Manipulating Structured Information in a Visual Workspace

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
TEL: 1-979-862-3217
{haowei,shipman}@csdl.tamu.edu

ABSTRACT
This paper describes the VITE system, a visual workspace 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. This is related to information visualization, but reflecting visual edits in the structured data requires a two-way mapping from data to visualization and from visualization to data. VITE provides users with an interface for designing two-way mappings. Mappings are reusable on different datasets and may be switched within a task. An evaluation of VITE was conducted to study how people use two-way mapping and how two-way mapping can help in problem solving tasks. The results show that users could quickly design visual mappings to help their problem-solving tasks. Users developed more sophisticated strategies for visual problem-solving over time.

Keywords
Information visualization, information workspace, two-way mappings, editable visualizations

INTRODUCTION
Much of the information generated and manipulated today is structured and represented formally in order to let computers process it. In contrast to the way computers process information, people often prefer less-formal representations to perceive things as a whole and to delay the abstraction process [25]. Instead of interpreting the decomposed information, people often look at the information from a higher level.

In this discussion, formalized or structured information is that which is represented with a pre-defined set of attributes, values, and relations. Examples include frames, objects in a knowledge base, and data records in a relational database, each of which is composed of a set of data fields.

For people to make use of this structured information, 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 interface used rarely support the representation and manipulation of less structured information related to the formally-represented content.

The next section describes problems that arise when working with structured information. It then presents an approach to manipulating structured information via a visual workspace enabled by a two-way mapping between data and visualization. This approach is instantiated in VITE, a system that supports the design and use of two-way mappings. A study of problem solving using VITE is then described, and the paper concludes with limitations of the current implementation, on-going work, and conclusions.

WORKING WITH STRUCTURED INFORMATION
There are two reasons why working with structured information does not fit into common problem-solving practice. First of all, the formalization process decomposes 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 "non-destructive formalization" during knowledge building that keeps the original "less formal representation as well as the formal representations" [24]. Secondly, in task-based decision making processes, 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.

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. For example, in a digital library, each book may be represented as an information object. These objects usually include 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. These attributes could be added to the schema, but there are always more that could be important to the library patron.

The inevitably incomplete nature [31] of the representation does not imply that 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.

Intermediate Representations for Problem Solving

Problem solving usually involves information gathering, categorizing, and knowledge building [33]. 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 predefined in a schema [11].

VISUALIZING AND MANIPULATING STRUCTURED INFORMATION

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. Database systems normally provide either table- or form-based interfaces to facilitate data presentation and data editing. For many tasks, a better way to present information is through information visualization [2]. Visualization techniques use both retinal properties and spatial arrangement for the presentation of structured information, taking advantage of the human perceptual system.

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 [7] by providing temporary external memory [15] and revealing spatial relations. In the process of presentation, some of the knowledge lost in the abstraction process may also become apparent. A 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 [1], 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 [1] 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 [19] or view transformations, such as a perspective wall [10], cone trees [18] and fisheye views [22]. Kumar's Timeline project [8] provides an example of the assignment of visual mappings for specific tasks.

A related area of research aims at providing multiple concurrent visualizations of a data set. FOCUS is an interactive data table viewer used for comparing multiple attributes in a table using fisheye techniques [30]. The adjustable focus+context for data exploration shares some concepts with the presented approach in its support for task or user-dependent visualizations of information. Table Lens is also an interactive table-viewing environment and uses a focus+context technique for exploring large data sets [16]. It adds a graphical mapping scheme into table cells in order to show cell values and patterns. SSR, SIV [3] and FINESSE [34] use a spreadsheet environment to view and change multiple visualizations at the same time.

Another area of research, exemplified by Visage [21] and GlyphMaker [17], aims at providing configurable visualizations of data. These systems allow unidirectional visual mappings to be modified using schema drawing and diagram manipulation.

Manipulating Spatialized 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 [32]. 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 the real decision making process.

Visual workspaces containing directly manipulable information objects take advantage of human spatial cognition and immediate feedback from the visual environment [35]. 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 [5] and Data Mountain [20] use direct manipulable visual workspace for exploring a natural form of interaction, but focus on providing collaborative support.

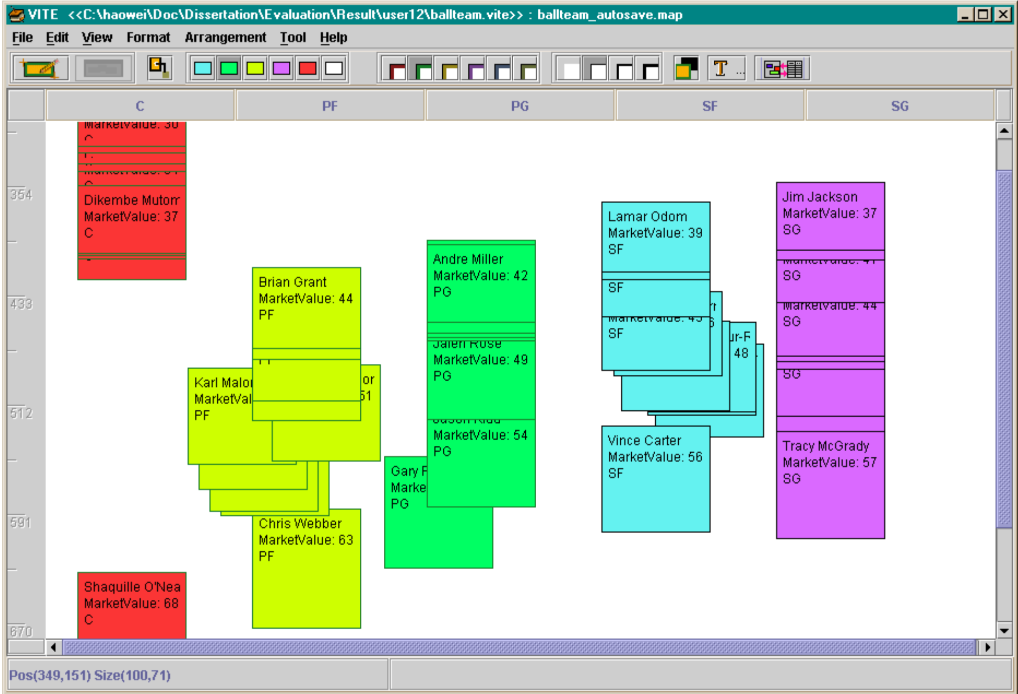

Figure 1 : Workspace containing visual objects representing basketball players being organized during selection of a fantasy basketball team.

Two-Way Visual Mappings

As described, visualization systems often let the user change the way the system presents information but not the information itself. 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, and the editing of the structured information within the visualization.

Visual workspaces that combine 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 users' changes to the information. The approach presented here is to take the unidirectional mapping of visualization systems, and make it bidirectional. Two-way mappings specify an initial visualization of the structured information and then parse any visual-property changes made by the user and reflect these changes back into the underlying data store. The need for user edits to be automatically interpretable constrains the visual mappings when compared to unidirectional visualization.

Configurable visual mappings not only extend the interaction between the system and users, but also allow users to create or modify visual mappings for different tasks. Mappings might be saved for reuse with another data set that has a similar set of attributes. In addition to reusing visual mappings, switching between visual mappings enables the same data set to have multiple visualizations. Visualizations tailored to different tasks can be applied to the same data.

VITE: A VISUAL WORKSPACE SUPPORTING TWO-WAY MAPPINGS

We have developed VITE to explore the design and use of systems incorporating two-way mappings. An early version of VITE was reported in [6]. VITE's design and development have been influenced by a variety of prior work in the area of hypertext and visualization. Aquanet [12] and VIKI [13] used spatial layout for arranging relationships among information objects. HOS [26] and Aquanet added an abstract layer by allowing attributes, relationships and types to be associated with information objects. Tivoli [14] incorporated specific domain knowledge into the system to support implicit perception in a free-form interaction environment. VKB [28] uses both spatial layout and abstract attributes to support incremental formalization within user-created information workspaces.

VITE expands on the use of spatial layout for arranging information. In VITE, the attribute/value pairs are the primary content rather than metadata attached to a larger plain text or image information object. VITE is designed to use 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 (including parsing edits), and the data store.

Visual Workspace

The two-dimensional workspace (Figure 1) is where users interact with the visual objects representing data elements.

A visual object is composed of the following graphical properties: position, size, background color, border color, border width, and text labels. 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. A variety of menu driven functions are also provided to enhance efficiency and usability including align, distribute, or resize all selected objects.

The two-way graphical mapping for moving information to and from the database is called a "perspective". Attributes of data are encoded as different graphical properties such as position, size, color, etc. Once the mapping is defined, information is projected onto the visual workspace and presented as information objects. Modifying information is by direct manipulation such as dragging the objects or changing other graphical properties using menu and applicator buttons. Interactions between the user and the interface are immediately interpreted by the system and the results are reflected in both the visual representation and the structured information stored in the database.

VITE supports discrete and continuous mappings but does not support mapping nominal (general text) attributes because such content can already be edited in a text editor. Discrete-valued attributes have a limited set of possible values, such as categories, names, etc. A discrete mapping matches each possible attribute value to a visual value or a range of visual values for a visual property. Continuous mappings are for numeric attributes. VITE scans the data range, the visual property range, and then calculates a linear transformation for the continuous mapping. For example, a given data attribute with values ranging from 1 to 20, when mapped to visual property "Width", will have an estimated visual property range of 50 to 140 (a default width range for information objects in VITE that is modifiable). This data range (1,20) will be mapped to the visual range (50,140) using the calculated linear transformation below.

$$\text{Size} = 50 + (\text{Value} - 1) \times \frac{(140 - 50)}{(20 - 1)}$$

VITE also allows users to manipulate some non-mappable visual properties such as object transparency, font and text color. These properties can be used for temporary external memory during the decision-making process. Mappable visual attributes such as position and color may be used in a similar fashion by removing the mapping definition.

Mapping Designer

The mapping designer includes a mapping overview (Figure 2) and mapping assignment interfaces (Figure 3) for each visual property. The mapping overview displays assignments between semantic attributes and visual properties on the left and a list of all attributes, their types, and the number of distinct values on the right. The mapping assignment interfaces are used to define detailed mapping pairs for each visual property. VITE automatically initializes the detailed mapping pairs with heuristically-determined defaults. For example, when a discrete attribute is selected to be used for visual property "symbol color", a set of distinct colors will be generated by default (Figure 3 bottom left) so users do not need to manually assign them.

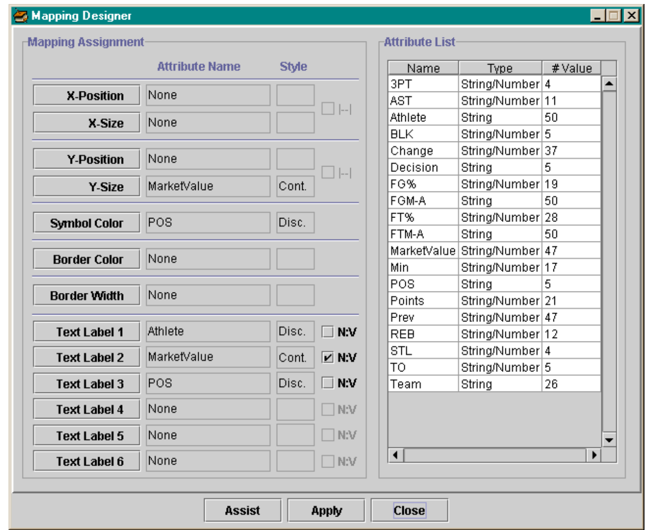

Figure 2 : Mapping Designer Interface.

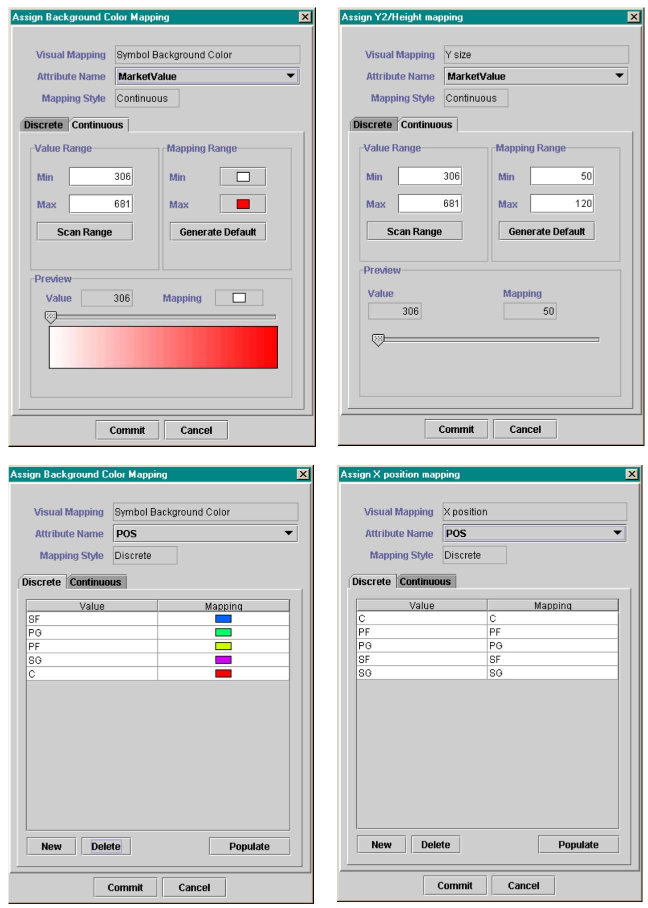

Figure 3 : Continuous and discrete mapping assignment interfaces for Color and Position/Size mappings.

These assignments may then be manually edited. If a continuous attribute is mapped to "symbol color", a linear gradient is generated (Figure 3 top left). Defaults for discrete and continuous values for position or size are generated using similar heuristics (Figure 3 bottom right and Figure 3 top right).

The mapping profile keeps a list of mapping details for each visual property included in the mapping. Mapping profiles can be saved and loaded separately from a particular data set. By storing a mapping profile separately, mappings can be used repeatedly for similar data schemes. For example, consider a mapping designed for analyzing a monthly revenue. By saving the mapping profile, the user does not need to design or generate the mapping profile each month. Instead, the user can import the new data and simply load the existing mapping profile to start the analysis. A data set can also have several mapping profiles (or perspectives) for different purposes. In the example, monthly revenue might be used by both accountants and managers, but each group only looks at certain aspects of the data. By switching perspectives, users can process only the necessary information and tailor the visual workspace to their current tasks.

Mapping Engine

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 onto 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 store and the graphical display – changes are reflected on the graphical display immediately.

Synchronization is necessary since an attribute may be mapped to more than one visual property. An example from Figures 1 and 2 is that "*POS*" (Player position) is mapped to both Symbol Color and Text Label 3. Thus, changing the color of an object would cause the text to also change.

The graphic parser works by monitoring workspace edits, and based on the type of edits, triggers proper look-up or conversion. For example, moving an object will trigger the mapping engine to check for X/Y-Position change. An object's position is determined by its center-point. Similarly, resizing an object triggers checking for X/Y-Size change. Other visual property changes are triggered by button events. When visual changes are triggered, the mapping engine first checks if the edit results in any change to an attribute's value. If so, it then propagates the change to the corresponding structured information as well as other visual properties mapped to the same data attribute.

USER STUDY

A study of how people use two-way mappings was conducted using VITE. Eleven computer science graduate students were recruited as test subjects. Subjects began with a tutorial 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). The tutorial also touched on the rationale behind the interface and explained in what circumstances some types of mapping are used. Each user was given two tasks to perform using VITE, including the design of two-way mappings. 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. Subjects' understanding of the task domains differed greatly, leading to 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 [29].

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. The first task, scheduling aircraft for landing runways, although more complex than the tutorial example, gave subjects more time to familiarize themselves with VITE. 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).

In addition to the questionnaires, subjects were asked to save their visual mapping profile and their final workspace. During the course of the task, VITE also saved changes to the visual workspace as a sequence of history events that can be replayed [27]. Subjects were specifically told to save each mapping profile separately if they designed more than one visual mapping.

Results and Implications

Subjects were given the role of a flight approach controller in the first task; 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* mapped to X or Y position and *ArrivalTime* mapped to the other dimension. Subjects mapped a variety of other attributes (*Status*, *TypeOfAircraft*, etc.) to color, border width, size, etc. *FlightNumber* and *ArrivalTime* were mapped to text labels by more than half of the users. The duplicated presentations of *ArrivalTime* through position and text let users perceive gaps between planes easily (via position) and confirm the accuracy of the information through the text label.

Task 2 was more demanding for the users because the data set was larger, the information included was more complex, and the constraint (the salary cap) was clear and stringent. Achieving a good result took much more effort, as compared to task 1. As a result there were a greater variety of mappings and solution strategies generated. Table 1 shows a partial list of the data attributes (18 attributes in total) provided.

Table 1 : Attribute List for Task 2 (partial).

Attribute	Description
Athlete	The name of the player
POS	Position the player plays
Decision	Decision to select, reject, or still pending
3PT	3 point shots made
AST	Assists
REB	Rebounds
STL	Steals
FT%	Free Throw Percentage
FG%	Field Goal Percentage
Min	Minimum field goal
MarketValue	Current salary to recruit the player
Prev	Salary of the player's previous contract

Table 2 : Task 2 Attribute Selection and Frequency Count

Position(13*2)		Size(13*2)		Color(13*2)		Label(13*6)	
POS	10	MarketValue	4	Decision	6	Athlete	11
MarketValue	5	Points	2	POS	2	MarketValue	10
Decision	3	FG%	2	MarketValue	2	FT%	4
FG%	2	3PT	1	REB	1	AST	4
Points	2	FT%	1	FG%	1	POS	3
		Change	1	FTM-A	1	Prev	2
		BLK	1			3PT	2
		AST	1			FG%	1
						REB	1
						STL	1
						BLK	1
						FGM-A	1
						FTM-A	1
						TO	1
						Team	1
						Decision	1

In task 2, two users insisted that they had to use at least two mapping profiles to complete the task. They did not simply modify their mappings but switched back and forth between different mapping profiles in order to apply the criteria and rules for the task. Because of this, although the data in Table 2 and Table 3 represent the results of 11 test subjects, they are actually a summary of 13 mapping profiles.

Table 2 shows the frequency of mapping particular semantic attributes to classes of visual properties. For example, player position ("POS") was mapped to either X or Y position in the display in ten of the thirteen mappings. Since not all visual characteristics were used in each mapping the sum of the use of X or Y position (22) does not equal the maximum of 26 (2*13). Four mappings left either X or Y position unmapped so objects could be moved without editing the underlying data.

Subjects were asked to provide one or more reasons for their selection of visual properties for information attributes. Subjects were also asked to prioritize the importance of their reasons if more than one was selected.

Table 3 shows the results with the number of times a reason was identified as the most important listed in "1st" columns. Regarding the results of task 1, subjects used position mappings (both X and Y) primarily for their deciding factor and for perceiving the distribution of flights. Size (width and height) and color (background and border) mappings were chosen to perceive values and distributions. Text labels were mainly used for viewing values. In task 2, results were similar but with subjects using position to view values more than distribution. Also, position was used less to indicate a decision. Instead, two subjects chose color mappings to represent their deciding factors (Table 3). Other statistics remain similar, with sizes and labels used mainly for viewing values.

General Questionnaire

The first set of questions in the general questionnaire evaluate VITE's ease of use, impact on performance, and subject satisfaction. Users were asked to rate the system with a score ranging from 1 to 9, where 1 meant they strongly disagreed and 9 meant they strongly agreed. The questions were stated "Given a scale of 1 to 9, how would you rate VITE in supporting the given tasks in the following criteria? Intuitive interface (easy to learn, easy to use)? Efficient (get results faster for the given task, compared with more traditional table form/text-based attribute editor interface)? Performance (get better result)? and Overall rating?"

The quantitative results show that users agreed most on the interface's improvement of efficiency (resulting in a higher mean value and lower standard deviation — see Table 4). Improvements in performance and intuitiveness of VITE's interface, although not much worse than that of efficiency, did not impress users as much.

Nine out of the eleven users thought that VITE provided sufficient visual mappings for the given tasks. The two

Table 4 : Quantitative Evaluation of VITE (1-9, 9:best)

	Intuitiveness	Efficiency	Performance	Overall
Mean $\bar{n}$	7.36	8.09	7.64	7.82
Std Deviation δ_n	0.88	0.51	1.07	0.72

Table 3 Reasons for Selecting Certain Visual Properties for Task 2

	Position		Size		Color		Label	
	all	1st	all	1st	all	1st	all	1st
(a) anticipation of constant/rapid modifications	1	1	0	0	0	0	0	0
(b) better perception of value	13	11	8	8	8	8	40	40
(c) better perception of variety/distribution	3	2	1	1	4	3	3	3
(d) it is the most important /deciding factor or the final result you were adjusting,	10	7	0	0	2	2	9	1
(e) other	0	0	2	2	0	0	3	3

users who wanted more visual properties used more visual properties than the averages of 7.64 for task 1 and 8.15 for task 2. One subject used all 12 visual properties in task 2, and the other subject used two sets of mapping profiles (each containing 9 visual properties) to complete the task.

Subjects were also asked for extensions they would have liked. The most popular suggestion for extra visual properties was a 3D display (Z-axis properties such as Z-position and Z-size). Other suggested properties included shape, more text labels and audio feedback.

Things users did not like about VITE were: (a) its lack of a read-only mapping that prevents accidental editing of values; (b) the SDI (Single Document Interface) design of VITE does not allow comparing different layouts simultaneously; (c) a lack of sub-scale or split scale to better utilize the space and visual properties, and (d) need to learn the concepts of two-way mappings to take advantage of the powerful visual encoding.

There seems to be a contradiction in that users thought VITE was easy to use and learn (from the quantitative evaluation), but was not intuitive to learn (from the question: the things you liked least about VITE). After careful examination, the subject who complained about VITE not being intuitive to learn was one of the subjects who developed advanced problem solving skills like those described in the next section. The subject was also the one who used the most visual functions. This contradiction may show that using the VITE interface is fairly natural for simple tasks, but takes some time to learn when tasks become complicated. Designing an efficient overall strategy for decision making that consists of one or more properly designed mapping profiles given a task with many important attributes can become like a puzzle. One of the difficulties for subjects was to understand the possibilities of visual expression.

Observation of Visual Problem Solving Strategies

Subjects in the study used a variety of problem solving strategies to make decisions using VITE. 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 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 4. In this case, the subject decided not to display the name of the basketball players during task two. 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 results was expected. Two users took this strategy a bit further by initially creating a perspective that mapped structured data to visual attributes in order to generate the initial layout and then removed mappings for selected visual attributes. Doing this allowed them to similarly manipulate the objects to represent partial and alternative solutions and intermediate results.

Use of multiple visual mappings for task two was already

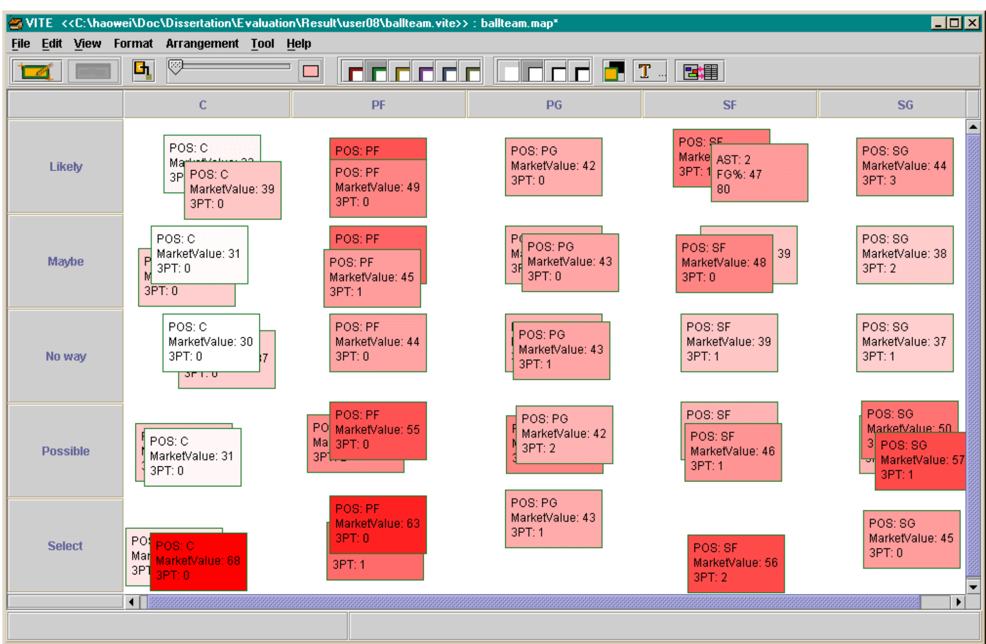

Figure 4 : Most users chose a much simpler visual mapping to complete the task.

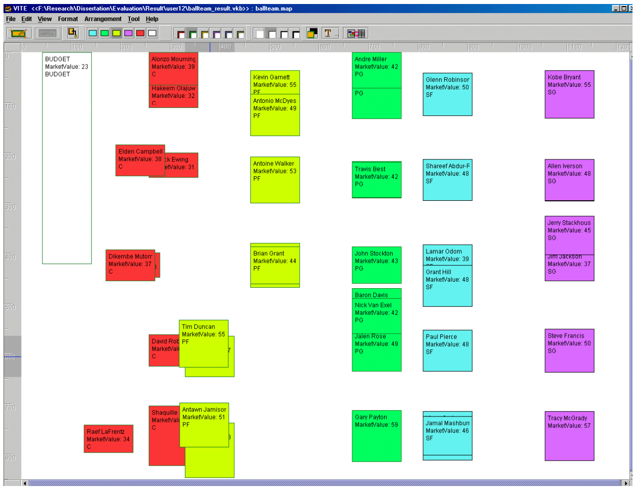

Figure 5 : Subject created a reference object (white object on the left) to compare with his selection.

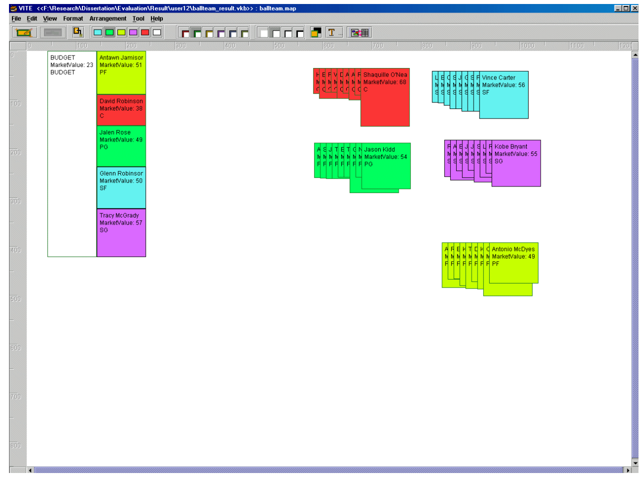

Figure 6 : Workspace showing final decision

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. An example of one particular subject's decision process follows.

The following example of visual problem solving was obtained from the subject who used the most visual functions of VITE to complete task 2. He started the process by initially mapping players' positions to X-position in order to spread out the players. 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) (Figure 1). Then he created a reference object "BUDGET" and a dummy player position "BUDGET" to categorize the reference object (Figure 5) (colored white for BUDGET position). 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. 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 total height of selected players just could not exceed the budget object (Figure 6), and a solution was found in the visual environment. The visual mapping used is shown in Figure 2.

A similar approach was performed by another subject. The main difference was the way he obtained each pile of positions. He aligned each pile on its left edge with MarketValue being the object width, and then evenly distributed each pile in the Y direction. The typical mapping design for the task by other users was to use the X-position (*POS*) 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 4 shows one such mapping with Color mapped to "MarketValue".

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.

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 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 subcategorizations 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.

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.

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 object. 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.

ON GOING PROGRESS

The evaluation of VITE indicated that the visualization and direct manipulation of structured data can be used to improve a variety of tasks. While the results showed subjects found the tool made their task more efficient, it also showed that there are issues surrounding ease of use in the generation of good mappings. The design of Mapping Assistant is an on-going effort to help address these issues.

The main goal of the Mapping Assistant is to help the user generate initial mappings quickly. Users in the study spent much of their time designing the initial mapping, with later adjustments and modifications taking less time. This is natural since the generation of an initial design is a cognitively more difficult task than reacting to an existing design [23]. For this reason, the focus of the work on the Mapping Assistant is to create algorithms and an interface for quickly generating an initial mapping.

The generated mapping will be based on a brief description of the user's task, a statistical analysis of the data set, and a combination of results from the VITE evaluation with existing graphic design principles.

CONCLUSION AND FUTURE WORK

The quantity of structured information continually grows as databases are created and made available via the Internet or Intranets. The structure enables searching and sorting, clearly important given the quantity of information, but interfaces to this structured information do not provide for easy manipulation of the content. The structuring process requires the selection of attributes to be represented, leaving out aspects important to some users. Additionally, interfaces and representations rarely support intermediate formal or informal representations that may be of use during the performance of particular tasks but not stored in the database.

This research took a simple step to address the problems of interacting with structured information by providing a two-way mapping mechanism 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.

VITE was implemented by integrating an information workspace that supports direct manipulation with an enhanced visualization that supports two-way mappings. This tool was used to study the effectiveness of visual properties in two-way mapping environments and to observe the design process of two-way mappings.

The evaluation of VITE indicated that the direct manipulation of structured information in the visualization improved efficiency in various decision-making tasks. Although no direct comparison was made with other tools, users felt confident that they obtained better results using VITE. Users took advantage of control over the visualization by using multiple visualizations and by mapping then unmapping particular attributes in order to work with information in a less constrained setting.

Through the evaluation, the way 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. Making mappings easy to try-out and refine provides users an environment for experimenting with their design instincts.

Two-way mappings partially solve some of the problems of interacting with structured information, but there are many questions that remains unanswered. Within the context of developments in visualization and graphic design principles, we hope others will expand their research related to generating and evaluating information visualizations 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 [9] and [4] would be valuable.

ACKNOWLEDGMENTS

This work was supported in part by grant IIS-9734167 from the National Science Foundation.

REFERENCES

1. Card, S. K., Robertson, G. G., and Mackinlay, J. D. 1991. The Information Visualizer, An Information Workspace. *CHI 1991*. 181-188.

2. 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.
3. Chi, E. H., Barry, P., Reidl, J., and Konstan, J. 1997. A Spreadsheet Approach to Information Visualization. *Information Visualization 1997*. 17-24.
4. Cleveland, W. S. and McGill, R. 1984. Graphical Perception: Theory, Experimentation and Application to the Development of Graphical Methods. In *Journal Am. Stat. Assoc. 79, 387* (September). 531-554.
5. 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*, 133–161.
6. Hsieh, H. and Shipman, F. M. III 2000. VITE: A Visual Interface Supporting the Direct Manipulation of Structured Data Using Two-Way Mappings. *IUI 2000*. 141-148.
7. 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.
8. Kumar, V., Furuta, R., and Allen, R. B. 1998. Metadata Visualization for Digital Libraries: Interactive Timeline Editing and Review. *Digital Libraries 1998*. 126-133.
9. Mackinlay, J. D. 1986. Automating the Design of Graphical Presentations of Relational Information. *ACM Transactions on Graphics, 5, 2* (April), 110-141.
10. Mackinlay, J. D., Robertson, G. G., and Card, S. K. 1991. The Perspective Wall: Detail and Context Smoothly Integrated. *CHI 1991*. 173-179.
11. 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.
12. 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. *Hypertext 1991*. 261-275.
13. Marshall, C. C. and Shipman, F. M. III 1997. Spatial Hypertext and the Practice of Information Triage. *Hypertext 1997*. 124-133.
14. Moran, T. P., Chiu, P., and van Melle, W. 1997. Pen-based Interaction Techniques for Organizing Material on an Electronic Whiteboard. *UIST 1997*. 45-54.
15. Norman, D. A. 1988. *The Design of Everyday Things*. Currency and Doubleday, 1990, New York, NY.
16. Rao, R. and Card, S. K. 1994. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information. *CHI 1994*. 318-322.
17. Ribarsky, M. W., Tumblin, J., and Vetter, J. 1994. Glyphmaker: An Interactive, Programmerless Approach for Custom Visualization and Analysis of Data. Georgia Tech Tech Report. URL: http://www.cc.gatech.edu/scivis/research/glyph/glyph.html.
18. Robertson, G. G., Mackinlay, J. D., and Card, S. K. 1991. Cone Trees: Animated 3D Visualizations of Hierarchical Information. *CHI 1991*. 189-194.
19. Robertson, G. G., Card, S. K., and Mackinlay, J. D. 1993. Information Visualization Using 3D Interactive Animation. *Communications of the ACM, 36, 4,* 57-71.
20. 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. *UIST 1998*. 153-162.
21. Roth, S. F., Lucas, P., Senn, J. A., Gomberg, C. C., Burks, M. B., Stroffolino, P. J., Kolojejchick, J. A., and Dunmire, C. 1996. Visage: A User Interface Environment for Exploring Information. *Information Visualization 1996*. 3-12.
22. Sarkar, M. and Brown, M. H. 1992. Graphical Fisheye Views of Graphs. *CHI 1992*. 83-91.
23. Schön, D. A. 1983. *The Reflective Practitioner – How Professionals Think in Action*. Basic Books, New York, NY.
24. Shipman, F. M. III 1993. Supporting Knowledge-Base Evolution with Incremental Formalization. *Ph.D. Dissertation*, Department of Computer Science, University of Colorado, Boulder, CO.
25. 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.
26. 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.
27. 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.
28. Shipman, F. M. III, Hsieh, H., Maloor, P., and Moore, J. M. 2001. The Visual Knowledge Builder: A Second Generation Spatial Hypertext. *Hypertext 2001*. 113-122.
29. Simon, H. A. 1996. *The Sciences of the Artificial*. The MIT Press, Cambridge, MA.
30. Spenke, M., Beilken, C., and Berlage, T. 1996. FOCUS: The Interactive Table for Product Comparison and Selection. *UIST 1996*. 41-50.
31. Suchman, L. A. 1987. *Plans and Situated Actions: The Problem of Human Machine Communication*. Cambridge University Press, New York, NY.
32. Tufte, E. R. 1990. *Envision Information*. Graphics Press, Cheshire, CT.
33. 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.
34. Varshney, A. and Kaufman, A. 1996. FINESSE: A Financial Information Spreadsheet. *Information Visualization 1996*. 70-71.
35. Wærn, A. 1997. Local Plan Recognition in Direct Manipulation Interfaces. *Intelligent User Interfaces 1997*. 7-14.