VITE: A Visual Interface Supporting the Direct Manipulation of Structured Data Using Two-Way Mappings

Hao-wei 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: 1-409-862-3216
E-mail: {haowei, shipman}@cs.tamu.edu

ABSTRACT
Information processed by computers is frequently stored and organized for the computer's, rather than for the user's, convenience. For example, information stored in a database is normalized and indexed so computers can efficiently access, process, and retrieve it. However, it is not natural for people to manipulate such formal/prescriptive representations. Instead, people frequently sort items by rough notions of association or categorization. One natural organizational process has been found to center around manipulations of objects in spatial arrangements. Examples of this range from the organization of documents and other items on a regular office desktop to the use of 3"x5" cards to organize a conference program. Using visual cues and spatial proximity, people change the categorizations of and relationships between objects. Without the help of indices or perfect memory people can still interpret, locate, and manipulate the information represented by the items and the higher-level visual structures they form. The VITE system presented here is an intuitive interface for people to manipulate information in their own way and at their own pace. VITE provides for configurable visualizations of structured data sets so users can design their own "perspectives" and a direct manipulation interface allowing editing of and manipulation on the structured data.

Keywords
Information visualization, visual languages, structured data manipulation, multiple perspectives, direct manipulation, visual parsing, spatial pattern recognition

INTRODUCTION
It is natural for people to perceive and arrange information visually. In the course of conference planning or product development, we see people taking notes on note cards (3"x5" cards), post-it notes, or sheets of paper and arranging them on tables or bulletin boards. The spatial arrangement helps people categorize, perceive, and communicate embedded relationships. Figure 1 shows an example of such a physical layout resulting from a product design meeting.

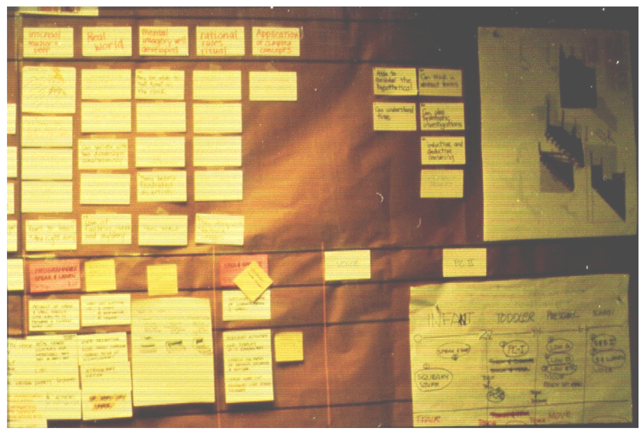


Figure 1: Note cards, post-its, and sheets of paper arranged during a product design session.

The process of mapping data on to a graphic representation is called visualization. Visualization techniques are well discussed in the research literature for graphic design [17] and information visualization [5], but the reverse operation, which we call graphic parsing, is rarely discussed together with it. Parsing is as important as visualization when deciding on a manipulable visual mapping because the visualization process must be reversible if the user's changes to the information are to be recognized and interpreted.

Our approach has been to use the visualization techniques and spatial arrangements that people naturally use to help people understand and manage structured information. These techniques have been incorporated into a visual interface for manipulating structured data called VITE.

The goal of the tool is to provide an intuitive interface for people to manipulate information. The system enables configurable visualizations of structured data sets -- data made up of multiple attribute values, such as frames or objects in a knowledge base or rows in a relational database. VITE includes a direct manipulation interface for graphically editing the structured data. By enabling multiple manipulable visual "perspectives" of data, people may organize, categorize, and manipulate information in their own way, making the characteristics that are most important for the particular person or task easiest to recognize. The multiple perspectives facility guarantees that the tool does not coerce the user into a premature commitment to a fixed visual structure, but allows the salient characteristics to change over time as the user discovers new requirements in their current task or identifies new uses for the information. Thus, not only is having one manipulable visual mapping of the information beneficial for many tasks, but supporting a variety of mappings for the same data will enable the system to support a wider variety of tasks over a longer period of time.

The next section of this paper describes VITE's interface and system architecture. A discussion of related work is then presented. This is followed by scenarios of use in a number of domains to convey the application and usefulness of the system and a discussion of issues related to the visual mappings for encoding structured data. We conclude with a description of our lessons so far and plans for future work.

VITE: DESIGN AND IMPLEMENTATION

Interface

VITE, a visual interface for manipulating structured data, is designed to support rapid visual/spatial interactions with information objects similar to those observed in the use of paper and electronic media [6]. The interface lets users view and edit information as if they were manipulating note cards on a working surface. VITE currently limits users to a two dimensional workspace where objects can overlap with each other.

The interface of VITE consists of a main working area similar to a drawing application (Figure 2). Data elements are represented as visual information objects, which operate much like desktop icons. Information objects can be moved and resized through direct manipulation. Other graphical properties, such as color and border width, can be modified by selecting visual attributes from the toolbar or from a popup menu.

VITE provides a two-way graphical mapping for moving information to and from the database or other knowledge store; we call this mapping a *perspective*. Each perspective encodes specific attributes of the data as graphical properties. The encoding for the information can be redesigned and saved for repeated use. Users may create a variety of different mapping profiles to serve as different perspectives for the same data set. Thus, VITE provides users of traditional databases or object stores the ability to view and manipulate the same set of data through a variety of visual mappings.

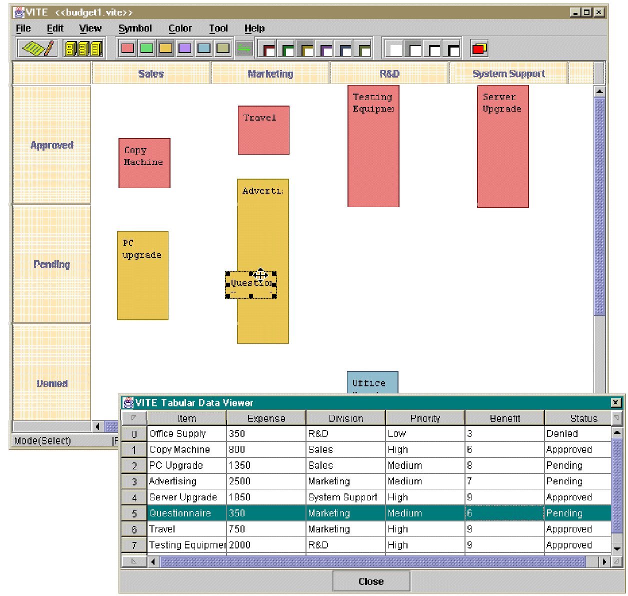


Figure 2: VITE interface showing the visual interactive working area and a tabular data viewer. Data elements are represented as visual objects in the working area and rows in the tabular viewer.

Once the database or knowledge base is mapped into a graphical display, information can be modified by dragging the object into a different area of the graphical layout or by changing other visual attributes (like color, border width, etc.). Interactions between the user and the interface are immediately interpreted by the system and the results are reflected in both the visual representation and in the structured information stored in the database. This allows the possibility that synchronous users could manipulate the same data set via different visual representations given such support from the database/knowledge-base layer.

System Architecture

The basic skeleton of VITE is similar to an interactive drawing program that provides simple object manipulation capabilities on the graphical display. This similarity allows skill transfer so that most users have no trouble learning to manipulate the data presented. Figure 3 shows the VITE component architecture.

The *Mapping Engine* is the heart of VITE. It contains a *Graphic Parser* and a *Graphic Renderer*. The mapping engine calculates the region allocated for the information categories in the data set and stores the boundaries and mapping criteria.

The graphic parser interprets user input and intentions interactively. The graphic renderer projects the data set into the graphical display according to the mapping assignments. Working together, the graphic parser and graphic renderer synchronize interpreted results between the structured data store and the graphical displays.

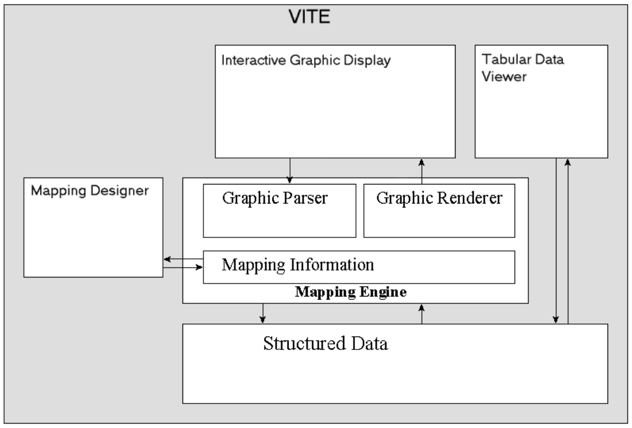


Figure 3: VITE components.

VITE also provides a traditional *Tabular Data Viewer* (Figure 6). It shows the structured data a table listing attribute values in a traditional way. Updates to data either in the visual workspace or in the table viewer are immediately reflected in the other.

The *Mapping Designer* (Figure 9) presents the visual mapping assignments to the user and can also be used to edit or create new visual mappings. Each attribute is listed with information about the types of values allowed and selections for visual attributes. The mapping designer and its use will be described in more detail later in the paper.

RELATED WORK

The design of VITE has been influenced by research in two related areas: (1) visualization systems and (2) systems supporting the direct manipulation of structured data.

Visualization Systems

Most visualization systems focus on transforming structured or unstructured data into a graphical display according to visualization constraints such as what characteristics of the data can be mapped to visual attributes. Visualization systems can help users perceive and understand patterns in large information spaces [11].

Robertson and colleague's Information Visualizer [10] and Kumar's Timeline project [4] provide examples of assigning visualization mappings for specific tasks. The attributes of a database are assigned to different graphical properties according to different task domains. Visualization can be a useful tool for identifying desired items in a corpus. Musai [7] demonstrates the effectiveness of combining data visualization, keyword search, and category search when looking for information using multiple, vaguely defined criteria.

Research aiming to provide multiple visualizations of a data set at the same time includes FOCUS [15], FINESSE [18], SSR & SIV [2] and Table Lens [9]. For example, FOCUS is an interactive table viewer for comparing multiple attributes in a table using fisheye techniques. The configurable focus+context for data exploration shares some concepts with VITE in the support for task-dependent or user-dependent visualizations of information. SSR, SIV, and FINESSE use a spreadsheet environment to view and edit multiple visualizations at the same time. The Table Lens is also a spreadsheet environment for exploring large data sets but with support for focusing attention on subsets.

SAGE [12] and GlyphMaker [19] aim at providing configurable visualizations of data. These systems allow visualization mappings to be edited using diagram manipulation and schema drawing. SAGE includes an intelligent mechanism for generating visualizations of data.

Software visualization systems, like Tango [16], couple algorithm execution to animation components with user designed mappings. The Tango framework focuses on algorithm animation and reveals some mapping design issues.

The systems described provide a variety of means to perceive, understand, and find information. They do not support easy modification of data once the data is mapped to the visual display or for visual changes to the data to result in changes to the underlying data.

Direct Manipulation of Structured Data

VITE uses direct-manipulation and visual attributes to modify the information displayed. Direct manipulation is common in operating system desktops and in drawing and graphics software due to the intuitive and efficient interaction provided to the user. Hollan [3] and Ballas [1] concluded that performance advantages and direct engagement are benefits of direct manipulation interfaces.

The second author's prior work on VIKI [14] combines a direct manipulation interface with spatial structure recognition but does not include a visualization system to generate initial views. VITE extends the work on VIKI by adding a graphic renderer to generate initial views, broadening the recognition capabilities to include mappings between visual and semantic attributes.

Tivoli [8], software for electronic whiteboards in meeting room situations, shows the possibility of interpreting direct manipulation interaction techniques based on boundaries and the implicit recognition of regions. Tivoli's domain objects are also similar to VIKI and VITE visual symbols, providing manipulable visual representations of domain-specific data.

Visage [13] provides an environment for information analysis that integrates a variety of mini-visualizations for specific tasks or applications. Users can drag information between the various visualizations and can modify elements within the visualizations (e.g. changing the color of specific objects.) Such changes are reflected in the other visualizations but have no semantic effect on the data.

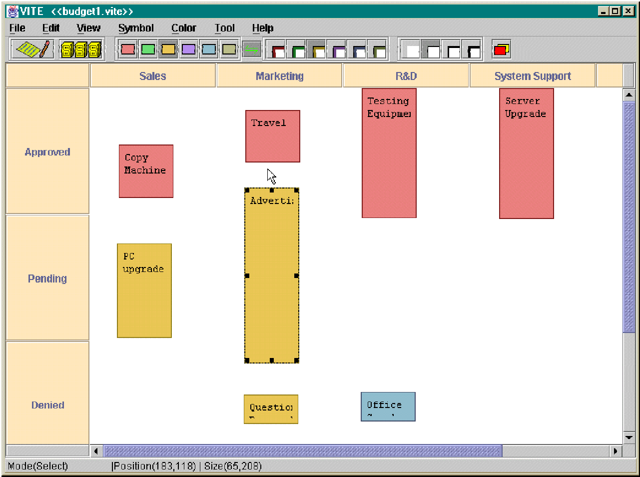


Figure 4: Budget meeting scenario with mapping setting shown in table 1.

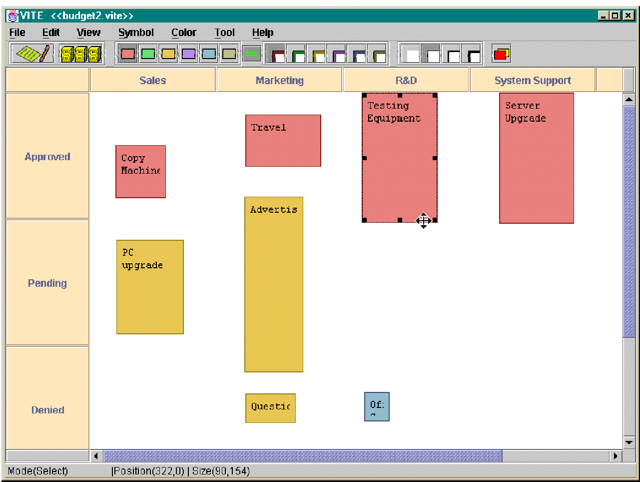


Figure 5: Budget meeting scenario, with semantic attribute "productivity benefit" mapped to visual attribute "width".

VITE, combining the concepts of visualization and direct manipulation with a general purpose two-way mapping engine, allows users to selectively visualize their data with preferred mappings according to their tasks and needs, and modify the database contents within the visual workspace.

EXAMPLE TASKS

This section provides examples of VITE use to motivate further discussion. The first example involves using VITE in a budget meeting. In this scenario, a total budget has been set for four departments and the task is to select expense requests that will be approved. Each expense has been assigned a priority and an expected productivity benefit. Spreadsheets, or a calculator and piece of paper, might be the most common tools for this task. VITE is not as precise as these tools for quantitative calculations, but it may benefit from the visual representation of individual expenses and the overview of current decisions.

As an example, the department managers decide on a first visualization of the requests and their status (shown in Figure 4) using the visual mapping in Table 1:

Visual attribute	**Semantic attribute**
Vertical Size	Budget amount/expense
Horizontal position	Department
Color	Priority
Vertical position	Status (approved/pending/denied)

Table 1: Mapping 1 for budget meeting.

The managers later decide to take into account the expected productivity benefit of pending requests and modify the mapping by setting the horizontal size to the productivity benefit. This second perspective (Figure 5) provides additional clues for obtaining a beneficial decision while staying within the budget limit. Now the width/height ratio of an expense indicates the expected payback on investment. When the managers decide to change the amount of a funding request, they use the tabular data, shown in Figure 6, since resizing the height would result in imprecise changes.

VITE Tabular Data Viewer

	Item	Expense	Division	Priority	Benefit	Status
0	Office Supply	350	R&D	Low	3	Denied
1	Copy Machine	800	Sales	High	6	Appproved
2	PC Upgrade	1350	Sales	Medium	8	Pending
3	Advertising	2500	Marketing	Medium	7	Pending
4	Server Upgrade	1850	System Support	High	9	Appproved
5	Questionnaire	350	Marketing	Medium	6	Denied
6	Travel	750	Marketing	High	9	Appproved
7	Testing Equipment Pu	2000	R&D	High	9	Appproved

Close

Figure 6: Budget meeting, Tabular View.

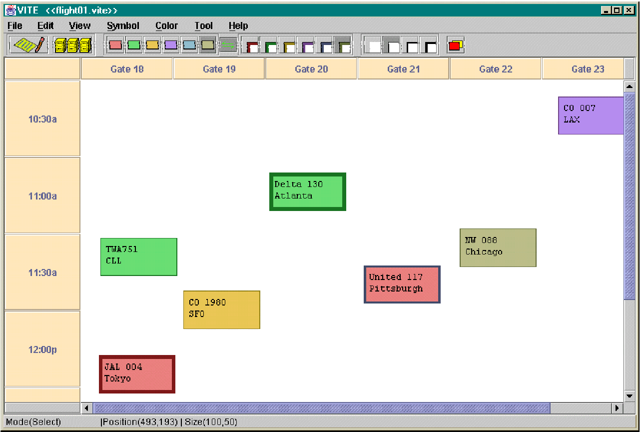


Figure 7: Airport flight management. A perspective used by ground control for terminal/gate assignment. Color, border width, and border color are assigned to aircraft type, engine type, or size category respectively. E.g. twin-engine, tri-engine, or four-engine; jet, propeller, or turbo fan; private, passenger, or cargo, etc.

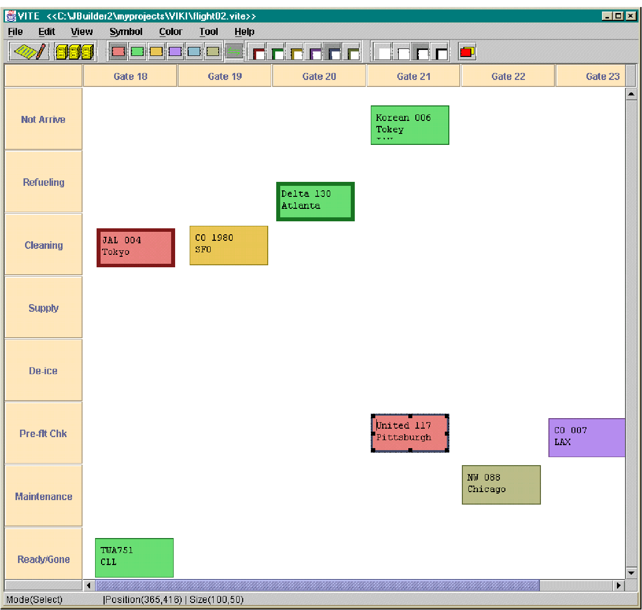


Figure 8: A perspective for ground crews to monitor aircraft preparation status, and help managing resource allocation among these gates.

The second example is an airport flight status management environment. In this scenario, passengers need easy-to-understand information about their flights and controllers, flight crews, ground crews, and gate personnel need the ability to review and update aspects of the flight status. Although all aircraft are monitored by all controllers, different tasks are performed by each controller. For example, approach control and departure control focus mainly on the approach/departure route and runway assignment. Ground control focus on taxi sequence, taxi way clearance and parking/terminal assignment. Airline liaisons are interested in the schedule status (on time, delayed, or canceled) of their aircraft.

Using VITE, visualizations can be tailored for each class of user so the relevant aspects of flights are clearly illustrated and easy to alter. Figure 7 shows a possible mapping used by a controller. When the airport personnel need to change any information, they can directly manipulate the attribute using their tailored perspective. Not currently in VITE, security could also be included so only certain information is editable in each perspective. Figure 8 shows a mapping of the same data set but with the monitoring and updating information needed by ground crews. Notice some of the information is needed in multiple views.

These two scenarios demonstrate some potential uses of VITE. VITE is useful for applications which require:

- easy to perceive structured information,
- constant modification of information,
- objects identified in multiple categorization schemes,
- multiple perspectives of the same data, and
- shared manipulation of structured information.

VISUAL MAPPINGS OF STRUCTURED DATA

General Design Principles

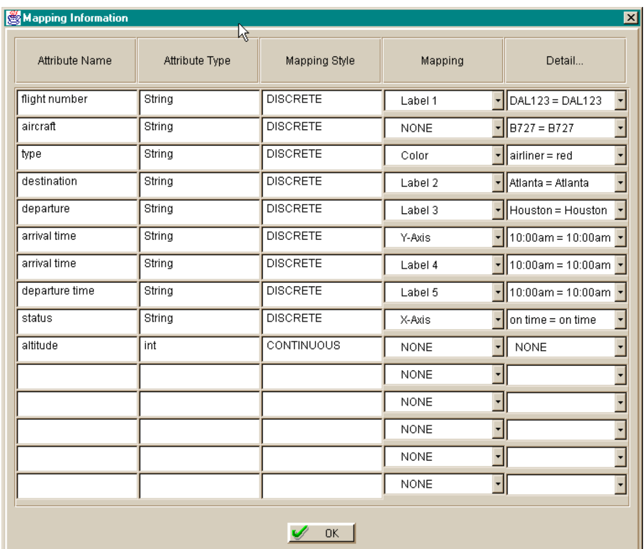
To support the users in the two scenarios above, VITE needs a two-way mapping engine. The mapping engine needs mapping information which is defined as a perspective for the users. This mapping information can be pre-determined and stored in a file or can be designed by the user. This information includes the mapping style, mapping constraints, graphical properties, attribute mapping pairs and special characteristics for each attribute in the database.

In the mapping designer interface, mapping information must be checked for conflicts. For example, consider visualizing car engine performance where a car's velocity is mapped onto the X-axis. In this case, the width of the information object is not yet defined and can be assigned to represent another attribute such as acceleration. If the information object represents a historical event, we may want the X-axis to display the starting time and ending time of the event. In this case, the width (X-size) of the information object is predetermined and cannot be assigned to any other attribute. Such constraints will be checked when the user designs a new mapping using the mapping designer.

Position may also be used to represent discrete attributes as the Y position did in the budget scenario in the prior section. For discrete data, information objects are parsed according to their region, not by a specific location value. These region boundaries are pre-calculated and stored in mapping database when the data is loaded into the working area.

Mapping Designer

A preliminary version of the mapping designer is shown in Figure 9. In this interface, users can match semantic attributes of the data set to visual attributes in the display. This tabular interface provides information about the semantic attribute (its name, type, and style.) Users select visual attributes and details the mapping from menus on the right. A new mapping designer is currently under construction. It will provide a drag and drop interface where graphical attributes are represented as icons located in a "mapping pool". Users drop the desired visual attribute icon on the semantic attributes to connect them. The system will restrict the use of visual attribute icons to enforce any exclusion or constraints, such as the ability to use only two of the X-size, X-start and X-end visual attributes, as discussed earlier.



Attribute Name	Attribute Type	Mapping Style	Mapping	Detail...
flight number	String	DISCRETE	Label 1	DAL123 = DAL123
aircraft	String	DISCRETE	NONE	B727 = B727
type	String	DISCRETE	Color	airliner = red
destination	String	DISCRETE	Label 2	Atlanta = Atlanta
departure	String	DISCRETE	Label 3	Houston = Houston
arrival time	String	DISCRETE	Y-Axis	10:00am = 10:00am
arrival time	String	DISCRETE	Label 4	10:00am = 10:00am
departure time	String	DISCRETE	Label 5	10:00am = 10:00am
status	String	DISCRETE	X-Axis	on time = on time
altitude	int	CONTINUOUS	NONE	NONE
			NONE	
			NONE	
			NONE	
			NONE	
			NONE	

Figure 9: Mapping Designer.

The mapping designer must also serve as a data analyzer because the system needs all the potential values of an attribute to determine the spatial parsing boundaries. This information is also useful when choosing a graphical mapping.

For instance, if an editable attribute has only 5 or 6 possible values, it may be suitable for a color mapping, or a border width mapping, but if the attribute has too many possible values, the user should consider using X/Y-position or simply text label mappings.

Deciding which semantic attribute is mapped to which visual attribute is complex. The following sections discuss issues related to some of the graphical properties in the mapping model.

Graphical Properties

Designing graphical mappings combines issues of designing graphical presentations with issues of the manipulability of attributes and parsibility of results. Graphical presentations can be viewed as sentences of graphical languages [6]. Graphical properties, such as position, encode functional information. Therefore, the choice of graphical properties has a great influence on the design of graphical languages, and thus the design of graphical presentations.

Visual effectiveness and expressiveness criteria [6] determine if a particular graphical presentation will result in a good design. These criteria are still important concerns in the mapping design, but for VITE, we have to consider not only the visualization effectiveness but also manipulation effectiveness.

The graphical properties manipulable in VITE are: position, size, color (both border and internal), border width, and label. The following identifies characteristics of these properties which affect mapping decisions.

Position: Position is one of the most obvious and distinguishable properties for visual perception, and is the most effective property for direct manipulation. Thus it is our first choice of graphical properties for semantic attributes requiring frequent manipulation.

Position can be used for two classes of visual mappings [4], one is categorical mapping, for representing discrete-valued attributes, and the other is numerical mapping, for representing continuous-valued attributes.[1] The number of discrete values that can reasonably be mapped to position is dependent on screen real estate and object size.

Size: The second spatial property of objects in VITE is size. As previously explained, sometimes this property is exclusive of the "position" property. One consideration in the assignment of attributes to the vertical and horizontal size properties is that they effect the perceived volume of the object as a whole. Size can be used for both continuous and discrete values. The number of discernable discrete size values depends on the screen's size.

Color: Color may or may not be a very distinguishable property. It can be easy to identify a small set of colors, e.g. red, blue, yellow, etc., but it can be tricky to distinguish shades of gray. Nevertheless, color is an important graphical property for information visualization. It is difficult for users to modify color precisely enough for continuous data. Thus, the color property is recommended to be used for discrete mapping or as a display-only attribute for continuous data. In VITE, both border color and background color can be assigned to each visual symbol.

---

1. Mackinlay distinguished between three styles: Quantitative, Nominal, and Ordinal.

It is also tedious to define the mapping pairs (data attribute value and color pairs) if you have to manually define them. Because there is not always an obvious way to assign colors to attribute values, a good mapping does not come directly by scanning all the candidate values from the database. VITE preprocesses and scans and can assign the mapping randomly or sequentially based on available colors. But this approach is as likely to choose red for “cold” and blue for “hot” as the other way around. For automatically generated color mappings, the mapping information needs to be exposed in advance.

Border width: Border width can be used to represent semantic attributes that have only a few possible values but need to be easily distinguishable. VITE uses easy-to-reach applicable tool buttons to support the rapid change of border width and color.

Label: Since labels are presented as text, assigning an attribute to the label is the most direct way to retain the information stored in the database in the visualization. Labels are also useful for making sure the users can understand the value of the attribute. The problem with using labels is that it supports little, if any, direct manipulation. Modifying a label is not any different than modifying a text-editor-based database.

Resolution and Error in Mapping Design

Resolution and error must be considered when selecting a two-way mapping. Error, the inaccurate selection of visual attribute values for given data values, is not a problem for visualization (data-to-graphics mapping) because data can be deterministically mapped to a visual attribute. Resolution remains an issue since the number of data attributes and values that need to be represented visually may require many semantic values map to similar visual values. Such mappings are more likely to cause user confusion.

Error is a problem for parsing (graph-to-data mapping), because users manipulating objects with a mouse cannot be expected to be as precise as the system is calculating the values and assigning the mappings. One simple solution for this problem is to provide real time display of the mapping-information (e.g. the real value-change of the operation) while the object is dragged on the screen. Other possible solutions include: layered operation for the requirement of precise manipulation, so the users do the coarse operation by direct manipulation and fine tune the value with the help of a value input or slide controller interface. Grid constraints also can be used and can be predefined so the values will always fall into these grid slots as desired.

Due to the limited number of easily manipulable visual attributes, VITE enables multiple perspectives so users can work with their favored perspective for their current need. A traditional text-based database editor interface is provided as well, so as to not force the users to change the perspective whenever they need to modify rarely modified attributes or in situations when a text-based table is appropriate.

DISCUSSION AND FUTURE WORK

There are limitations to the semantic and visual representations that VITE currently supports. One such limitation is that it cannot visually represent explicit relationships between data objects. Explicit relationships in the database would be naturally represented by links in the graphical environment, but such links are currently not supported.

Multiple visual occurrences for one data object is another challenge. Consider the case of a data categorization task. Different users may want to classify the same object in different categories, or it simply belongs to two or more categories. For example, we may want to classify “Industrial Design” as an “art” subject, but some may classify it as a “science” subject, or perhaps it belongs to both categories. For better support of progressive categorization, VITE should encourage vague classifications as well as distinct categorizations. Taken one step further, the user may even specify the classification using a visual attribute to assign a rough belief or expectation value, say 70% towards “art” and 30% “science”, so “Industrial Design” is classified more “art” than “science”.

SUMMARY

The use of two-way visual mappings to support the manipulation of structured data has many potential uses. By providing users with multiple perspectives of their data which they can edit through a direct manipulation interface, VITE enables work practices seen in other media.

Through VITE, we are investigating the combination of visualization and graphical parsing to provide domain-and task-specific interfaces to structured data sets. This work has pointed out the complexity of designing good visualizations that also work as good visual editors. The work on VITE suggests many additional research issues worth investigation.

ACKNOWLEDGMENTS

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

REFERENCES

1. James A. Ballas, Constance L. Heitmeyer and Manual A. Perez. Evaluating two aspects of direct manipulation in advanced cockpits. In *Proceedings of CHI ‘92*, May 3-7, 1992, pp.127-134.
2. Ed Huai-hsin Chi, Phillip Barry, John Riedl, Joseph Konstan, A Spreadsheet Approach to Information Visualization. In *Proceedings of Information Visualization ‘97*, pp.17-24.

3. Edwin L. Hutchins, James D. Hollan, and Donald A. Norman, Direct manipulation interfaces. In Donald A. Norman & Stephen W. Draper (Eds). *User-centered system design*. Erlbaum Associates, Hillsdale, NJ, 1986, pp. 87-124.
4. Vijay Kumar, Richard Furuta, and Robert B. Allen, Metadata Visualization for Digital Libraries: Interactive Timeline Editing and Review. In *Proceedings of Digital Libraries '98*, pp.126-133.
5. Jock D. Mackinlay. Automating the design of graphical presentations of relational information. In *ACM Transactions on Graphics*, Vol. 5, No. 2 April 1986, pp. 110-141.
6. Catherine C. Marshall, and Frank M. Shipman III. Searching for the Missing Link: Discovering Implicit Structure in Spatial Hypertext. *Proceedings of Hypertext '93*, Seattle, Washington, November 14-18, 1993, pp. 217-230.
7. Toshiyuki Masui, Mitsuru Minakuchi, George R. Borden IV, Kouichi Kashiwagi. Multiple-view Approach for Smooth Information Retrieval, In *Proceedings of UIST '95*, November 14-17, 1995, pp.199-206.
8. Thomas P. Moran, Patrick Chiu, William van Melle. Pen-based interaction techniques for organizing material on an electronic whiteboard. In *Proceedings of UIST '97, October*,1997, pp. 45-54.
9. Ramana Rao and Stuart K. Card. The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. *Proceedings CHI '94* Human Factors in Computing Systems, ACM, April 1994, pp.318-322.
10. George G. Robertson, Stuart K. Card, and Jock D. Mackinlay. Information visualization using 3D interactive animation. *Communications of the ACM*, 36(4), April 1993, pp.57-71.
11. Russell R. Rose, *P1000 Science and Technology Strategy for Information Visualization*, September 1996.
12. Steven F. Roth, John Kolojejchick, Joe Mattis, Jade Goldstein, Interactive Graphic Design Using Automati Presentation Knowledge, in *Proceedings of CHI '94*, pp.112-117.
13. Steven F. Roth, Peter Lucas, Jeffrey A. Senn, Cristina C. Gomberg, Michael B. Burks, Philip J. Stroffolino, John A. Kolojejchick and Carolyn Dunmire, Visage: A User Interface Environment for Exploring Information, in *Proceedings of Information Visualization '96*, pp.3-10.
14. Frank M. Shipman, Catherine C. Marshall, and Thomas P. Moran, Finding and Using Implicit Structure in Human-Organized Spatial Layouts of Information. *Proceedings of CHI '95*, Denver, Colorado, May 7-11, 1995, pp. 346-353.
15. Michael Spenke, Christian Beilken, Thomas Berlage, FOCUS: The Interactive Table for Product Comparison and Selection, in *Proceedings of UIST '96*, pp.41-50.
16. John T. Stasko, Tango: A Framework and System for Algorithm Animation, in *IEEE Computer*, Sept 1990, pp.27-39.
17. Edward R. Tufte, *The Visual Display of Quantitative Information*, 1983.
18. Amitabh Varshney and Arie Kaufman, FINESSE: A Financial Information Spreadsheet, in *Proceedings of Information Visualization '96*, pp.70-71.
19. 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.