

A Visual Query Language for Correlation Discovery and Management

Geoffrey M. Draper*
School of Computing
University of Utah

Yarden Livnat†
Scientific Computing and
Imaging Institute
University of Utah

Richard F. Riesenfeld‡
School of Computing
University of Utah

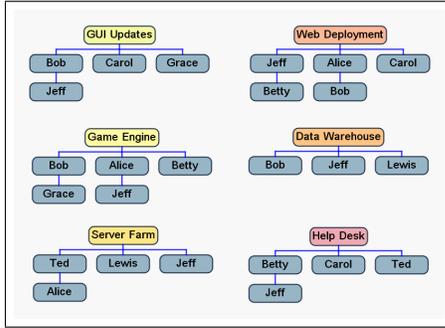


Figure 1: Typical assignment chart.

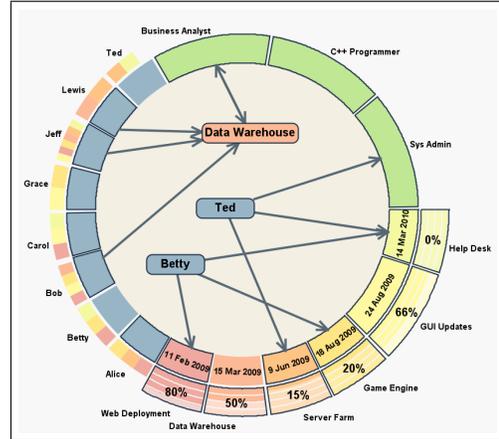


Figure 2: Our proposed visualization.

ABSTRACT

Discerning relationships among entities in a large iconic chart can be a challenging task. As an alternative to traditional chart layout schemes, we introduce an interactive radial query language for simplifying the task of searching for and identifying subtle correlations among data. Our approach allows the user to select which entities to show relationships for, thereby decreasing the cognitive overload associated with static charts. We likewise present a compact visual representation for comparing the differences between two versions of a chart. We have also implemented an intuitive gesture-based interface for creating and removing links between entities, thus enabling users to edit data, not just view it. Our preliminary user trial suggests that users can discover correlations significantly more quickly and accurately with our method as compared to traditional chart representations.

Keywords: Project management, resource allocation, situational awareness, gestural interfaces, visual diffs, social networks.

Index Terms: K.6.1 [Management of Computing and Information Systems]: Project and People Management—Staffing; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

1 INTRODUCTION

Locating specific information in a large iconic chart can be a challenging task. Traditional charts are both *static* and *accessible* [48], meaning their complete state is visible to the viewer at all times. Consequently, as a chart grows in size, certain tasks become increasingly complex. For one thing, it is difficult to locate specific entities within the chart. Also, it is quite time-consuming to determine relationships among entities in the chart, other than the relationship for which the chart was constructed.

*e-mail: draperg@cs.utah.edu

†e-mail: yarden@sci.utah.edu

‡e-mail: rfr@cs.utah.edu

Consider, for example, the “assignment chart” shown in Figure 1. This chart depicts the current projects of a hypothetical small business, along with the names of the employees currently assigned to each project. While useful in some situations, charts such as this are not well suited for answering queries like, *Which projects is a particular individual working on?* Or, *Given two employees, which projects do they work on together?* Such queries are likely to come up whenever a manager needs to evaluate the state of a project or to reallocate resources among projects. Although this information is present on the chart, it is not displayed in an easily accessible format. The situation is further complicated if the manager wants to view a dimension of the data that is not visible on the chart; for example, a list of skill sets required to complete a certain project.

To address these issues, we propose a computer-based visual query language for data sets typically rendered as static charts (Figure 2). Icons representing individual entities are placed around the circumference of a ring, and relationships among multiple entities are shown when an icon is dragged into the ring’s interior. In this manner, correlations in the data are displayed on demand, without the need to manually search for them within a chart.

In contrast to traditional chart rendering techniques, our method only shows links for entities selected by the user, helping to reduce visual clutter. However, for each selected entity, every link between that entity and any others are displayed, thus allowing the user to discover relationships that are not readily apparent from static diagrams. Our visual language also supports the ability to add and remove links between entities, as well as visually compare two versions of a data set.

The concepts presented in this paper have applicability to a wide range of problem domains. For the purposes of our initial prototype, however, we have chosen to focus on the specific problem of helping project managers allocate their resources effectively. We discuss other possible application domains in Section 5.

Although project managers have many variables to coordinate, we have chosen to focus on three: the human *resources* on his or her team, the *projects* or tasks assigned to the team, and the *skill*

sets or proficiencies possessed by the team members, and required by the projects.

The remainder of the paper is organized as follows. We first summarize related work in the areas of project management visualization, visual query languages, and radial visualization. We then present a detailed overview of our visual query language, including its syntax for both standard queries and difference queries. The next section mentions some other application domains to which our visualization could be adapted, suggesting the generality of our technique. We next describe our prototype software implementation, with an emphasis on its editing features and level of detail controls. Finally, we report on the findings of our informal user study, and conclude by proposing directions for future work.

2 RELATED WORK

Our work arises from the needs of a number of fields, namely, project management, visual query systems, and radial visualization. We briefly review some of the related work in each of these areas.

2.1 Project management

During the late 1980s and extending into the 1990s, there was a surge of interest in software-based tools for project management. This led to the development of a number of systems [9, 36, 40, 44], with Microsoft Project eventually becoming a *de facto* standard [37]. As Microsoft Project aims to be a general purpose system, other project management tools also exist as specialized solutions to niche problems [19, 25, 39].

There has also been a great deal of research incorporating aspects of information visualization into project management. The most well-known visualizations for project management are the PERT chart and the Gantt chart. One of the earliest examples of computer-generated visualizations for project management is the *progress diagram* [33] which displays some of the same information as a Gantt chart, but resembles a tree structure in appearance.

Szekely et al. [57] use the established visualization technique of *star fields* [1] to assist in planning military missions. Wang and Lillehagen [59] employ a variety of known diagram types (*swim lanes*, *trees*, and others) in their system, but do not introduce any new visualizations per se. Stamey and Honeycutt [53] apply a *quad chart* to the domain of project management. Kontio et al. [30] use a form of directed acyclic graph (DAG) to visualize risks associated with projects. The *SSD graph* [32] is an attempt to visualize both the structure and the state of a project.

Halin and Hanser [18] present a novel visualization system to show relationships between resources and tasks. Of all the systems reviewed herein, this one perhaps is most closely related to our work. However, we take a somewhat different approach, and focus, in particular, on interactivity and ease of use.

2.2 Visual query systems

Making data analysis easier has been a subject of research for several decades. Cammarano et al. [3] make the observation that most user interfaces for data analysis take either one of two approaches. In the first case, the user interface simplifies the formulation of the query; in the second case, the user interface helps to visualize the results.

Catarci et al. [5] provide a comprehensive, if slightly dated, survey of the literature in the first category. Of particular interest is the problem of querying semistructured or unstructured data, such as websites or emails. Sinha and Karger [51] propose a system for aiding in navigation of semistructured data sets by suggesting navigation hints to the user. Trigoni [58] lets the user refine a query over time by gradually disclosing the underlying data one part at a time. Goldman and Widom [17] propose a method for exploiting similarities among pages in the same website to perform more effective queries. Polyviou et al. [46] describe an interface for performing

database queries based on the ubiquitous filesystem browser interface. The VisTrails system [49] makes use of provenance data to maintain a history of past queries for creating visualizations.

In the second category, there are many systems which offer a direct-manipulation interface for browsing the results of a query. Furnas and Rauch [15] as well as Stonebraker [56] present canvas-based visualizations that support zooming and panning. Xmdv-Tool [60] facilitates the creation of standard statistical graphs to display query results. Visage [47] by Roth et al. is a highly interactive direct manipulation system that uses a variety of graphing techniques to communicate results to the user. In addition, Keim [27] and de Oliveira and Levkowitz [8] present surveys of many database visualization techniques.

In contrast, the Polaris system [55] incorporates both a novel query interface mechanism and an integrated visualization. Based on the well-known metaphor of a pivot table in a spreadsheet, Polaris allows users to view correlations in the data with respect to any attribute in the data set. Our work is similar in that we also provide a streamlined interface for performing queries and for *in situ* visualization of the query results.

2.3 Radial visualization

Our design metaphor is based on the radial layout, a charting technique whose tradition spans back hundreds of years, and is an increasingly popular metaphor in contemporary information visualization science. Pre-digital examples of radial information layouts include William Playfair's 1801 invention of the pie chart [45, 52], and Florence Nightingale's rose diagrams [41] for communicating sanitary conditions in British military hospitals during the Crimean War. In the mid-twentieth century, Northway used radial diagrams to track the social behaviors of gradeschool children [43].

Much of the recent work in radial user interfaces traces its lineage to research in graph layout algorithms for computer graphics [21]. These algorithms, in turn, have inspired techniques for visualizing multivariate data. Many such designs involve positioning data points as nodes on the spokes on a wheel [20, 22]. In these visualizations, the center point of the canvas holds some semantic meaning, and the distance of each node from the center shows a relationship relative to it. A recent example of this is the DataRose from [11].

In contrast, a second variety of radial visualization (called *radial space filling* or RSF [54]), the data points are typically arranged in compact concentric rings [29] or in a spiral [2, 4], and rendered so as to form a circle. Each ring contains a different category of data. Examples include polar treemaps [24], fan charts [10], and Radial Traffic Analyzer [28].

Another general category of radial visualization arranges the data points around the circumference of a ring, while reserving the interior of the ring for other data. Correlations among data points are often rendered as lines between nodes on the circumference and nodes in the interior. Examples include *Daisy* [7], *NetMap* [16], and *VisAlert* [34].

3 VISUALIZATION CONCEPTS

In our visualization metaphor, all available entities (projects, resources, and their related skill sets) are arranged in a ring. This egalitarian approach places all variables on equal footing; none receives greater emphasis than the others until it is selected by the user. From a user interface perspective, this visualization has the advantage that each entity is located within close proximity to its neighbors [13], rather than being distributed over a large area as in traditional charts.

3.1 Interactive relationship discovery

The mode of interaction is perhaps best communicated by example. We present a scenario of how a project manager

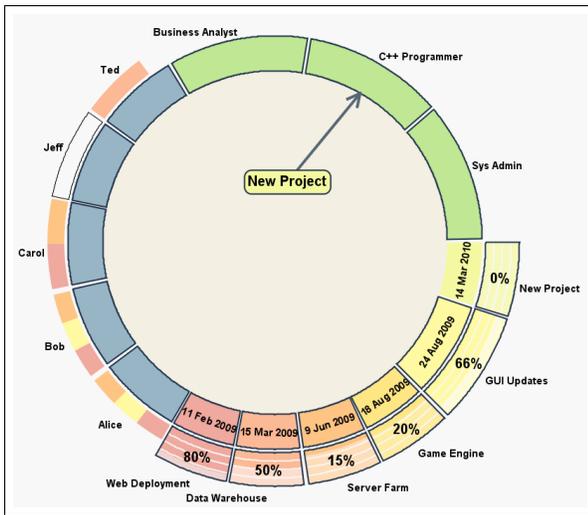


Figure 3: Relations between a project and a required skill.

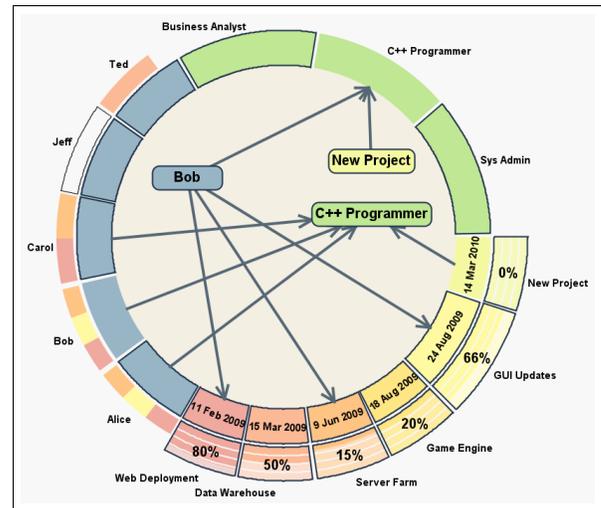


Figure 5: Relations between a resource, the skills it possesses, and its assigned projects.

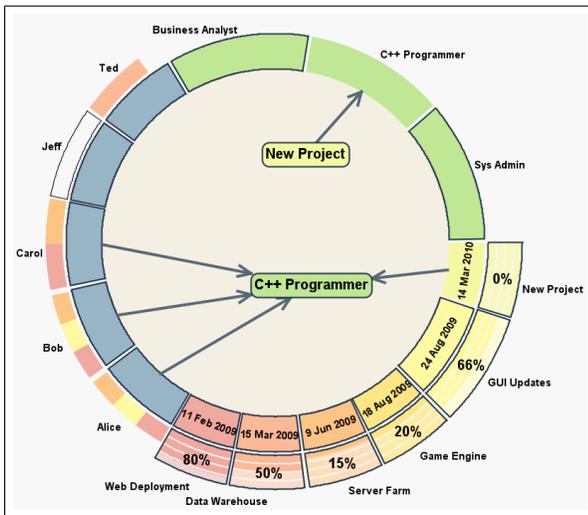


Figure 4: Relations between a skill, the projects requiring it, and the resources possessing it.

might use our visual query language to handle a typical task. For a more visual example, we refer the reader to a demo video (in QuickTime format) which may be downloaded here: <http://www.cs.utah.edu/~draperg/research/>.

Suppose that Jane, a team manager, receives a request for her team to take on a new project. The procedure described below outlines the steps that Jane might take in choosing which resources to allocate to this project, using our visualization.

Step 1 Jane selects the proposed project and brings it into the interior of the ring to discern its staffing requirements. Jane sees that this project requires a C++ programmer (Figure 3).

Step 2 Next, by introducing the skill labeled “C++ Programmer” into the ring’s interior, Jane recalls that there are three people on her team who have this skill: Alice, Bob, and Carol (Figure 4).

Step 3 Jane decides to look further into Bob’s profile. Placing Bob into the ring, she sees which projects he is currently involved

with: Web Deployment, Server Farm, and GUI Updates (Figure 5).

By examining the current commitments of Bob and others, Jane can make an informed decision regarding whether to accept this new project, and how best to distribute the workload among her team members.

In addition to the basic interaction metaphor outlined above, our visualization’s core strength is in allowing the user to uncover complex relationships within the data, beyond what can be effectively displayed in traditional charts. When multiple icons are placed in the interior of the ring, all known relations between these entities and the other entities are made visible. Whenever two or more arrows converge on the same point on the circumference (Figures 2 and 14), that is a clear visual indicator that these entities have a relationship in common.

For example, if two people consistently work on the same set of projects together, that might signal to their manager a need for them to collaborate more with others. Likewise, when multiple skills are placed inside the ring at the same time, the user can immediately see the distribution of skills among the team, and whether there are enough resources to fill the requirements of the team’s assigned projects. Or, if multiple projects are selected, the relationships of skills and people to projects become visible. In each of these examples, there is admittedly some overlap on the information being presented. The key is the flexibility of our approach — the user is free to select the method that best lends itself to the question he or she is trying to answer.

3.2 Space-saving visualization

One advantage of our method is its conservative use of space. In a standard assignment chart, every time an entity participates in a relationship, a separate instance of that entity must be included in the chart — thus linearly increasing its overall size. In our visualization, the size of the ring is constant, and each entity is shown at most twice: once along the circumference, and once inside the ring if that entity is currently selected by the user.

For example, consider the chart shown earlier in Figure 1. For each project to which Betty is assigned, a separate icon for Betty must be added to the chart. Contrast this with the visualization in Figure 2: Betty’s multiple assignments are still clearly indicated; however, Betty’s name appears only once in the ring’s interior. Perhaps more importantly, the overall “footprint” of the visualization

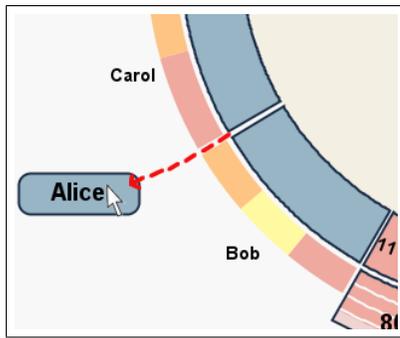


Figure 6: Control level of detail by removing unwanted entities from the ring.

does not change, regardless of how many relationships are shown.

3.3 Scalability and level of detail

One known weakness of our current approach is that the visualization does not scale well as the number of nodes exceeds about 50 or 60. This is due to the constraint of fitting all nodes within the limited space around the ring’s circumference. This is an issue shared by most radial visualization systems in general. Nonetheless, we offer two suggestions for ameliorating the scalability problem in our visual language.

First, any icons that are not of current interest to the user may be dragged outside of the ring (Figure 6). This not only reduces crowding around the circumference of the ring, but reduces the number of relations that are displayed for icons inside the ring, because no lines are drawn for nodes that are not currently on the circumference. Whenever the user wishes to include this entity again, its icon can be dragged back onto the circumference of the ring. In this manner, the amount of detail displayed about any resource, project, or skill is configurable on the fly — allowing the user to view as much or as little information about a given entity as desired.

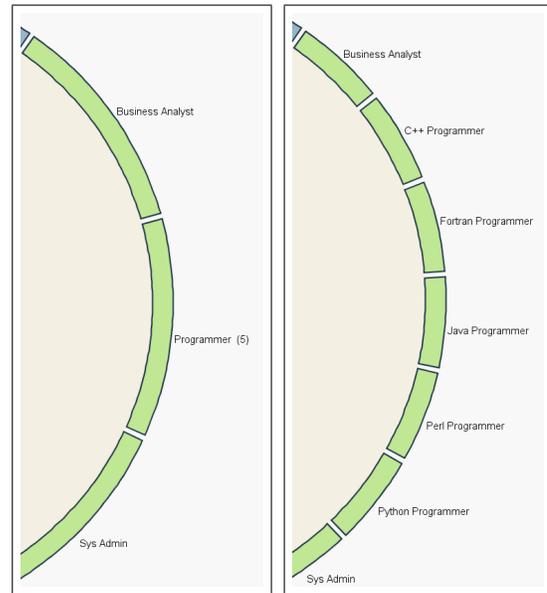
A second way to address the visualization’s scalability is through aggregating similar entities. For example, if a team contains people with skills in several different programming languages, all these skills could be aggregated into one composite skill, labeled “programming” (Figure 7(a)). At the user’s option, this skill could be expanded to reveal the individual skills (Figure 7(b)). Similarly, individual resources could be aggregated into teams of resources, and projects could be grouped according to categories of related projects. Furthermore, these aggregate entities may be combined into still more generalized entities until a desired level of abstraction has been achieved. In this manner, scalability is not so much limited by the size of the circle, but by the level of detail the user wishes to see [42].

In practical usage, a manager over a team of engineers would likely want to see his or her resources, projects, and skills at the lowest level of granularity. A middle-tier manager or executive, on the other hand, might prefer a broader, less detailed overview of what the organization as a whole is doing.

Our prototype system, described in Section 6, currently supports the first of these two proposed techniques. The second is an area for future work.

4 DIFFERENCE VISUALIZATION

Once one succeeds at visualizing a single data set, a logical next step is to visualize the differences between two data sets. By far, the most common domain for difference-finding is in text documents. A *de facto* tool for file comparison is `diff`, which has been included in the Unix operating system since 1974.



(a) A composite skill. (b) Detailed view of skills.

Figure 7: Composite and expanded entities.

However, `diff`’s output, which uses ‘>’ and ‘<’ symbols to signal differences between files, is not obvious for beginning users. Current versions of `diff` optionally support more intuitive symbols, such as ‘+’ and ‘-’, to denote insertions and deletions within the text.

Although the algorithmic problem of isolating differences between two text files is essentially solved, comparatively little work has been done on the problem of finding differences in other kinds of data, e.g. graphs and charts, although there are important exceptions [12, 14, 31, 49].

We propose two complementary visualizations for comparing different versions of a chart. The first is based on a traditional chart-based representation, while the second uses an interactive radial approach, similar to the one presented above.

4.1 Chart Visualization

Our visual `diff` metaphor is based on the observation that between any two versions of a chart, a link between two nodes can either be:

- present in both charts,
- present in the first chart and absent in the second chart, *or*
- absent in the first chart and present in the second chart.

Our visualization is inspired by modern implementations of `diff` that use the ‘+’ and ‘-’ characters to signal changes in the data. A link that is present in the first chart, but absent in the second chart is shown by drawing a ‘-’ in the icon adjacent to the link. A link that is absent in the first chart, but present in the second chart, is similarly depicted by a corresponding ‘+’. Figure 8 shows an example of our representation. This visualization is significantly more compact than simply putting the two versions of the chart side by side.

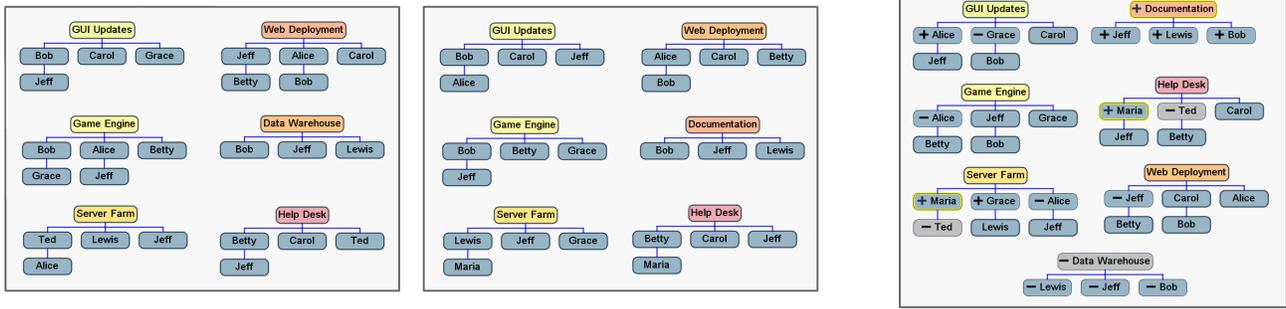


Figure 8: The leftmost chart represents Version 1 of a chart; the middle chart is Version 2. The chart on the right is a “Visual Diff” of the two versions.

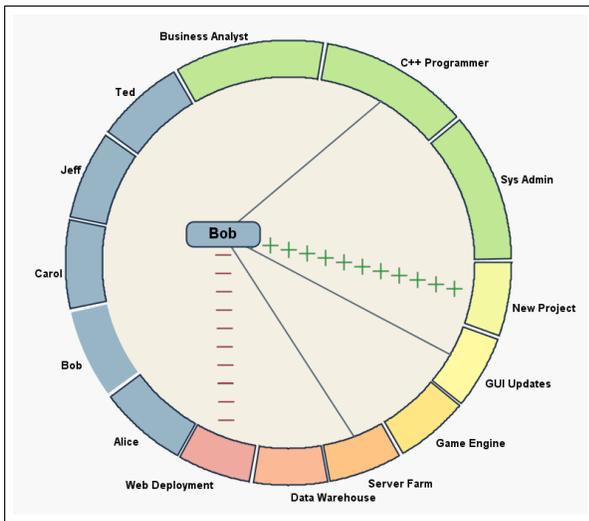


Figure 9: Interactive radial representation of a “Visual Diff” chart.

4.2 Radial Visualization

Our second representation for finding differences is an interactive visual language based on the radial query metaphor introduced in Section 3. As before, icons representing the resources, tasks, and skills are arranged in a ring. Icons shown in yellow highlight are those that are new to the second version of the chart. Icons shown in a muted gray color are those that appeared in the first version of the chart, but were absent from the second. When the user drags the node’s icon into the ring’s interior, links are rendered as lines between the interior node and one or more nodes on the circumference of the ring. Now, however, the line style indicates the status of a link. An old link, that is, a link that existed in the first chart but not the second chart, is rendered as a sequence of ‘-’ symbols. A new link, one that is unique to the second chart, is rendered as a line of ‘+’ signs. Predictably, a link that exists in both versions of a chart is rendered as a solid line (Figure 9).

5 “ONE-TO-MANY” CHARTS

Although we have focused thus far on the application domain of assignment charts, project management is not the only setting in which charts like this may be used. An assignment chart is simply a specific instance of a more general class of charts which we call “one-to-many” charts. In an assignment chart, for example, each (one) project is linked to all the (many) people who are assigned to it.

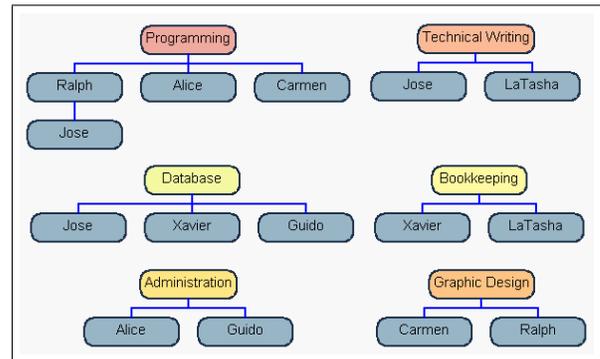


Figure 10: Graph showing skill distribution among team members

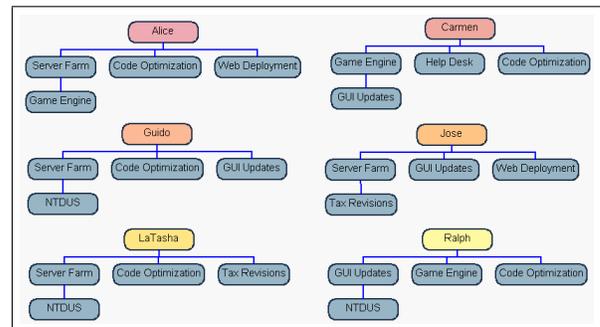


Figure 11: “Inverted” assignment chart

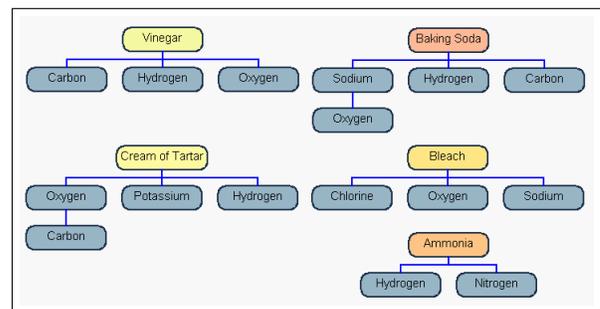


Figure 12: Graph of ingredients in household chemicals

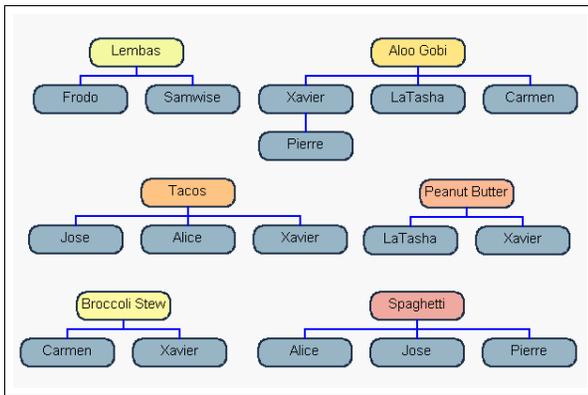


Figure 13: Graph of individuals' culinary preferences

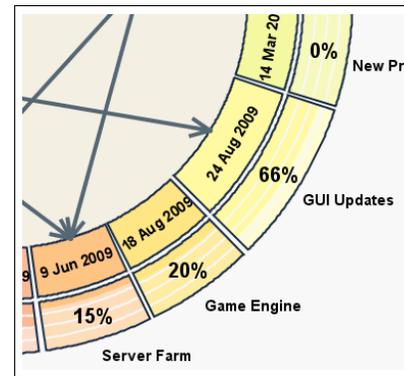


Figure 15: Project icons.

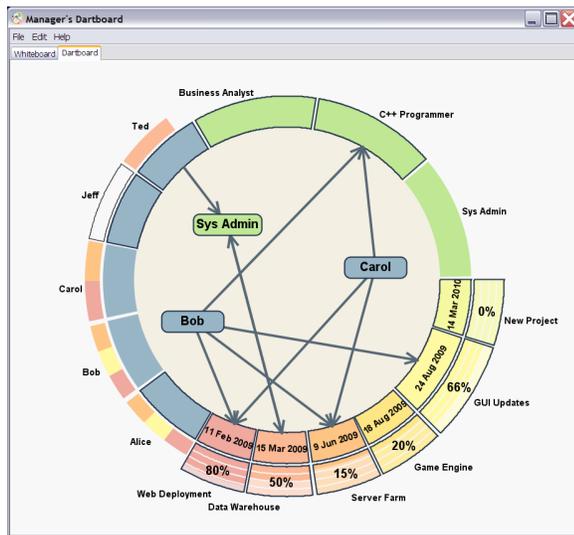


Figure 14: Screenshot of the prototype system, *Manager's Dartboard*.

Although assignment charts are a common type of one-to-many chart, they are certainly not the only kind. Consider a “skill chart”, showing which skills are possessed by individuals within an organization (Figure 10). Or an “inverted assignment chart” might show the tasks to which a resource is assigned, as opposed to which resources are assigned to a task (Figure 11). Stepping away from our “project management” motif, one could imagine a chart that displays the names of atomic elements alongside which household chemicals they are found in (Figure 12), or a chart that shows which people enjoy certain types of food (Figure 13).

As the above examples show, this class of chart has a variety of applications. The interactive visual language presented in this paper could be easily adapted to any one of these domains.

6 MANAGER'S DARTBOARD

As a proof of concept of the visualization methodology presented herein, we have implemented a prototype software system called *Manager's Dartboard* (Figure 14).

In order to free the user from the burden of frequently referring to another part of the interface (or to another program entirely) for supplementary information, we have tried to make the icons as informative as possible without making our interface overly busy. One purpose for doing this is to allow the user to maintain focus

on the visualization task, rather than having to perform a time-consuming (and potentially distracting) “context switch” to find the information elsewhere.

As such, *Manager's Dartboard* supports multiple levels of inquiry:

1. The icons themselves provide basic information about a particular entity at first glance.
2. More detailed information is available via a tooltip if the user hovers the cursor over the icon.
3. The menu system provides access to dialog boxes allowing the user to view and edit all of an entity's attributes.

We feel that this *details on demand* approach [50] strikes a fair balance between the opposing ideals of maintaining a clean user interface, and providing detailed information on specific entities as required by the user.

6.1 Project Visualization

In our system, each project possesses a unique 4-tuple of attributes: *name*, *end date*, *percent complete*, and *priority*. We designed the project icon so that each of these attributes is immediately visible without further inspection.

First and most simply, the *name* of the project is printed next to the icon. Also, the expected *end date* for each project is displayed prominently in the icon's interior. Furthermore, the icons are positioned around the ring counterclockwise, sorted in order of the project's end date. In addition, the icon also shows the progress of the project in terms of *percent complete*. The percentage is shown textually in bold type, and graphically as a curved progress bar. Finally, the *priority* of a project is suggested by the icon's radial magnitude (i.e. its “length”) relative to the other project icons (Figure 15), with higher priority projects occupying more space than the others.

In this manner, the user can quickly scan the list of projects to see which projects are due first, their relative priority, and how far they are from completion. This helps the project manager readily discern which projects may be in need of further attention.

6.2 Resource Visualization

The resource icons are designed to let the user quickly see both how many projects a resource is assigned to, and — perhaps more significantly — what proportion of the resource's time is allocated to each project. A resource's assigned projects are represented as a band of colored sectors along the radial exterior of the resource icon. We call this band the *project previewer* because it provides the user a hint of how busy each resource is, without needing to drag

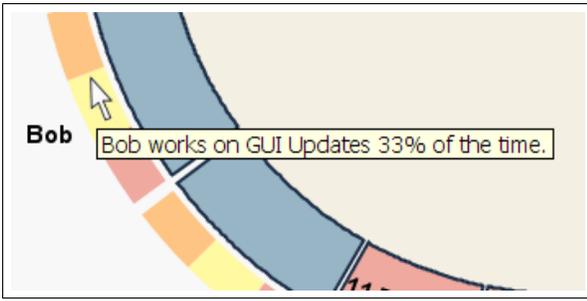


Figure 16: Hovering the cursor over an icon reveals additional details.

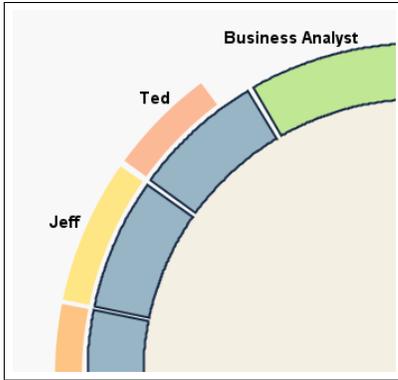


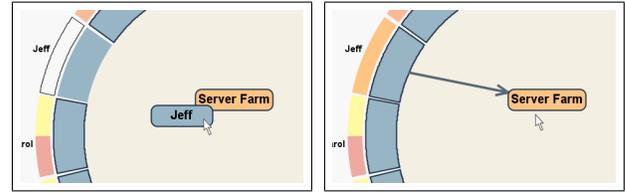
Figure 17: The size of the “project previewer” shows the comparative capacity of resources.

the resource icon into the ring’s interior. The color of each sector in the project previewer corresponds to the color of the corresponding project icon elsewhere on the ring’s circumference. Furthermore, the relative size of a sector as compared to its neighbors is a visual indicator of how much of the resource’s time is devoted to the given project. To reduce the amount of visual clutter in the display, we do not print the names of the projects alongside the project previewer; however, this information is available via a tooltip when the user hovers the cursor over the project previewer (Figure 16).

Even on a highly talented team, no two resources are likely to have the same levels of experience or competency, even though there may be overlap in their skill sets [26, 35]. For this reason, our system allows resources to be assigned a “capacity” percentage to reflect their level of ability relative to other resources. Visually, this is reflected by the length of the project previewer relative to the resource’s icon. For example, if Ted is a newer employee, his capacity might be 75% of Jeff’s, who has more experience. Thus, his project previewer only takes up three-fourths the length of his resource icon, whereas Jeff’s runs the full measure (Figure 17). Note that this does not prevent Ted from being assigned as many projects as Jeff or anyone else; however, the smaller size of his project previewer serves as a visual reminder to managers that he has proportionally less capacity to offer to his projects than do perhaps his counterparts.

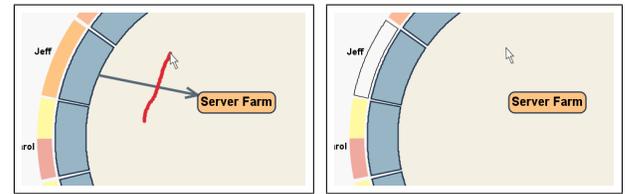
6.3 In-place editing

In keeping with our goal of giving the user maximum flexibility, we have also implemented a series of gestural interfaces to allow the user to edit relationships among entities directly within the display. The user may assign resources to projects, indicate skill requirements for projects, associate resources with their respective skills, as well as unassign or dissociate any of the above relationships. The gesture for associating two entities is accomplished by the familiar



(a) Drag an icon atop another icon to associate them. (b) The relation is created.

Figure 18: Interface for creating relations.



(a) The “cutting” gesture. (b) The relation is broken.

Figure 19: Interface for removing relations.

drag and drop metaphor.

This gesture is illustrated in Figure 18. Figure 18(a) shows the user dragging the icon for a resource, Jeff, atop the icon for project Server Farm. Alternatively, the icon representing the project could have been dragged atop the icon representing the resource. When the link is created, the dragged icon disappears and an arrow is rendered between the icon inside the circle and the icon on the circumference, as shown in Figure 18(b). The color of Jeff’s project previewer likewise changes to show that Jeff, who was not assigned to any projects, is now associated full-time with the Server Farm project.

The gestures for associating projects and resources with skills are analogous — simply drag and drop the icons representing the desired entities atop each other, and the system creates the underlying association.

6.3.1 Sketch-based Interface

Motivated by the interface of [23], the gesture for removing a relation between two entities consists of drawing a straight line which intersects the arrow connecting two entities. For example, in Figure 19(a), the user draws a line bisecting the arrow connecting the icons for resource Jeff and project Server Farm. In Figure 19(b), notice how the color of the resource’s project previewer changes to reflect the fact that Jeff is no longer assigned to any projects.

To reduce the chance of error, our system requires that the input gesture for removing an association between entities be an approximately straight line rather than a curved one. We do this by comparing the total length of the *ink stroke* with the distance between its endpoints. A gesture is considered straight if and only if it satisfies the conditions in Equation 1 below.

Let $\rho_{1..n}$ be the vertices of the gesture as inputted by the user.

Let $\delta(i, j)$ be the Euclidean distance between vertices ρ_i and ρ_j .

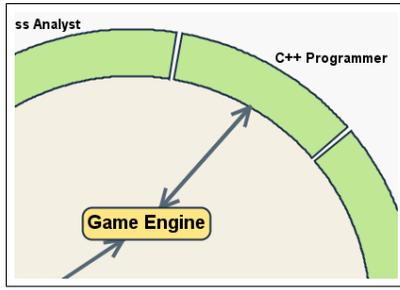


Figure 20: A project whose assigned resources match its skill requirements.

Let τ be the threshold above which the line is considered straight.

$$\frac{\delta(1, n)}{\sum_{i=1}^{n-1} \delta(i, i+1)} \geq \tau \quad (1)$$

In our experience, setting $\tau = 0.8$ is sufficient for determining the straightness of a line.

6.4 Directed Relations

Our system renders connections between icons as arrows instead of undirected line segments. This was a conscious design choice, intended to make explicit the nature of the relationship between entities. Specifically, arrows are drawn from resources to projects, rather than the inverse. We render arrows from projects to skills to show a dependency relationship: projects require skills. Arrows are drawn from resources to skills to show that resources possess skills. These design decisions are somewhat subjective; other valid mappings could likewise be employed. However, for consistency of presentation, we have adhered to these guidelines for our prototype implementation.

For convenience, we draw a double-headed arrow between a project and a skill whenever a resource possessing a certain skill is allocated to a project requiring that skill (Figure 20).

In our system, we have opted to draw arrows only between an icon on the circumference of the ring and another icon inside the interior of the ring. We do not draw arrows between two icons that both lie on the circumference of the ring. Contrast this to the approach used by [7] and [16]. Similarly, we do not draw arrows from one icon in the interior of the ring to another icon also in the interior of the ring. These decisions were made in order to reduce the amount of “noise” in our visualization, and in light of the fact that no additional information is conveyed by displaying an individual relation more than once.

6.5 Whiteboard

In addition to the “Dartboard” interface which demonstrates the concepts of our research, our software prototype also includes an alternate “Whiteboard” user interface, based on the familiar notion of an assignment chart. The user can switch between the two views by selecting their respective tabs. Both the Whiteboard and the Dartboard views operate on the same data, albeit via a different user interface. When a change to the data is made in either view, that change is automatically reflected in the other.

Whiteboard is little more than a computerized version of an assignment chart that one might find on a typical manager’s whiteboard (Figure 21). It allows the user to associate projects and resources by dragging and dropping icons onto a blank panel. Placing an icon near another icon associates the two; similar behavior can

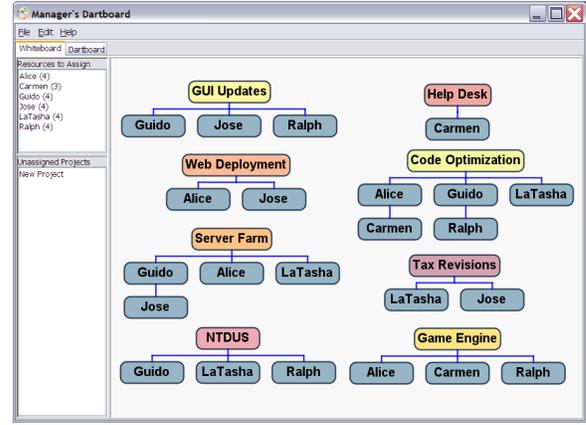


Figure 21: Alternate “Whiteboard” interface

be found in commercial diagram-drawing software like Microsoft Visio [38].

Its convenience of use notwithstanding, Whiteboard was never intended to be especially innovative. Indeed, we implemented Whiteboard solely as a benchmark against which to compare Dartboard, which is the actual focus of our work. For simplicity, the Whiteboard is missing some features that are supported by the Dartboard interface; most noticeably, the ability to manipulate skill attributes.

6.6 Implementation Notes

Our prototype is written in Java. We use the Swing API for the traditional user interface elements (such as the menu system and dialog boxes) and Java2D for the graphics.

Having successfully tested Manager’s Dartboard on Linux, Solaris, and Windows platforms, we believe it should run equally well on any platform for which a Java 1.6-compliant virtual machine exists.

7 USER EVALUATION

To gauge the effectiveness of our visual language, we conducted an informal preliminary user trial consisting of 22 participants, primarily graduate students in computer science. Our goal was to compare the speed and accuracy with which users are able to identify relationship information from both a traditional assignment chart and our visualization. Our participants all performed the test using the same computer, a 2.5GHz Pentium4 PC with 500MB of memory, running Red Hat Linux 9. The screen resolution was set to 1600x1150 pixels.

The trial was in two parts. During Part 1, we showed an assignment chart to the participants, and asked them seven questions based on the chart. The chart was on-screen during the question-answering period so that the participants could refer to it as needed. For Part 2, we introduced them to the Manager’s Dartboard system and gave them time to familiarize themselves with its interface. We then presented the participants seven additional questions similar to those in Part 1, letting them use our software to explore the data. Manager’s Dartboard displayed all the same resources, projects, and skills as in the assignment chart from Part 1, although we did scramble the project and skill assignments to prevent participants from parroting back the same answers as in Part 1.

The questions consisted of a combination of simple and compound queries, namely:

1. Which projects is Cindy working on?
2. Which projects are both Jeff and Daniel working on?

3. Who is assigned to the greatest number of projects?
4. Which people are assigned to work on exactly three projects?
5. Which one person is assigned to all of these projects: GUI Updates, Web Page Update, Workshop?
6. Which people always work together on the same projects?
7. What skills does Alice have?

Although we initially hypothesized that our visualization's chief advantage would be speed, not accuracy, we were pleasantly surprised to find that the participants not only answered the questions 13% faster on average using Manager's Dartboard, but that they also made an average of 80% fewer mistakes. We interpret this as a sign that our method for visual correlation offers a strategic advantage over traditional charts — especially when applied to mission-critical application domains where misinterpretations of data may be costly.

Subjective feedback from users was likewise positive. Upon completing the questions, the majority of participants commented on how much easier Manager's Dartboard was to use than a static chart, especially for determining working relationships among the resources, projects, and skills. While the informal nature of our study makes it difficult to draw definitive conclusions, we are nonetheless encouraged by these preliminary results, and we would like to validate them with additional user studies in the future.

8 FURTHER WORK

In our prototype's current state of development, all projects have deadlines. However, some tasks, termed *workflows* by Craven and Mahling [6], are ongoing, repetitive assignments and do not have specified end dates. Examples of workflows could be maintenance, technical support, or committee assignments. We would like to implement support for workflows in the future.

We also plan to expand the ways in which entities can be related to one another. Currently, only entities of different types may be associated together: projects with resources, resources with skills, etc. Additional semantic information could be conveyed by associating entities of the same type. For example, affiliating one resource to another could perhaps express a mentor/trainee relationship, whereas linking one project to another project may designate a chronological or dependency relationship.

Furthermore, we intend to incorporate additional gesture-based commands into the user interface. For example, the proportion of a resource's time that is spent on a given project is currently editable via the menu system and dialog boxes. In the ideal situation, this attribute could be edited by direct manipulation within the visualization canvas.

9 CONCLUSION

In this paper, we have presented a novel method for visualizing and editing resource allocation data in the domain of project management. Our current prototype supports three types of entities; namely projects, resources, and skills; however, our visualization concept is general enough to accommodate many more.

The main contributions of this paper are:

- a highly interactive canvas for querying multivariate data,
- a gesture-based interface for adding and removing links between nodes, and
- two complementary visual schemes for comparing the differences between two versions of a data set.

Our visual language aims to be a general-purpose strategy for discovering relationships among disparate entities in a wide variety of data sets. Users report that this interface is satisfying to use; furthermore we have found that complex relationships within a set of data can be identified more quickly and accurately using our method than with standard charting techniques.

ACKNOWLEDGEMENTS

The authors wish to thank Glenn McMinn, CEO of Delcam USA, and Tom McCollough, VP of software development at Delcam USA, for their helpful feedback on an early prototype of this system. We also thank all the participants in our preliminary user study.

This work was supported in part by the Scientific Computing and Imaging Institute at the University of Utah, and by the Air Force Research Laboratory in Rome, NY, under IC-ARDA contract no. F30602-03-C-0257.

REFERENCES

- [1] C. Ahlberg and B. Shneiderman. Visual information seeking: tight coupling of dynamic query filters with starfield displays. In *CHI '94*, pages 313–317, 1994.
- [2] E. Bertini, P. Hertzog, and D. Lalanne. SpiralView: Towards Security Policies Assessment through Visual Correlation of Network Resources with Evolution of Alarms. In *IEEE Symposium on Visual Analytics Science and Technology (VAST 2007)*, volume 2, pages 139–146, Oct. 2007.
- [3] M. Cammarano, X. L. Dong, B. Chan, J. Klingner, J. Talbot, A. Halevy, and P. Hanrahan. Visualization of heterogeneous data. In *InfoVis 2007*, pages 1200–1207, 2007.
- [4] J. V. Carlis and J. A. Konstan. Interactive visualization of serial periodic data. In *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 29–38, New York, NY, USA, 1998. ACM Press.
- [5] T. Catarci, M. F. Costabile, S. Levialdi, and C. Batini. Visual query systems for databases: a survey. *Journal of Visual Languages and Computing*, 8(2):215–260, 1997.
- [6] N. Craven and D. Mahling. Goals and processes: a task basis for projects and workflows. In *COCS '95: Proceedings of conference on Organizational computing systems*, pages 237–248, New York, NY, USA, 1995. ACM Press.
- [7] Daisy Analysis Ltd. Daisy 2003. <http://www.daisy.co.uk/>. Accessed 14 November 2007.
- [8] M. C. F. de Oliveira and H. Levkowitz. From visual data exploration to visual data mining: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):378–394, July 2003.
- [9] D. Dixon. Integrated support for project management. In *Proceedings of the 10th international conference on Software engineering*, pages 49–58, 1988.
- [10] G. M. Draper and R. F. Riesenfeld. Interactive fan charts: A space-saving technique for genealogical graph exploration. In *8th Annual Workshop on Technology for Family History and Genealogical Research*, Provo, Utah, USA, 2008. Brigham Young University.
- [11] N. Elmqvist, J. Stasko, and P. Tsigas. DataMeadow: A Visual Canvas for Analysis of Large-Scale Multivariate Data. In *IEEE Symposium on Visual Analytics Science and Technology (VAST 2007)*, volume 2, pages 187–194, Oct. 2007.
- [12] C. Erten, P. J. Harding, S. G. Kobourov, K. Wampler, and G. Yee. Graphael: Graph animations with evolving layouts. In *International Symposium on Graph Drawing*, 2003.
- [13] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. In *Journal of Experimental Psychology*, pages 381–391, 1954.
- [14] D. Forrester, S. G. Kobourov, A. Navabi, K. Wampler, and G. Yee. graphael: A system for generalized force-directed layouts. Available at <http://graphael.cs.arizona.edu/graphael/>, 2005. Accessed 22 August 2007.
- [15] G. W. Furnas and S. J. Rauch. Considerations for information environments and the NaviQue workspace. In *INEX Workshop*, pages 79–88, 2004.

- [16] J. Galloway and S. J. Simoff. Network data mining: methods and techniques for discovering deep linkage between attributes. In *APCCM '06: Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling*, pages 21–32, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.
- [17] R. Goldman and J. Widom. Interactive query and search in semistructured databases. In *International Workshop on the Web and Databases, WebDB'98*, pages 52–62, 1998.
- [18] G. Halin and D. Hanser. Vers une visualisation contextuelle de la conception coopérative. In *IHM 2003: Proceedings of the 15th French-speaking conference on human-computer interaction on 15eme Conference Francophone sur l'Interaction Homme-Machine*, pages 208–211, New York, NY, USA, 2003. ACM Press.
- [19] J. Hathaway. Project juggling: the evolution of a process. In *Proceedings of the 32nd annual ACM SIGUCCS conference on User services*, pages 247–249, 2004.
- [20] S. Havre, E. Hetzler, K. Perrine, E. Jurrus, and N. Miller. Interactive visualization of multiple query results. In *InfoVis 2001*, pages 105–112, 2001.
- [21] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, Jan. 2000.
- [22] B. Hetzler, P. Whitney, L. Martucci, and J. Thomas. Multi-faceted insight through interoperable visual information analysis paradigms. In *InfoVis 1998*, pages 137–144, 161, 1998.
- [23] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3D freeform design. In *SIGGRAPH '99*, pages 409–416, 1999.
- [24] B. S. Johnson. *Treemaps: Visualizing Hierarchical and Categorical Data*. PhD thesis, University of Maryland, 1993.
- [25] G. Jones. One solution for project management. In *Proceedings of the 29th annual ACM SIGUCCS conference on User services*, pages 65–69, 2001.
- [26] D. Joslin and W. Poole. Agent-based simulation for software project planning. In *Proceedings of the 37th conference on Winter simulation*, pages 1059–1066, 2005.
- [27] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, Jan. 2002.
- [28] D. A. Keim, F. Mansmann, J. Schneidewind, and T. Schreck. Monitoring network traffic with radial traffic analyzer. In *2006 IEEE Symposium On Visual Analytics Science And Technology (VAST)*, pages 123–128, 2006.
- [29] D. A. Keim, J. Schneidewind, and M. Sips. CircleView: a new approach for visualizing time-related multidimensional data sets. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 179–182, New York, NY, USA, 2004. ACM Press.
- [30] J. Kontio, J.-P. Jokinen, and E. Rosendahl. Visualizing and formalizing risk information: an experiment. In *Proceedings of the 10th International Symposium on Software Metrics*, pages 196–206, 2004.
- [31] G. Kumar and M. Garland. Visual exploration of complex time-varying graphs. In *InfoVis 2006*, pages 805–812, 2006.
- [32] H. Lee-Kwang and J. Farrel. The ssd graph: a tool for project scheduling and visualization. *IEEE Transactions on Engineering Management*, 35(1):25–30, Feb. 1988.
- [33] P. Lempp. A possibility to make project progress visible: the progress diagram within the epos system. *ACM SIGSOFT Software Engineering Notes*, 11(2):32–35, Apr. 1986.
- [34] Y. Livnat, J. Agutter, S. Moon, and S. Foresti. Visual correlation for situational awareness. In *InfoVis 2005*, pages 95–102, 2005.
- [35] M. Mantei. The effect of programming team structures on programming tasks. *Communications of the ACM*, 24(3):106–113, Mar. 1981.
- [36] B. Meyer. The software knowledge base. In *Proceedings of the 8th international conference on Software engineering*, pages 158–165, 1985.
- [37] Microsoft Corporation. Microsoft Project. <http://office.microsoft.com/project/>. Accessed 15 May 2007.
- [38] Microsoft Corporation. Microsoft Visio. <http://office.microsoft.com/visio/>. Accessed 15 May 2007.
- [39] Z. Mihajlovic and D. Velasevic. Tracking software projects with the integrated version control in smit. *ACM SIGSOFT Software Engineering Notes*, 26(2):38–43, Apr. 2001.
- [40] M. J. Muller, J. G. Smith, J. Z. Shohar, and H. Goldberg. Privacy, anonymity and interpersonal competition issues identified during participatory design of project management groupware. In *Proceedings of the 6th annual ACM SIGUCCS conference on User services*, pages 82–87, 1990.
- [41] F. Nightingale. *Notes on matters affecting the health, efficiency, and hospital administration of the British army : founded chiefly on the experience of the late war*. London: Harrison and Sons, 1858.
- [42] D. A. Norman. *The Design of Everyday Things*, page 209. Basic Books, 2002 edition, 2002.
- [43] M. L. Northway. *A Primer of Sociometry*. University of Toronto Press, 2nd edition, 1967.
- [44] J. P. Peddicord. A.p.i.s. academic project information system. In *Proceedings of the 6th annual ACM SIGUCCS conference on User services*, pages 74–81, 1978.
- [45] W. Playfair. *The statistical breviary; shewing, on a principle entirely new, the resources of every state and kingdom in Europe; illustrated with stained copper plate charts, representing the physical powers of each distinct nation with ease and perspicuity. To which is added, a similar exhibition of the ruling powers of Hindoostan*. London: for J. Wallis, 1801.
- [46] S. Polyviou, G. Samaras, and P. Evripidou. A relationally complete visual query language for heterogeneous data sources and pervasive querying. In *ICDE 2005. Proceedings of 21st International Conference on Data Engineering*, pages 471–482, 2005.
- [47] S. F. Roth, P. Lucas, J. A. Senn, C. C. Gomberg, M. B. Burks, P. J. Stroffolino, J. A. Kolojechick, and C. Dunmire. Visage: A user interface environment for exploring information. In *InfoVis 1996*, pages 3–16, 1996.
- [48] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*, page 46. Prentice Hall, first edition, 1995.
- [49] C. E. Scheidegger, H. T. Vo, D. Koop, J. Freire, and C. T. Silva. Querying and creating visualizations by analogy. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):1560–1567, 2007.
- [50] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *IEEE Symposium on Visual Languages*, pages 336–343, 1996.
- [51] V. Sinha and D. R. Karger. Magnet: Supporting navigation in semistructured data environments. In *SIGMOD*, pages 97–106, 2005.
- [52] I. Spence. No humble pie: The origins and usage of a statistical chart. *Journal of Educational and Behavioral Statistics*, 30(4):353–368, 2005.
- [53] J. Stamey and T. Honeycutt. Quad charts in software project management. *Journal of Computing Sciences in Colleges*, 21(2):322–327, Dec. 2005.
- [54] J. Stasko and E. Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *InfoVis 2000*, pages 57–65, 2000.
- [55] C. Stolte, D. Tang, , and P. Hanrahan. Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65, 2002.
- [56] M. Stonebraker. Visionary: A next generation visualization system for data bases. In *SIGMOD*, page 635, 2003.
- [57] P. Szekely, M. Becker, S. Fitzpatrick, G. Gati, D. Hanak, J. Jin, G. Karsai, R. T. Maheswaran, B. Neches, C. M. Rogers, R. Sanchez, and C. van Buskirk. Csc: Criticality-sensitive coordination. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 1441–1442, 2006.
- [58] A. Trigoni. Interactive query formulation in semistructured databases. In *5th International Conference on Flexible Query Answering Systems*, pages 356–369, 2002.
- [59] W. Wang and F. Lillehagen. A cooperative hypermedia solution to work management in real-time enterprises. In *Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, pages 196–197, 2003.
- [60] M. O. Ward. XmdvTool: Integrating multiple methods for visualizing multivariate data. In *IEEE Visualization*, pages 326–333, 1994.