Manipulating Structured Information in Spatial Hypertext

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
{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. VITE uses information visualization techniques to render structured information in the workspace and provides users an environment to interact with information in a spatial hypertext setting. 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, spatial hypertext, information workspace, information interfaces and presentation

INTRODUCTION
Since the invention of computers, the information generated and manipulated in our cultures has become more formalized and structured in order to let computers process it. 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 [12]. Instead of interpreting the dissected 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 data records in a relational database that contain a set of data fields, frames, or objects with attributes in a knowledge base.

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 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 "non-destructive formalization" during knowledge building that keeps the original "less formal representation as well as the formal representations" [11].

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.

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 [16] of the representation 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.

Intermediate Representations for Problem Solving
Problem solving usually involves information gathering, categorizing, and knowledge building [17]. 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 [6].

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 to users is through information visualization [1]. 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.

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 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. The approach presented here is to take the unidirectional mapping of visualization systems, and make it bidirectional. Two-way mappings visualize the structured information (as visualization systems do), while also parse visual-property changes made by the user. This poses new mapping constraints when compared to visualization, since user edits must be reasonably interpretable.

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 [3] and [4]. VITE's design and development have been influenced by a variety of prior work in the area of hypertext and visualization. Aquanet [7] and VIKI [8] used spatial layout for arranging relationships among information objects. HOS [13] and Aquanet added an abstract layer by allowing attributes, relationships and types to be associated with information objects. Tivoli [9] incorporated specific domain knowledge into the system to support implicit perception in a free-form interaction environment. VKB [14] utilizes both spatial layout and abstract attributes for supporting incremental formalization within user-created information workspaces.

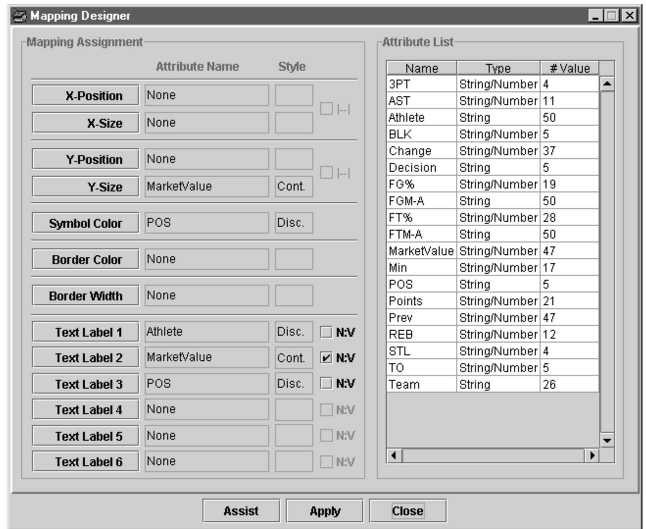

Figure 1 : Mapping Designer Interface.

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 (Figure 2) used to view and edit the information, the mapping designer (Figure 1) used to create and edit two-way mappings, the mapping engine which instantiates the two-way mappings, and the data store.

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.

USER STUDY

A study of how people use two-way mappings was conducted using VITE. 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, 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

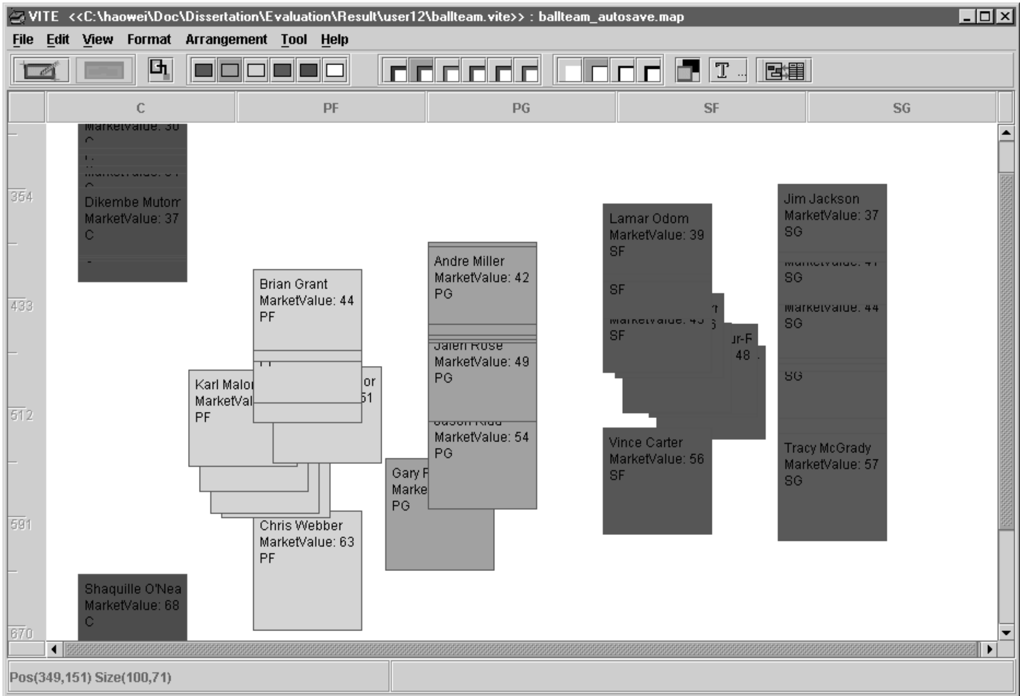

Figure 2 : Workspace containing visual symbols representing basketball players being organized during selection of a fantasy basketball team.

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 [15].

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 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 1). Improvements in performance and intuitiveness of VITE's interface, although not much worse than that of efficiency, did not impress users as much.

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. The subject was also the one who most utilized the 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.

Table 1 : 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

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 particular visual property.

Unique values and names for each entity exist in structured data but were not always used by the subjects. In one case, the subject decided not to display the name of the basketball players during the task. 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 the players 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.

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 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.

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 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.

ON GOING PROGRESS

While the results showed subjects found the tool to make their task more efficient, it also showed that there are issues surrounding ease of use in the generation of good mappings. The Mapping Assistant is an on-going design 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 [10]. 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 base 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

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. 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. Advances in computing technology such as graphical environments and information workspaces 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 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 [5] and [2] 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.

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.

REFERENCES

1. 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.
2. 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.
3. 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.
4. Hsieh, H. and Shipman, F. M. III 2002. Manipulating Structured Information in a Visual Workspace. *UIST 2002*. 217-226.
5. Mackinlay, J. D. 1986. Automating the Design of Graphical Presentations of Relational Information. *ACM Transactions on Graphics, 5, 2* (April), 110-141.
6. 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.
7. 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.
8. Marshall, C. C. and Shipman, F. M. III 1997. Spatial Hypertext and the Practice of Information Triage. *Hypertext 1997*. 124-133.
9. 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.
10. Schön, D. A. 1983. *The Reflective Practitioner – How Professionals Think in Action*. Basic Books, New York, NY.
11. Shipman, F. M. III 1993. Supporting Knowledge-Base Evolution with Incremental Formalization. *Ph.D. Dissertation*, Department of Computer Science, University of Colorado, Boulder, CO.
12. 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.
13. 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.
14. 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.
15. Simon, H. A. 1996. *The Sciences of the Artificial*. The MIT Press, Cambridge, MA.
16. Suchman, L. A. 1987. *Plans and Situated Actions: The Problem of Human Machine Communication*. Cambridge University Press, New York, NY.
17. 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.