pipeline states separate. It does so through two main features: the addition of *pipeline state meta-data* at each pipeline transformation, and *caching* to encapsulate data from previous pipeline states. These two features are necessary for the pipeline to be able to make an arbitrary number of branches at any subsequent pipeline state. For example, when data items (com-

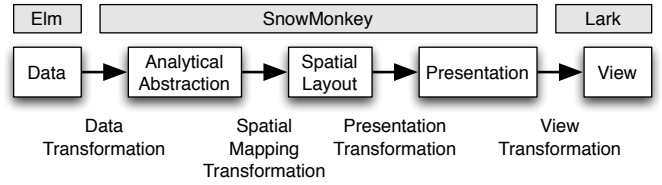


Fig. 12. The visualization pipeline and the specific software components which implement sections of the pipeline.

prised of vertices and edges) move from the *analytical abstraction* state to the *spatial layout* state, the *spatial mapping transformation* augments these items with geometrical meta-data, assigning each vertex and edge a position, size, and shape. Since we want to keep only one copy of the main dataset in memory, SnowMonkey creates a map of meta-data such as position, colour, visibility, etc. to the original data values. In addition, caching makes it possible for later pipeline states to modify definitions made by earlier states. For example, a cache is important to implement fish eye distortions: such distortions should occur at the presentation state. Without caching, the distortion would have to change the positions of items, a definition that was made at the spatial layout state. In this case, one presentation state would be modifying an earlier state, potentially affecting all other views connected at the layout state that do not want the distortion. With caching, fisheye distortion can remain local to the presentation state.

These two features—the addition of meta-data at each pipeline transformation and the caching of encapsulated data—are integral in realizing algorithmic support for *scoped interaction* throughout the pipeline. Furthermore, the creation of SnowMonkey was of necessity, as no known visualization libraries exhibit these features, crucial in the realization of our system.

At the top of the stack is Lark, the application which ties in SnowMonkey with the Large Display Framework and OpenGL to provide the analysis workspace environment, as shown in Figure 1. The Large Display Framework (LDF) is an interaction framework which uses an underlying buffer concept [10] to enable scalable, interactive response of interface components for large displays. The framework handles object management via a scene graph, and like Lark, uses OpenGL for rendering. Lark is also responsible for integrating the end products of SnowMonkey’s visualization pipeline into LDF and rendering this geometry with OpenGL. Lark leverages LDF to provide *spatial flexibility* for all visual components within the workspace, allowing components to be freely oriented and positioned throughout the workspace. Lark is separated from SnowMonkey so that it would be easy to use different windowing toolkits to implement the end user interface of the system, but keeping all the back end functionality provided by Elm and SnowMonkey. Lark is essentially responsible for interaction and visuals while Elm and SnowMonkey handle the coordination logic.

6 EXAMPLE USE-CASE SCENARIO

Lark is a visualization environment that lets people collaborate and flexibly build on each others’ explorations. To show the use of our system we describe a scenario that follows a fictitious team of three biologists, Chris, Ben, and Ana, working together to analyze clustered gene expression data from their latest experiment. A large number of genes and the complexity of their biological networks motivated the three to jointly analyze the data. We will explain how our design features are used as we follow how the analysts transition through several different phases of mixed-focus collaboration [5].

6.1 Parallel Work

The three analysts load their data on the shared workspace and begin their investigation. Since they do not know what to expect from their data, they first enter into an exploratory analysis phase. To broaden their data coverage, they initially decide to explore the data in parallel to look for interesting patterns. Chris, Ben, and Ana all create separate branches from the data source (Figure 13). Chris wants to get

an overview of the data and chooses to change presentation details by switching to an appropriate color scale, adding labels, and filtering out data that he deems uninteresting. Ben takes a different approach and wants to explore the largest branch first. He filters on the analytical abstraction level by touching the AA icon on the view. By performing a filter gesture he now filters parts of the data and a new layout is created that only shows the remaining information in a larger size. Ana creates two different views branched at AA, each with different types of layouts, and decides to do a comparative exploration to see if any representation will help her to see patterns better. This initial work partitioning outlines some of the collaboration features we want to address. First of all, **spatial flexibility** allows the three to pick their own parts of the workspace for exploration. On the three sides of the table, they can establish personal territories in which they place their views. By establishing a completely independent **interaction scope** at the data level, all three can work in parallel without affecting each others' views of the data. As to freedom of **collaboration style**, in this scenario, all three team members chose to work in parallel and this was simple to achieve. Lastly, Lark is also designed so that interactions are **temporally flexible**. As can be seen in this scenario, data analysis can be started from many different perspectives from identifying data items through use of colour and labeling to concentrating on a subsection by filtering to exploring different layout options. Also, no global interactions interfere with the sequence each worker chooses.

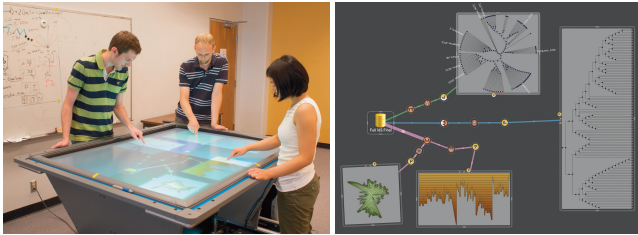


Fig. 13. Parallel work: Initially Chris, Ben, and Ana start by exploring the data with separate views

6.2 Parallel and Joint Work

While Chris has found some interesting patterns in the data that he wants to examine, Ana was not successful with her approach and walks over to Ben's work area and glances over his shoulder. She notices that Ben is closely examining an interesting branch. In order not to disturb Ben, Ana simply creates a new view from P and closely watches Ben's interactions. As the work continues Ben and Ana start a closer discussion of the data and decide to enlarge one of the views, examining data details together (see Figure 14). We see that Chris, Ben, and Ana are working in different collaboration styles. Chris is in loosely coupled collaboration to the other two, while Ben and Ana work more closely together. In this scenario, spatial flexibility lets Ben and Ana reposition and resize the view they are jointly analyzing. By specifically establishing a closely linked interaction scope Ana can follow Ben's interactions with the data for a while until they join in more closely coupled work. Chris' work remains separate.



Fig. 14. Parallel and joint work: Ben and Ana are discussing a view together while Chris still focuses on his own analysis.

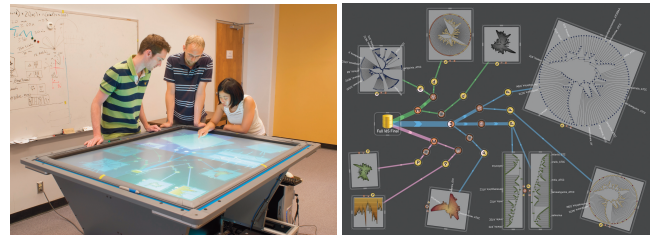


Fig. 15. Joint work: Chris, Ben, and Ana have moved to the same work region and enlarged one view to discuss.

6.3 Joint Work

As our three team members continue with their work, they create a number of different views of the data that they want to compare and organize in the workspace. When they want to save a certain state, they clone complete branches and continue their work from the new branch. At some point all three decide to come together to see what they have found and they closely discuss and negotiate their findings. Figure 15 shows Chris, Ben, and Ana having moved to one area of the workspace to discuss one view together. As the views that were created in the meantime are still in the workspace, they can see how each team member progressed through the analysis. If the discussion makes closer examination of the data necessary, each view can be further interacted upon.

7 DISCUSSION, LIMITATIONS, AND FUTURE WORK

Lark has arisen from an on-going collaboration with a group of biologists who are, by choice, gradually moving towards more collaborative data analysis. The Lark concept and its design criteria (Section 3), and the meta-visualization design goals (Section 4) arose out of iterative and collaborative discussions with the biologists and related literature. For example, consider the temporal flexibility aspect of the Lark concept. When the biologists worked with printed visualizations of their data, their data-analytic discussions would frequently cycle back to previously examined items. This type of temporal flexibility is supported in our system (1) in part by removing the sequencing of actions and (2) in part by allowing any work paths to be left available within the working environment. Keeping the work paths of this data exploration history, in evidence on the display, provides parallels to the freedoms of working with printed visualizations.

Also, the view and value filtering arose from direct separate requests from the biologists for the ability (1) to filter trees so that the layout does not move and only the presently unimportant part is temporarily removed to reduce clutter, and (2) to filter trees so that layouts are re-drawn after filtering in order to fully utilize space. Both types of filtering are available in Lark. The first is available through presentation (view) filtering and the second is available through the analytical abstraction (value) filtering.

Over the course of several iterative design sessions with our biologist collaborators, we have received encouraging positive feedback about Lark and its applicability to their everyday research problems. Even though the Lark environment is radically different from their accustomed software (for example Lark makes no use of menus), the biologists moved readily into data analysis. We did not put their current data in the system. They were almost immediately talking about their data to each other, suggesting view changes, making notes, and engaged in learning the system. They made active use of system features, generating new views of the data for comparison, and often expanding single views to maximize their display size. This did commonly cover the meta-visualization during a given discussion. However, uncovering the meta-visualization to create a new data view, did not seem to be an issue. In fact, it was during these times that they would make really positive comments about how the system let them interactively, within a given data exploration session, generate new views.

The biologists we have been working with actively want to make use of Lark and we are currently organizing a field deployment. How-

ever, our informal design sessions have also suggested that interaction with the pipeline meta-visualization does need to be learned and practiced, as the whole interaction paradigm is quite different from familiar software. We agree with Weaver [19] that further investigations into the possible advantages of meta-visualizations are warranted.

Also, while Lark was designed for tabletop interaction, we have frequently used it on a large dual monitor, high-resolution desktop setup. Interestingly, many of the tabletop features such as free rotation for collaboration communication [12], were also used in a similar communicative manner in the desktop setup. This would be an interesting direction for future work. While most of our sessions were with two or more biologists, when used by one person the Lark meta-visualization was also accessed, creating variant views and filtering.

The biologists made several requests for future work. The main request was the ability to merge branches, with a full suite of set operations for doing these merges: union, intersection, and complement. This would enable operations as: “show me all the common elements in these two views.” Another request was the possibility of moving operations on the pipeline, for instance, making it possible to request that a presentation filter become an analytical abstraction filter. In brief, some possible future directions include:

- provide support for more data types and operations,
- add the ability to merge,
- add methods for comparisons between branches,
- provide algorithmic support for automatic spatial organization of the meta-visualization and view-panes,
- explore different conflict resolution approaches, and
- add extended support for interaction history within the meta-visualization.

8 CONCLUSION

In this paper we have presented Lark a collaborative information visualization system that provides active support for collaboration by setting the data-visualization within a meta-visualization. This meta-visualization is based on the information visualization pipeline and uses the levels within the pipeline to create collaboration coordination points. Specifically, Lark supports:

Changing Collaboration Styles: Lark supports changing collaboration styles by allowing people to switch between tightly and loosely coupled work within a multiple coordinated view system.

Temporal Flexibility: Lark requires no specific temporal flow of activities, letting team members follow their own unique analysis approaches.

Spatial Flexibility: Lark’s views can be individually placed, scaled, and organized through out the workspace. This flexible approach to space usage supports mobility among team members, which is a common factor in changing collaboration cohesion during work.

Scoped Interaction: Lark’s interactions are scoped in the data-visualization gestures and commands all occur within individual view-panes. The scope of Lark’s meta-visualization interactions is explicitly set via the collaboration coordination point icons in the local view window. The choice of CCP setting specifies which view-panes will be affected.

Our research was motivated by the benefits and requirements of face-to-face collaboration with information visualizations. With the possibility of simultaneous, concurrent interaction with a visualization on a shared large display comes the need to support the coordination of joint data analysis efforts. Lark was designed to support a range of collaboration styles by providing collaborative coordination mechanisms in an extended multiple-coordinated view system. For the individual information worker this provides a new type of interacting with multiple views of the same data that are near to hand. For team work, the view structure can inform collaborators about not only what they have done but how their work relates to what their team members have done and the locally scoped interaction controls help coordinate collaboration. Our primary goal is to provide an effective visual analysis mechanism for both individual as well as team work, help team members to switch between both types of work, and build on each others’ findings.

ACKNOWLEDGMENTS

The authors wish to thank our funding agencies: Alberta Ingenuity, iCORE, CFI, NSERC, and SMART Technologies.

REFERENCES

- [1] S. E. Brennan, K. Mueller, G. Zelinsky, I. Ramakrishnan, D. S. Warren, and A. Kaufman. Toward a multi-analyst, collaborative framework for visual analytics. In *Proc. of the IEEE Symp. on Visual Analytics Science and Technology (VAST)*, pages 129–136. IEEE Comp. Soc., 2006.
- [2] M. S. T. Carpendale. *A Framework for Elastic Presentation Space*. PhD Thesis, Simon Fraser University, 1999.
- [3] E. H. Chi and J. Riedl. An operator interaction framework for visualization systems. In *Proc. of the IEEE Symp. on Information Visualization (InfoVis)*, pages 63–70. IEEE Comp. Soc., 1998.
- [4] M. C. Chuah and S. F. Roth. Visualizing common ground. In *Proc. of the Intl. Conference on Information Visualization (IV)*, pages 365–372. IEEE Comp. Soc., 2003.
- [5] C. Gutwin and S. Greenberg. Design for individuals, design for groups: tradeoffs between power and workspace awareness. In *Proc. of the ACM Conference on Computer Supported Cooperative Work (CSCW)*, pages 207–216. ACM, 1998.
- [6] J. Heer and M. Agrawala. Design considerations for collaborative visual analytics. *Information Visualization*, 7(1):49–62, 2008.
- [7] 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, 2007.
- [8] P. Isenberg and D. Fisher. Collaborative brushing and linking for co-located collaborative visual analytics of document collections. *Computer Graphics Forum*, 28(3):1031–1038, 2009.
- [9] P. Isenberg, A. Tang, and S. Carpendale. An exploratory study of visual information analysis. In *Proc. of the Conference on Human Factors in Computing Systems (CHI)*, pages 1217–1226. ACM, 2008.
- [10] T. Isenberg, A. Miede, and S. Carpendale. A buffer framework for supporting responsive interaction in information visualization interfaces. In *Proc. of the Intl. Conference on Creating, Connecting and Collaborating through Computing (C5)*, pages 262–269. IEEE Comp. Soc., 2006.
- [11] P. E. Keel. Collaborative visual analytics: Inferring from the spatial organization and collaborative use of information. In *Proc. of the IEEE Symp. on Visual Analytics Science and Technology (VAST)*, pages 137–144. IEEE Comp. Soc., 2006.
- [12] R. Kruger, M. S. T. Carpendale, S. D. Scott, and A. Tang. Fluid integration of rotation and translation. In *Proc. of the Conference on Human Factors in Computing Systems (CHI)*, pages 601–610. ACM, 2005.
- [13] C. North and B. Shneiderman. Snap-together visualization: A user interface for coordinating visualizations via relational schemata. In *Proc. of the Working Conference on Advanced Visual Interfaces (AVI)*, pages 128–135. ACM, 2000.
- [14] S. D. Scott, M. S. T. Carpendale, and K. M. Inkpen. Territoriality in collaborative tabletop workspaces. In *Proc. of the ACM Conference on Computer Supported Cooperative Work (CSCW)*, pages 294–303. ACM, 2004.
- [15] J. G. Siek, L. Lee, and A. Lumsdaine. *The boost graph library: user guide and reference manual*. Addison-Wesley, 2002.
- [16] A. Tang, M. Tory, B. Po, P. Neumann, and S. Carpendale. Collaborative coupling over tabletop displays. In *Proc. of the Conference on Human Factors in Computing Systems (CHI)*, pages 1181–1290. ACM, 2006.
- [17] J. J. Thomas and K. A. Cook, editors. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Center, 2005.
- [18] C. Weaver. Building highly-coordinated visualizations in improvise. In *Proc. of the IEEE Symp. on Information Visualization (InfoVis)*, pages 159–166. IEEE Comp. Soc., 2004.
- [19] C. Weaver. Visualizing coordination in situ. In *Proc. of the IEEE Symp. on Information Visualization (InfoVis)*, pages 165–172. IEEE Comp. Soc., 2005.
- [20] J. Wood, H. Wright, and K. Brodlie. CSCV—computer supported collaborative visualization. In *Proc. of BCS Displays Group Intl. Conference on Modeling*. Academic Press, 1995.