Supporting Visual Problem Solving in Spatial Hypertext

Haowei Hsieh and Frank M. Shipman III
Department of Computer Science & Center for the Study of Digital Libraries
Texas A&M University, College Station, TX 77843-3112, USA
{haowei,shipman}@csdl.tamu.edu

ABSTRACT

This paper describes the VITE system, a spatial hypertext system that supports two-way mapping for projecting structured information to a two-dimensional workspace and updating the structured information based on user interactions in the workspace. VITE uses information visualization techniques to render structured information in the workspace and provides users an environment to interact with digital information in a spatial hypertext setting. The two-way mapping connects the objects in the workspace to the structured information and provides users direct access to the information. The spatial hypertext environment encourages users to engage more directly with the information related to their tasks. An evaluation of VITE was conducted to study how people adapt to two-way mappings and how two-way mappings can help in problem solving tasks. The results show that users could quickly design visual mappings to help their problem-solving tasks and developed more sophisticated strategies for visual problem-solving over time.

Keywords

Information visualization, spatial hypertext, information workspace, information interfaces and presentation

1. INTRODUCTION

Information generated and manipulated in computer systems has become more formalized and structured in order to let computers process it. In this discussion, *formalized* or *structured information* is that which is represented with a pre-defined schema or a set of attributes, values, and relations. Examples include data records in a relational database that contain a set of data fields, frames, or objects with attributes in a knowledge base. As opposed to how computers process information, people often prefer less-formal representations in order to perceive things as a whole and to delay the abstraction process [Shipman and Marshall 1999].

For people to make use of structured information in certain tasks, they must interpret and manipulate the structured representation and the information encoded in this representation. Although it is possible for users to access formalized information directly, the interfaces used rarely support the representation and manipulation of less structured information related to the formally-represented content. The reason for this is two-fold. First of all, the formalization process deconstructs the original information, and cannot express knowledge outside of that envisioned by the schema designer. The lossy nature of the formalization process is why Shipman called for “nondestructive formalization” during knowledge building that keeps the original “less formal representation as well as the formal representations” [Shipman 1993]. Secondly, in real task-based decision making processes, users often generate intermediate information not part of the formalized information. Formal representations rely on the representation’s designer anticipating use situations. Work practices that start and end with structured information may make use of other informal or formal representations in between. For example, while dividing a set of items into two categories a third category may appear for those that are still undecided.

1.1 Problem of Incomplete Representation

Structured information is often only a partial abstraction of the information it represents. In these cases, formalized data loses certain aspects of the information it intends to represent, and the resulting discrete data chunks are not meaningful if not treated properly. For example, in a library database, each book is represented as an information object. Books are described within a designed schema, with attributes such as title, authors, publishers, etc. Attributes abstracted from the book information enable the database to be searchable by indexing, and sorting these attributes. However, the attributes cannot represent the whole book. If a library patron asks for the “thick gold book by Stephenson,” the book color and size must be part of the schema for them to become searchable. In the end, there is a never ending set of attributes.

The inevitably incomplete nature of the representation [Suchman 1987] does not imply the representation is not useful, just that it is not a replacement for the original entity. Deciding what characteristics to represent and how to represent them is situation dependent.

1.2 Problem of Losing Intermediate Representations for Problem Solving

Problem solving usually involves information gathering, categorizing, and knowledge building [van Mechelen et al. 1993]. With database systems, structured information can be categorized or merged using indexing techniques, queries, or other database operations. However, the process of knowledge building remains a highly creative activity not mastered by computers. Without proper support, translating and using the structured information during knowledge building processes can be cumbersome.

Formal representations predefine many of the categories and concepts available. This removes the opportunity for information consumers to take part in the concept building process. Presentations of the structured information that enable intermediate category and concept formation are needed. Such intermediate representations can take the form of less formal representations attached to the structured data. This semi-structured information may not be the end product of the decision making process, but can be essential in reaching the final decision. Malone et al. discuss further advantages of semi-structured information to that defined in a schema [Malone et al. 1986].

2. STRUCTURED INFORMATION IN SPATIAL HYPERTEXT

The above mentioned problems are due to the differences between representations designed for human perception and use, and those designed for computer manipulation. One way to address these problems is to provide a user interface that allows people to work with structured information in an interface appropriate for their task. The interface needs to present information to the user and allow users to interact with the information. For example, database systems normally provide either table- or form-based interfaces to facilitate data presentation and data editing. More complicated interactions with the database can often be accomplished through a command-line interface or pre-fabricated applications.

For many tasks, a better way to present information to users is through information visualization [Card et al. 1999]. Visualization techniques utilize both retinal properties and spatial arrangement for the presentation of structured information in a way that is more natural for human perception and understanding.

2.1 Visualization: Presenting Structured Information

Visualizations reveal broader patterns within information sets and help people recognize characteristics of the information set as a whole. The process has the inherent benefit of lower cognitive overhead [Hutchins et al. 1986] by providing temporary external memory [Norman 1988] and revealing spatial relations. In the process of presentation, some of the knowledge lost in the abstraction process may also become apparent. Visualization may not only convey content and relations lost during formalization, but can also reveal implied or derived information to the user.

Most visualization systems focus on transforming structured data into graphical displays according to unidirectional mapping constraints such as which characteristics of the data can be mapped onto visual attributes. Visualization helps users perceive and understand information spaces [Card et al. 1991], but these systems are generally designed either for a specific domain or provide for only a small set of pre-determined visual mappings. Card et al.'s Information Visualizer [Card et al. 1991] implements the concept of information workspace by providing an environment tailored for information-based processes. Some systems are designed to provide an expanded visualization space by including 3D, animation [Robertson et al. 1993] or view transformations, such as the perspective wall [Mackinlay et al. 1991], cone trees [Robertson et al. 1991] and fisheye views [Sarkar and Brown 1992]. Kumar's Timeline project [Kumar et al. 1998] provides an example of the assignment of visual mappings for specific tasks.

2.2 Manipulating Spatial Information

Databases and file systems provide tools for the storage, access and maintenance of information. These systems provide little help for the more idiosyncratic practices surrounding the piece-wise manipulation and management of information, except through the form-based interfaces for data entry and report generation that are commonly found in integrated database packages.

Information visualization techniques bridge the gap between structured information and user-friendly representations because they take advantage of human visual perception [Tufte 1990]. Many visualization systems allow interactivity with, and the configuration of, the visualization. However, most visualization systems do not support the visual editing of the structured information. The lack of direct manipulation of structured information in visualization systems means that there is no expression in such an environment, and expression is part of real decision making processes.

Visual workspaces containing directly manipulable information objects take advantage of human spatial cognition and immediate feedback from the visual environment [Wærn 1997]. Direct manipulation is common in operating system desktops and in drawing and graphics software due to the intuitive and efficient interaction provided for the user. Document management systems such as Presto [Dourish et al. 1999] and Data Mountain [Robertson et al. 1998] use directly manipulable visual workspaces for exploring a natural form of interaction. These systems allow users to arrange and locate objects in the space via instinctive 2D interactions. However, manipulation of these objects is for organization purposes rather than modifying their underlying content.

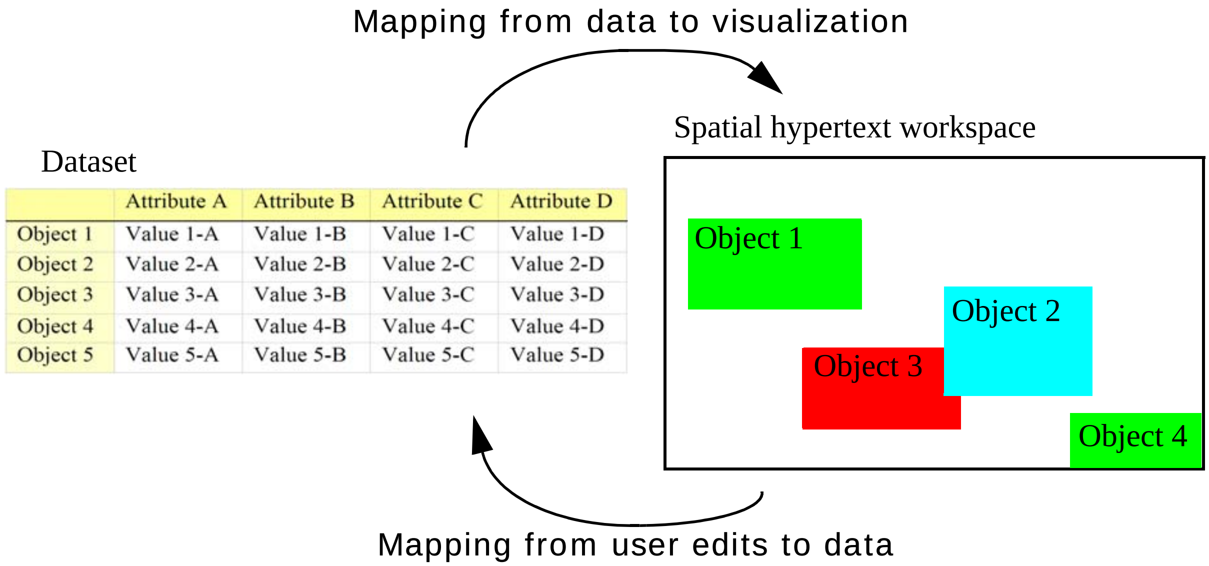
2.3 Two-Way Visual Mappings in a Spatial Hypertext Workspace

As described, visualization systems often let users change the way the system presents information but not the information itself. Visualization systems are what we call one way mapping systems where visualization settings merely specify how the information is rendered and presented. Users interact with visualization system by modifying the visualization mapping. The approach presented here extends this notion of interaction so that the interaction is between the user, the visualization system, and the underlying structured information. The extended interactions include the editing and switching of visual mappings (which exist in visualization systems), and the direct editing of the structured information within the visualization.

Visual workspaces combining visualization with direct manipulation enable users to visually and kinesthetically work with information objects. Combining visualization with editing through manipulation requires a mechanism to reflect user's changes to the information. Our approach is to take the unidirectional mapping of visualization systems, and make it bidirectional. These bidirectional mappings (we call them two-way mappings) visualize the structured information (as visualization systems do), as is shown by the top arrow in **Figure 1**. Two-way mappings also parse visual-property changes made by the user, represented by the bottom arrow in **Figure 1**. This poses new mapping constraints when compared to visualization, since user edits must be automatically interpretable. Ordinal, nominal, and continuous valued attributes place different constraints on the two-way mapping.

3. VITE: A VISUAL WORKSPACE SUPPORTING TWO-WAY MAPPINGS

We have developed VITE to explore the design and use of systems incorporating two-way mappings in spatial hypertext. An early version of VITE was reported in [Hsieh and Shipman 2000]. VITE's design and development have been influenced by a variety of prior work in the area of



	Attribute A	Attribute B	Attribute C	Attribute D
Object 1	Value 1-A	Value 1-B	Value 1-C	Value 1-D
Object 2	Value 2-A	Value 2-B	Value 2-C	Value 2-D
Object 3	Value 3-A	Value 3-B	Value 3-C	Value 3-D
Object 4	Value 4-A	Value 4-B	Value 4-C	Value 4-D
Object 5	Value 5-A	Value 5-B	Value 5-C	Value 5-D

Figure 1: A two-way mapping connects the structured data to the information objects in spatial hypertext workspace.

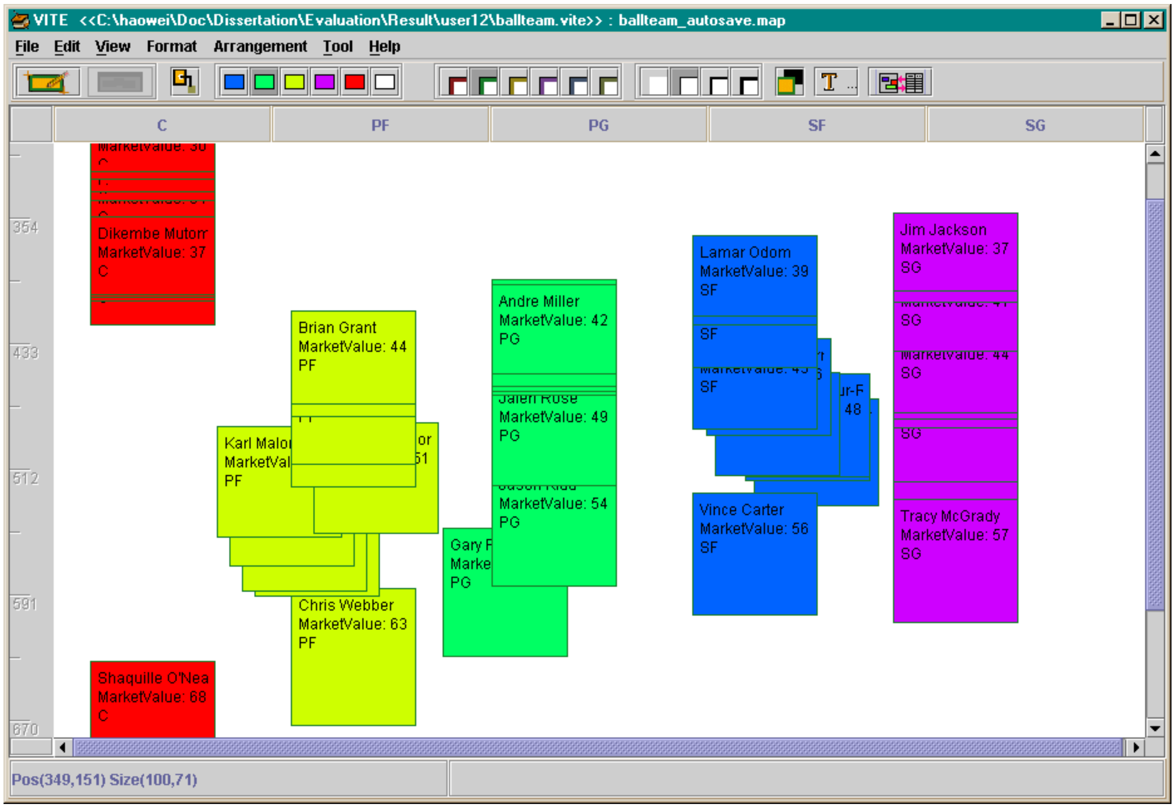
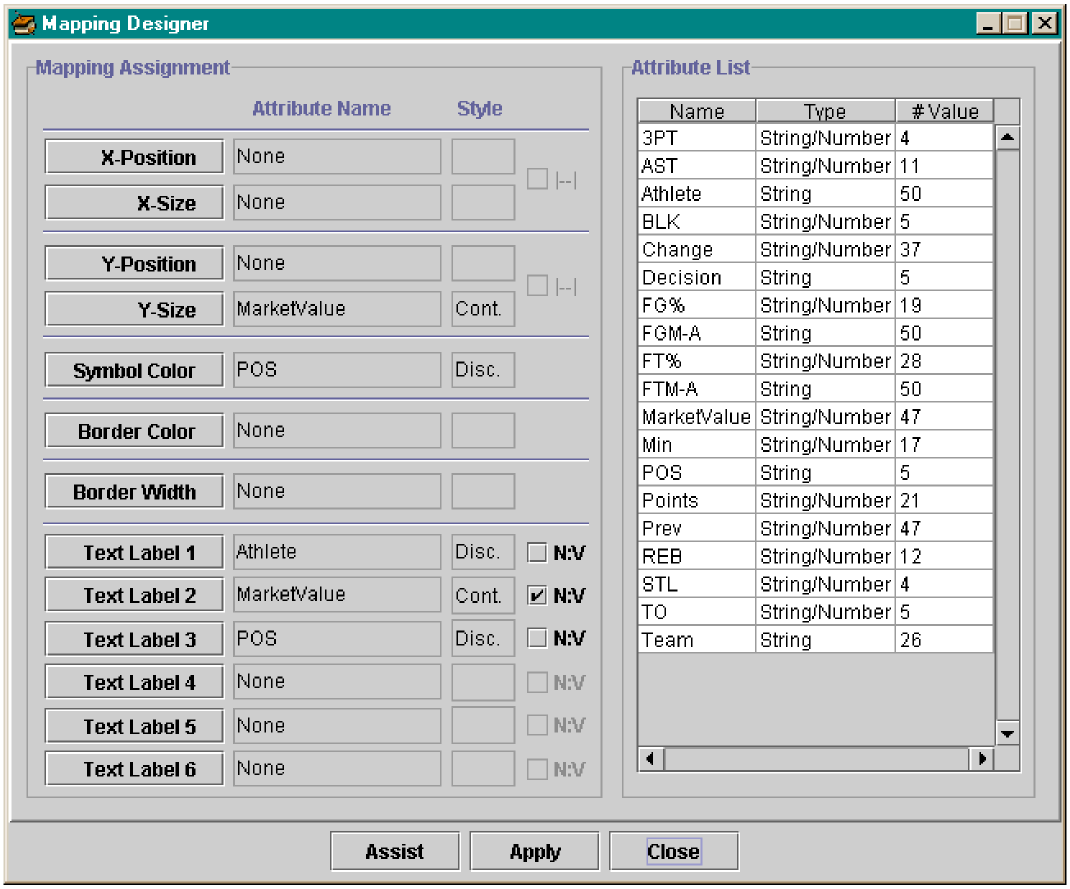


Figure 2: Workspace containing visual symbols representing basketball players being organized during selection of a fantasy basketball team.

spatial hypertext and visualization. Systems like Aquanet [Marshall et al. 1991] and VIKI [Marshall and Shipman 1997] use spatial layout for arranging relationships among information objects. HOS [Shipman and McCall 1999] and Aquanet added an abstract layer by allowing attributes, relationships and types to be associated with information objects for augmented semantic or annotation. Tivoli [Moran et al. 1997] incorporated specific domain knowledge into the system to support implicit perception in a free-form interaction environment. Visual Knowledge Builder (VKB) [Shipman and Hsieh 2000] utilizes both spatial layout and abstract attributes for supporting incremental formalization within user-created information workspaces.

VITE expands on the design of VKB by using spatial layout for arranging information, but its attributes are the primary content rather than an augmentation to the information object. VITE is different from the systems including formal representations in that it uses an existing structured data source such as a database table. The spatial workspace is a tailorable visualization of the structured data enabling the user to edit the structured data by visually manipulating the information objects.

VITE includes four major components: the visual workspace used to view and edit the information, the mapping designer used to create and edit two-way mappings, the mapping engine which instantiates the two-way mappings, and the data store for accessing structured information.



	Attribute Name	Style	
X-Position	None		☐ |--|
X-Size	None		
Y-Position	None		☐ |--|
Y-Size	MarketValue	Cont.	
Symbol Color	POS	Disc.	
Border Color	None		
Border Width	None		
Text Label 1	Athlete	Disc.	☐ N:V
Text Label 2	MarketValue	Cont.	☑ N:V
Text Label 3	POS	Disc.	☐ N:V
Text Label 4	None		☐ N:V
Text Label 5	None		☐ N:V
Text Label 6	None		☐ N:V

Name	Type	# Value
3PT	String/Number	4
AST	String/Number	11
Athlete	String	50
BLK	String/Number	5
Change	String/Number	37
Decision	String	5
FG%	String/Number	19
FGM-A	String	50
FT%	String/Number	28
FTM-A	String	50
MarketValue	String/Number	47
Min	String/Number	17
POS	String	5
Points	String/Number	21
Prev	String/Number	47
REB	String/Number	12
STL	String/Number	4
TO	String/Number	5
Team	String	26

Figure 3: Mapping Designer Interface.

The visual workspace (**Figure 2**) is where users interact with the visual objects representing data elements. Position and size are edited through direct manipulation. Application buttons on the VITE toolbar are used for changing the object color, border color and border width. The mapping designer includes a mapping overview (**Figure 3**) and mapping assignment interfaces for each visual property. The mapping overview displays assignments between semantic attributes and visual properties.

The mapping engine contains a graphic parser and a graphic renderer to implement the two-way mapping. It monitors changes in the mapping designer and the activities in the workspace. The graphic parser interprets user input and intentions. The graphic renderer projects the data store into the graphical display according to the mapping assignments defined in the mapping designer. Working together, the graphic parser and graphic renderer synchronize interpreted results in the data storage and the graphical display, and results are reflected on the graphical display immediately.

4. DEVELOPMENT OF VISUAL PROBLEM SOLVING STRATEGY

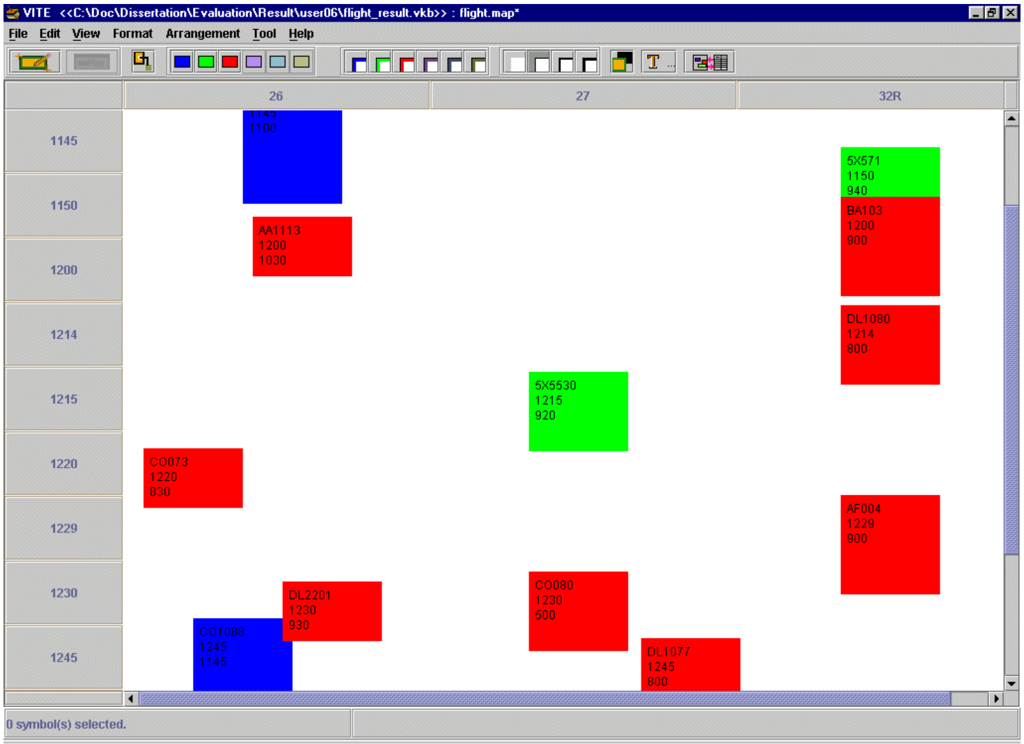
A study of how people use two-way mappings was conducted using VITE [Hsieh and Shipman 2002]. Eleven test subjects were recruited and given a tutorial and training session to explain the concept of two-way mappings and the VITE interface. The tutorial discussed the process of designing a two-way mapping for an example task (a class scheduling task). Each user was given two tasks to perform using VITE and each included the design of a two-way mapping. After completing each task, a questionnaire was used to gather information specific to the task, covering topics such as the design rationale for the visual mappings chosen. After completing both tasks and the task-specific questionnaires, a general questionnaire was used to evaluate the VITE interface and to gather users' general impressions on the use of two-way mappings.

There was no time limit restricting the tasks although subjects generally took about two hours. Subject's understanding of the task domains differed greatly, leading to a variety in the depth of their decision making processes. As there was no correct answer for the given tasks, the open-ended duration allowed users to achieve a satisficing decision [Simon 1996].

The tasks selected required subjects to make decisions based on information provided within VITE. Subjects were told to work until they felt confident about their final decision. Part of their goal, as described in the instructions, was to design mappings that could be used again during similar tasks in the future.

4.1 Observation of Visual Problem Solving

Subjects in the study used a variety of problem solving strategies to make decisions using VITE. Generally, subjects get accustomed to the use of visual tool and develop visual problem solving strategy in two to three phases of usage. In the first phase, subjects designed mappings as if VITE is a simple visualization system. In the second phase, the requirement of manipulating certain



VITE <<C:\Doc\Dissertation\Evaluation\Result\user06\flight_result.vkb>> : flight.map*

0 symbol(s) selected.

Figure 4: A common workspace layout and mapping used for Task 1.

attributes became apparent, and the mappings were changed accordingly. Some users continued to improve their strategy and explore more advanced possibilities of VITE's two-way mappings.

In the first task where subjects were given the role of a flight approach controller; their job was to assign runways for approaching aircraft, ensuring both safety and timeliness. Most subjects (9 of 11) used similar mappings for this task with *Runway Assignmen*t mapped to X position and *Arrival Time* mapped to Y position (**Figure 4**). Other attributes were mapped with greater variety, such as *Aircraft Type* to object color, *Flight Number* to text labels, etc. The idea behind the similarity in the use of X-Y position mapping is apparent - spatial relation reveals possible conflict clearly. Subjects know the concept and can apply it in VITE without difficulty. The variety in mapping none-critical attributes is expected because subjects use them as optional information when making the decision.

The second task put the subjects into the role of a manager of a fantasy basketball team. The task was to hire the best team (one player at each position) they could afford given a limited budget. Choosing the best (and most expensive) player in one position meant sacrificing at other positions. A further complication was that different statistics were more important than others for different positions. Subjects were given 50 players to choose from (10 for each position).

Most subjects approached this task with similar concept learned from the earlier task. The typical mapping design for the task was to use the X-position (*Player Position*) to categorize players and the Y-position to represent the "Decision" attribute, so adjusting the decision was just a matter of moving the player's object up or down. **Figure 5** shows one such mapping with Color mapped to "MarketValue". In this mapping setting, the most obvious restriction was applied by mapping it to the most prominent visual properties to create a visual constraint, so the restriction can be revealed or enforced. Here, the restriction *one player at each position* was realized by the described mapping. With this visualization, there should only be one object (one player) in the bottom row (selected player) for each column (for each position) if the restriction is to be met.

4.2 Two-way Mapping Strategies

Among the approaches taken were mapping decisions and uses of two-way mappings that would not be likely using a pure visualization system. These include the removal of object identifiers, the use of unmapped visual properties, and the mapping and later unmapping of a particular visual property.

Unique values and names for each entity exist in structured data but were not always used by the subjects. One example is shown in **Figure 5**. In this case, the subject decided not to display the name of the basketball players. This subject thought that the names would bias his decisions when the task should really just be about the statistics since that is how fantasy basketball is scored. Although this approach may not be favored by most people, the two-way mapping enabled the subject to make decisions without ever needing to know the name/identifier of each player selected.

Use of unmapped visual properties was common in one form or another. This included some subjects preserving one or more of the most effective visual properties to use during decision-making. Some subjects decided not to map one of the position dimensions (or color) to a semantic attribute so they could cluster or categorize the entities based on a criteria not included in the structured information. This use of unmapped visual attributes for temporary or intermediate

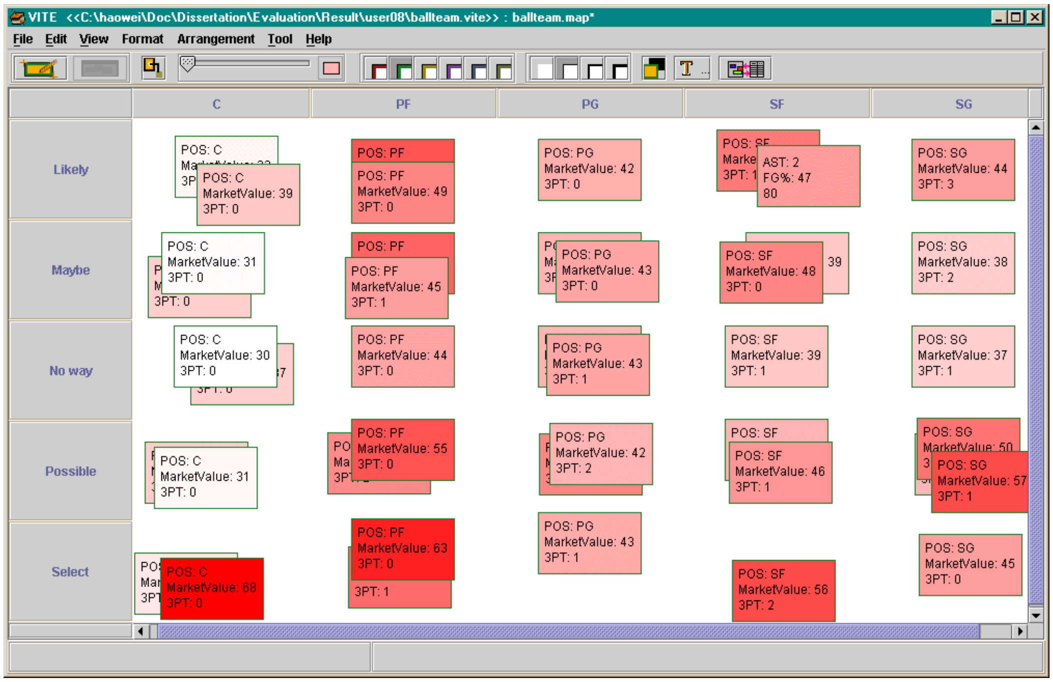


Figure 5: A common workspace layout and mapping used by most users for Task 2.

results was expected. Two users took this strategy a bit further by initially creating a two-way mapping in order to generate the initial layout and then removing mappings for selected visual attributes. Doing this allowed them to manipulate the objects more freely in order to represent partial and alternative solutions without editing the structured data.

Use of multiple visual mappings for task two was already mentioned. This occurred when subjects decided different attributes in the structured data were important for different aspects of their decision. For example, three point shooting and steals are important in selecting the guards for the fantasy basketball team but rebounds and blocks are important for centers and power forwards.

4.3 Advanced Visual Problem Solving

Subjects in the study did not work with the system long enough to explore a wide variety of problem-solving strategies. A few subjects went beyond the basic mapping capabilities of VITE and combined with layout functions (e.g., alignment and distribute) to achieve more complex activities during the task. For example, visual sorting can be achieved by consecutively executing two alignment functions as is described in the following example. This example of visual problem solving was obtained from the subject who used the most visual functions of VITE to complete task 2 in the study. He started the process by initially mapping *players' positions* to X-position in order to spread out the players (similar to the mapping **Figure 5**). He then mapped *player position* to Color and removed the X-position mapping in order to get a freely manipulable workspace. Each player's salary (*MarketValue*) was presented by the object height (Y-Size). Then he created a reference object "BUDGET" (**Figure 6**) (colored white for distinguishing it from other position) with its height representing the allowed budget.

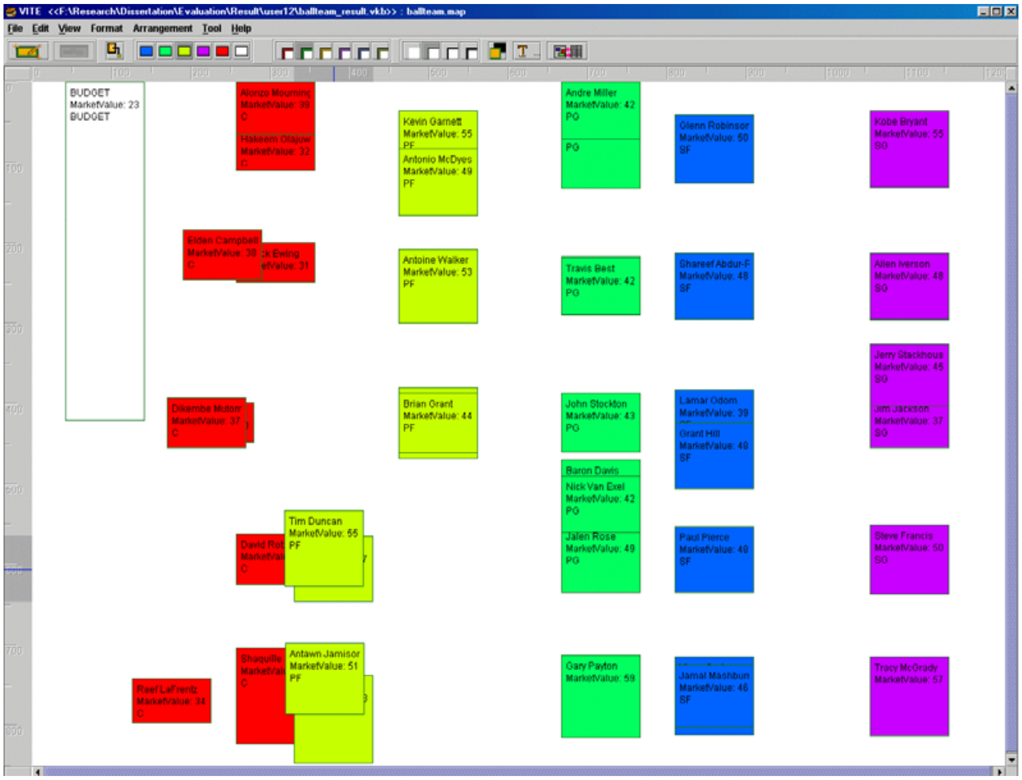
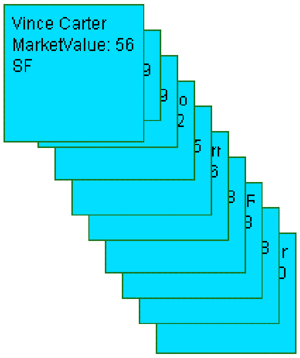


Figure 6: Subject created a reference object (white object on the left)
to compare with his selection.

Next, he started trying to fit players under the salary cap. Soon he found that locating a better fit from the players' pool was a little troublesome, so he experimented with the visual arrangement functions provided by VITE. He first "Stacked" the selection and then aligned them at the top of the selection (**Figure 7**). He repeated the same process for all piles (each pile representing players for the same position) multiple times. With this "stacking" and "aligning" operation, finding a player paid slightly more or less was simply a matter of locating an object that was a little larger or smaller in height. At one point, he even managed to have two possible line ups. At the end, the



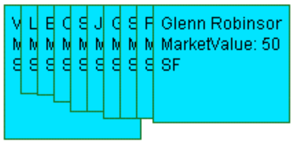


Figure 7: Visual Sorting - steps to form a easy-to-compare
pile by stacking (left) then aligning at top (right).

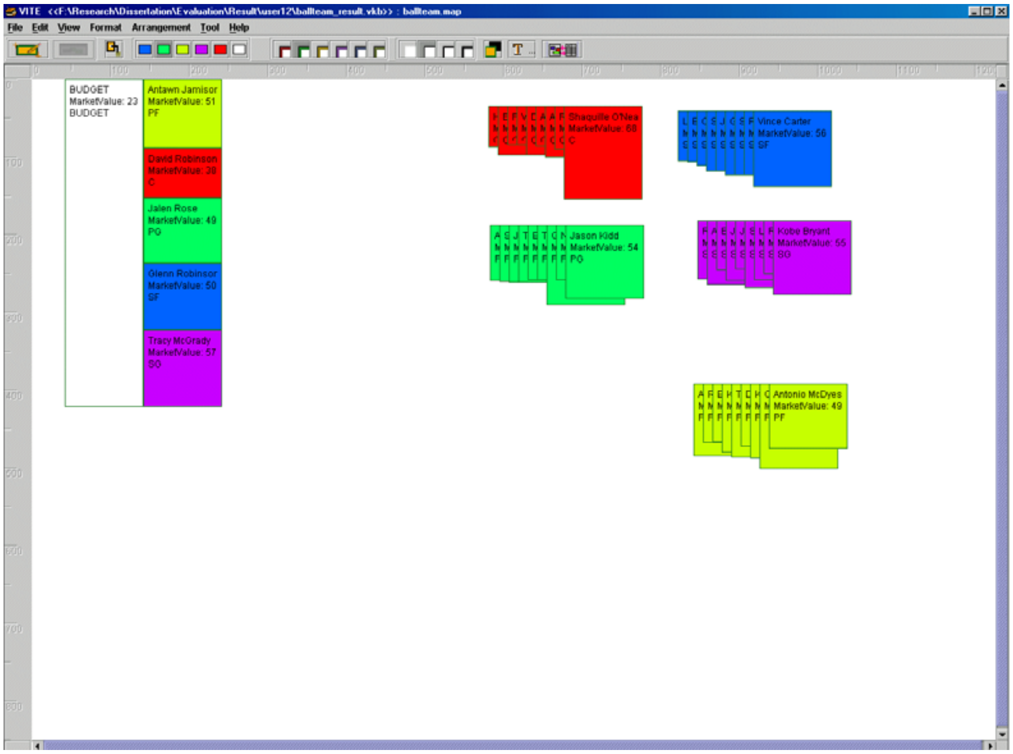


Figure 8: Workspace showing final decision.

total height of selected players could not exceed the budget object (**Figure 8**), and a solution was found in the visual environment. The visual mapping used is shown in **Figure 3**.

A similar approach (**Figure 9**) was performed by another subject. The main difference was the way he obtained each pile of positions. The alternative visual sorting he used was to align each pile on its left edge (Align Left function) with MarketValue being the object width, and then evenly distributed each pile in the Y direction (Distribute Y function).

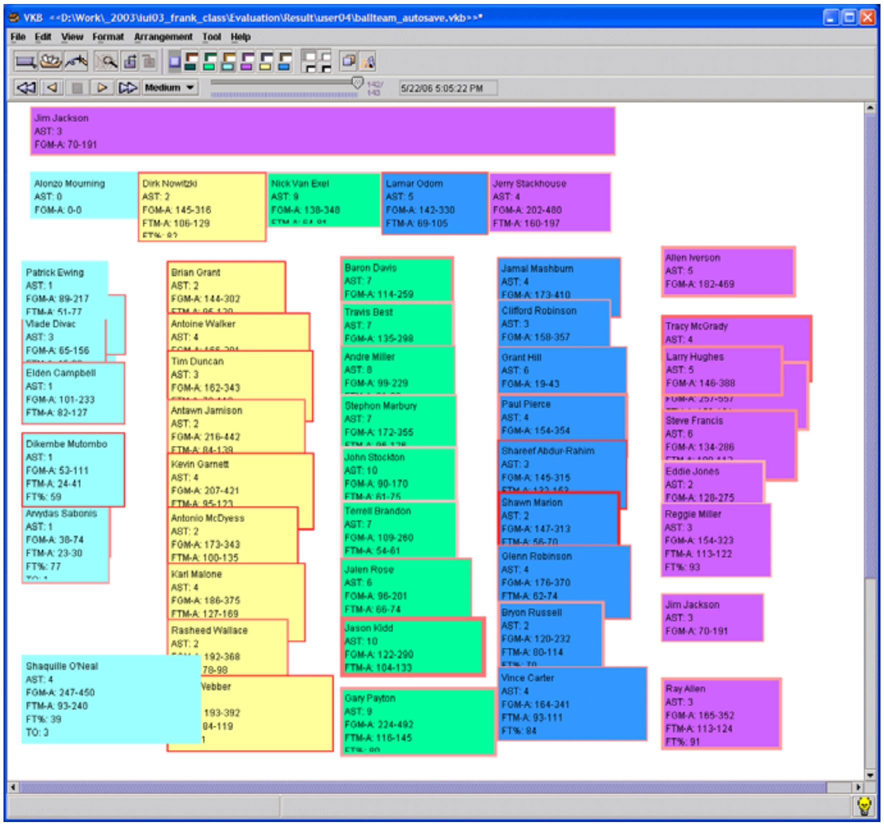
Both of these approaches use a visual function to assist in comparison. The selected players were placed side by side with the reference object (budget). In order to determine if the total salary requirement has exceeded the available budget, an adjoin function was applied on the five objects so they are placed together with zero distance but without overlapping.

5. ISSUES AND IMPLEMENTATION LIMITATIONS

A number of issues arose during the implementation of the VITE system, along with some lessons learned during implementation and evaluation.

5.1 Computationally interpretable edits

The basic requirement for the two-way mapping to work is that the visual information must be decodable, i.e., the visualization that results from user editing must be parseable. This does not mean that all information created in the visual workspace during the performance of the task must be represented in the structured data. A strict one-to-one mapping (one visual value for one data



Jim Jackson
AST: 3
FGM-A: 70-191

Alonzo Mourning
AST: 0
FGM-A: 0-0

Dirk Nowitzki
AST: 2
FGM-A: 145-316
FTM-A: 106-129

Nick Van Exel
AST: 9
FGM-A: 138-348

Lamar Odom
AST: 5
FGM-A: 142-330
FTM-A: 69-105

Jerry Stackhouse
AST: 4
FGM-A: 202-480
FTM-A: 160-197

Patrick Ewing
AST: 1
FGM-A: 89-217
FTM-A: 51-77

Vlade Divac
AST: 3
FGM-A: 65-156

Elden Campbell
AST: 1
FGM-A: 101-233
FTM-A: 82-127

Dikembe Mutombo
AST: 1
FGM-A: 53-111
FTM-A: 24-41
FT%: 59

Arvydas Sabonis
AST: 1
FGM-A: 38-74
FTM-A: 23-30
FT%: 77

Shaquille O'Neal
AST: 4
FGM-A: 247-450
FTM-A: 93-240
FT%: 39
TO: 3

Brian Grant
AST: 2
FGM-A: 144-302

Antoine Walker
AST: 4

Tim Duncan
AST: 3
FGM-A: 162-343

Antawn Jamison
AST: 2
FGM-A: 216-442
FTM-A: 84-139

Kevin Garnett
AST: 4
FGM-A: 207-421
FTM-A: 95-123

Antonio McDyess
AST: 2
FGM-A: 173-343
FTM-A: 100-135

Karl Malone
AST: 4
FGM-A: 186-375
FTM-A: 127-169

Rasheed Wallace
AST: 2
192-368
78-98

Vebber
193-392
84-119
1

Baron Davis
AST: 7
FGM-A: 114-259

Travis Best
AST: 7
FGM-A: 135-298

Andre Miller
AST: 8
FGM-A: 99-229

Stephon Marbury
AST: 7
FGM-A: 172-355

John Stockton
AST: 10
FGM-A: 90-170
FTM-A: 61-75

Terrell Brandon
AST: 7
FGM-A: 109-260
FTM-A: 54-61

Jalen Rose
AST: 6
FGM-A: 96-201
FTM-A: 66-74

Jason Kidd
AST: 10
FGM-A: 122-290
FTM-A: 104-133

Gary Payton
AST: 9
FGM-A: 224-492
FTM-A: 116-145

Jamal Mashburn
AST: 4
FGM-A: 173-410

Clifford Robinson
AST: 3
FGM-A: 158-357

Grant Hill
AST: 6
FGM-A: 19-43

Paul Pierce
AST: 4
FGM-A: 154-354

Shareef Abdur-Rahim
AST: 3
FGM-A: 145-315

Shawn Marion
AST: 2
FGM-A: 147-313
FTM-A: 56-70

Glenn Robinson
AST: 4
FGM-A: 176-370
FTM-A: 62-74

Bryon Russell
AST: 2
FGM-A: 120-232
FTM-A: 80-114

Vince Carter
AST: 4
FGM-A: 164-241
FTM-A: 93-111
FT%: 84

Allen Iverson
AST: 5
FGM-A: 182-469

Tracy McGrady
AST: 4

Larry Hughes
AST: 5
FGM-A: 146-388

Steve Francis
AST: 6
FGM-A: 134-286

Eddie Jones
AST: 2
FGM-A: 120-275

Reggie Miller
AST: 3
FGM-A: 154-323
FTM-A: 113-122
FT%: 93

Jim Jackson
AST: 3
FGM-A: 70-191

Ray Allen
AST: 3
FGM-A: 165-352
FTM-A: 113-124
FT%: 91

Figure 9: Visual problem solving with different layout and an alternative visual sorting method. This visual sorting of a pile (same colored objects in a column) can be obtained by "Align Left", then "Distribute Evenly in Y".

value) is the easiest way to ensure the parse-ability. However, one of the major advantages of working in a visual workspace is the ability to express uncertainty and sub categorizations through slight modifications to prototypical values. By loosening the one-to-one mapping restriction a bit, the workspace can preserve this more subtle visual information. Thus VITE uses ranges of visual values for each attribute value. This consideration is especially important for discrete style mapping. For instance, changing the location of the information object in a strict one-to-one mapping would change the value of the attribute that is mapped to location. With VITE's range of values, the underlying data will only change when the object is moved out of its mapped region.

5.2 Visually representing no values

Objects in the data store do not always have a value for each attribute. Indeed, having N/A (not applicable) as the default value for attributes is common. However, not every visual property can easily represent no value. Continuous mappings are particularly problematic. Since the value-mapping conversion is through a linear transformation, there is no way to define a mapping for no value. Discrete mappings can include an additional visual value, such as a row or column in the case of position, an additional color, size, or border width to represent no value. So far, VITE does not automatically support no value, but users can add a new N/A value for discrete mappings in the mapping assignment interface as they desire.

5.3 Traditional database functionality

VITE provides some basic functionality for managing a structured database. Some database functions may be expected by users, such as adding new data records, and adding new data fields. These operations are translated into adding new objects and adding attributes to objects in a visual workspace. Adding a new object in the visual workspace is performed by creating a new visual symbol. When this happens, VITE creates a new information object in the data storage with a set of default values. Adding new values is quite different than in a traditional database system. A value cannot just be added to an information object, but has to be registered in the mapping profile; otherwise the mapping engine will not know how to visually represent the newly added value. VITE includes this ability in the mapping assignment interface, and it forces the user to add a new value in the mapping assignment before using the new value in an information object. There is a problem with this approach though. If the attribute is mapped to more than one visual property, adding the new value in one visual property does not add the value in the other visual property that use the same attribute. A future version should check the consistency and suggest default values for the user.

6. CONCLUSION AND FUTURE WORK

Structured information has been growing in importance for a long time. We have learned to utilize its strength of abstraction for concept building and learning, and to limit its application so as to avoid problems due to its deconstructive nature. The fact that spatial hypertext can support manipulating information in a less-formal environment gives us a new opportunity to see how such environment can help dealing with structured information. Computer technology has also come a long way from requiring structured information in order to process it, to being able to handle more and more less-formal information and media types and provide support to make use of structured information. These advances give us another chance to evaluate what we can do to more actively support knowledge-intensive work, such as decision-making, rather than to simply provide information.

This research took a simple step to address the problems of interacting with structured information by providing a two-way mapping mechanism in spatial hypertext environment to bind the visual representation of structured information with a direct manipulation workspace. It is hoped that, by bringing the structured information into the visual workspace, the power of human vision and perception, combined with the direct nature of manipulation, can increase the applicability of structured information and better support information-intensive work.

Through the evaluation, how people interact with visual properties and data characteristics was observed. Examples, such as choosing an attribute for the visual property Position, indicate people's instincts generally match the more theoretical rankings of visual properties for graphic design. They determined appropriate attribute selections in a few tries, and managed to explore several uses of a property in a short period of time.

Two-way mappings partially solve some of the problems of interacting with structured information, but there is a lot that remains unanswered. With all the developments in visualization and graphic design principles, we hope others will expand their research related to generating and evaluating information visualization to consider two-way mapping mechanisms. In particular, an in-depth study on the effectiveness of manipulability ranking similar to those of perceptual ranking by [Mackinlay 1986] and [Cleveland and McGill 1984] would be valuable.

In the future, we plan to integrate VITE into VKB, where users have more control over the spatial hypertext workspace, with VITE-like two-way mapping engine to help users work with formal information. One idea for this integration is to give each VKB collection an optional VITE engine. When the VITE engine of a collection is enabled, the collection serves as an automated organizer for objects thrown at it. Objects will be arranged with new position, color, or size based on their attributes and the design of the mapping profile. Users can turn off the VITE engine of a collection when more delicate arrangements are desired.

7. ACKNOWLEDGMENTS

This work was supported in part by grant IIS-9734167 and DUE-0226321 from the National Science Foundation. We appreciate the comments of Cathy Marshall on earlier drafts of this paper.

8. REFERENCES

Card, S. K., Robertson, G. G., and Mackinlay, J. D. (1991) “The Information Visualizer, An Information Workspace”. *Proceedings of CHI 1991 (New Orleans, LA, April 27-May 2)*. 181-188.

Card, S. K., Mackinlay, J. D., and Shneiderman, B. (1999) *Readings in Information Visualization - Using Vision to Think*. Morgan Kaufmann Publishing, Inc. (San Francisco, CA).

Cleveland, W. S. and McGill, R. (1984) “Graphical Perception: Theory, Experimentation and Application to the Development of Graphical Methods”. *Journal Am. Stat. Assoc.* 79, 387 (September). 531-554.

Dourish, P., Edwards, W. K., LaMarca, A., and Salisbury, M. (1999) “Presto: An Experimental Architecture for Fluid Interactive Document Spaces”. *ACM Transactions on Computer-Human Interaction, 6, 2* (June), 133-161.

Hsieh, H. and Shipman, F. M. III (2000) “VITE: A Visual Interface Supporting the Direct Manipulation of Structured Data Using Two-Way Mappings”. *Proceedings of Intelligent User Interface* (IUI) 2000. 141-148.

Hsieh, H. and Shipman, F. M. III (2002) “Manipulating Structured Information in a Visual Workspace”. *Proceedings of User Interface Software and Technology* (UIST) 2002. 217-226.

Hutchins, E., Hollan, J. D., and Norman, D. A. (1986) Direct Manipulation Interfaces. In *User-Centered System Design*, D. A. Norman & S. W. Draper, Eds. Lawrence Erlbaum Associates, (Hillsdale, NJ) 87-124.

Kumar, V., Furuta, R., and Allen, R. B. (1998) Metadata Visualization for Digital Libraries: Interactive Timeline Editing and Review. *Proceedings of Digital Libraries 1998* (DL 1998, Pittsburgh, PA, June 24-27). 126-133.

Mackinlay, J. D. (1986) “Automating the Design of Graphical Presentations of Relational Information”. *ACM Transactions on Graphics*, 5, 2 (April), 110-141.

Mackinlay, J. D., Robertson, G. G., and Card, S. K. (1991) “The Perspective Wall: Detail and Context Smoothly Integrated”. *Proceedings of CHI 1991* (New Orleans, LA, April 27-May 2). 173-179.

Malone, T. W., Grant, K. R., Lai, K., Rao, R., and Rosenblitt, D. (1986) “Semi-structured Messages Are Surprisingly Useful for Computer-Supported Coordination”. *ACM Transactions on Office Information System*, 5, 2 (April 1987), 115-131.

Marshall, C. C., Halasz, F. G., Rogers, R. A., and Janssen, W. C. Jr. (1991) “Aquanet: A Hypertext Tool to Hold Your Knowledge in Place”. *Proceedings of Hypertext 1991*. 261-275.

Marshall, C. C. and Shipman, F. M. III (1997) “Spatial Hypertext and the Practice of Information Triage”. *Proceedings of Hypertext 1997*. 124-133.

Moran, T. P., Chiu, P., and van Melle, W. (1997) “Pen-based Interaction Techniques for Organizing Material on an Electronic Whiteboard”. *Proceedings of UIST 1997*. 45-54.

Norman, D. A. (1988) *The Design of Everyday Things*. Currency and Doubleday, 1990, (New York, NY). Originally published as *The Psychology of Everyday Things*. Basic Books, (New York, NY), 1988.

Robertson, G. G., Mackinlay, J. D., and Card, S. K. (1991) “Cone Trees: Animated 3D Visualizations of Hierarchical Information”. *Proceedings of CHI 1991* (New Orleans, LA, April 27-May 2). 189-194.

Robertson, G. G., Card, S. K., and Mackinlay, J. D. (1993) “Information Visualization Using 3D Interactive Animation”. *Communications of the ACM, 36, 4* (April), 57-71.

Robertson, G. G., Czerwinski, M., Larson, K., Robbins, D. C., Thiel, D., and Dantzich, M. V. (1998) “Data Mountain: Using Spatial Memory for Document Management”. *Proceedings of UIST 1998,* (San Francisco, CA, November 1-4). 153-162.

Sarkar, M. and Brown, M. H. (1992) “Graphical Fisheye Views of Graphs”. *Proceedings of CHI 1992* (Monterey, CA, May 3-7). 83-91.

Schön, D. A. (1983) *The Reflective Practitioner – How Professionals Think in Action*. Basic Books, (New York, NY).

Shipman, F. M. III (1993) Supporting Knowledge-Base Evolution with Incremental Formalization. Ph.D. Dissertation, Department of Computer Science, University of Colorado, Boulder, CO.

Shipman, F. M. III, and Marshall, C. C. (1999) “Formality Considered Harmful: Experiences, Emerging Themes, and Directions on the Use of Formal Representations in Interactive Systems”. *Computer-Supported Cooperative Work*, 8, 4, 333-352.

Shipman, F. M. III and McCall, R. J. (1999) “Incremental Formalization with the Hyper-Object Substrate”. *ACM Transactions on Information Systems*, 17, 2, 199-227.

Shipman, F. M. III and Hsieh, H. (2000) “Navigable History: A Reader's View of Writer's Time”. *New Review of Hypermedia and Multimedia*, 6, 147-167.

Simon, H. A. (1996) *The Sciences of the Artificial*. The MIT Press, (Cambridge, MA).

Suchman, L. A. (1987) *Plans and Situated Actions: The Problem of Human Machine Communication*. Cambridge University Press, (New York, NY).

Tufte, E. R. (1990). *Envision Information*. Graphics Press, (Cheshire, CT).

van Mechelen, I., Hampton, J., Michalski, R. S., and Theuns, P. Eds. (1993) *Category and Concepts — Theoretical Views and Inductive Data Analysis*. Academic Press, (San Diego, CA).

Wærn, A. (1997) “Local Plan Recognition in Direct Manipulation Interfaces”. *Proceedings of Intelligent User Interfaces 1997* (IUI 1997, Orlando, FL, January 6-9). 7-14.