

Finding and Using Implicit Structure in Human-Organized Spatial Layouts of Information

Frank M. Shipman III, Catherine C. Marshall

Department of Computer Science
Texas A&M University
College Station, TX 77843-3112
(409) 862 - 3216
{shipman, marshall}@cs.tamu.edu

Thomas P. Moran

Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304
(415) 812 - 4351
moran@parc.xerox.com

ABSTRACT

Many interfaces allow users to manipulate graphical objects, icons representing underlying data or the data themselves, against a spatial backdrop or canvas. Users take advantage of the flexibility offered by spatial manipulation to create evolving lightweight structures. We have been investigating these implicit organizations so we can support user activities like information management or exploratory analysis. To accomplish this goal, we have analyzed the spatial structures people create in diverse settings and tasks, developed algorithms to detect the common structures we identified in our survey, and experimented with new facilities based on recognized structure. Similar recognition-based functionality can be used within many common applications, providing more support for users' activities with less attendant overhead.

KEYWORDS: emergent structure; spatial diagrams; spatial structure recognition; informal systems; hypermedia.

INTRODUCTION

Many kinds of software, ranging from operating systems to graphics editors to collaboration substrates to hypermedia applications, provide users with two or two-and-a-half dimensional spaces in which they can organize information in the form of graphical objects. These manipulable objects include icons or other visual symbols representing larger pieces of information, as well as complete chunks of information. People tend to follow certain conventions when they lay out these objects to indicate relationships among them. We are investigating methods of recognizing such implicit spatial structure and using it to support users' activities.

We first examined the kinds of spatial layouts that people produce in both computational and noncomputational media. Based on these results, and the results of analogous

studies of the way people organize materials in their physical environments [4][6][7], we developed a "spatial parser" to recognize common structures. The parser is built to be portable and extensible so that it can be added to new systems or augmented to recognize new kinds of structures with little effort.

We integrated the parser with VIKI [11], a spatial hypertext system, to explore the use of recognized structure. VIKI currently includes two types of support based on the parser. First, VIKI provides users with ready access to implicit structures. Second, at user initiative, VIKI suggests formally represented structures based on the implicit structure.

The next section summarizes our study of user-constructed spatial arrangements and the conventions we found. We then describe the architecture developed for recognizing these common spatial structures. Finally, we discuss the application of structure recognition to support users performing information management and analysis tasks in VIKI.

LOOKING AT PRACTICE

Experiences with the use of Aquanet [9] led us to recognize the importance users place on implicit spatial structures. This realization prompted us to perform a survey of spatial structures people created in diverse hypermedia systems and in non-computational settings (such as notecards and Post-its stuck on a wall) [10].

As part of the survey, we analyzed nine layouts, each the result of a long-term information management or analysis task. Three of the layouts were created in NoteCards [2], one in the Virtual Notebook System [14], and three in Aquanet [8]. The two non-computational layouts in our survey involved wall-sized arrangements of 3" x 5" cards, Post-its, and other pieces of paper. Table 1 shows the task, source, and number of objects for each of these layouts.

To perform the analysis, we introduce three analytic abstractions. The first is the notion of graphical *information objects* already discussed. Also, we use the idea of *object type*. Here, type is used to reflect an object's distinguishing visual characteristics and functional role within a given spatial layout. We also use the idea of *spatial structure*,

task	source	# of objects	# of types
comparing editors	paper	72	3
product design analysis	paper	284	8
anthropological analysis	NoteCards	36	3
anthropological analysis	NoteCards	20	3
anthropological analysis	NoteCards	50	3
bug tracking (VNS)	VNS	63	6
bug tracking (Aquanet)	Aquanet	130	6
analysis of tech. field	Aquanet	1506	14
analysis of software	Aquanet	193	6
writing paper	Aquanet	74	3

Table 1: Spatial layout task, source, and data characteristics

which is a perceptually distinct group of objects. Table 1 includes the number of object types identified in our analysis.

Following this analytic framework, we encoded the data in a canonical form which recorded spatial and visual aspects of the graphical information objects, including the relative planar location and the extent of each object. Each object was also assigned a type based on its system type (if it had one) and distinguishing visual characteristics (for example, font or color). Discussions with the people who created the layouts helped us to understand the intended structures in each example and their meaning. We discovered that a small set of common structures, like stacks and lists, were common across the layouts, even though they were created in a variety of systems and in service of different tasks.

Analysis of the content and function of the objects used in these structures revealed that people used spatial layouts to represent different types of relationships among constituents. First, spatial structure was used as a means of categorization or to build up sets. Second, spatial proximity was used to indicate specific relationships among objects or among types of objects. Finally, spatial arrangement was in some instances dictated by the way in which objects were used together in a task.

The following two examples, one computational, one non-computational, illustrate the kinds of conventional structures we found, and preview the issues that arise in developing heuristics for recognizing these structures and using them to support users' activities.

Example from Aquanet

Figure 1 is an example of the kind of layouts we gathered for our survey; the layout is the result of a group effort to write a paper in the Aquanet hypertext system. Each long thin gray rectangle represents a major topic heading;

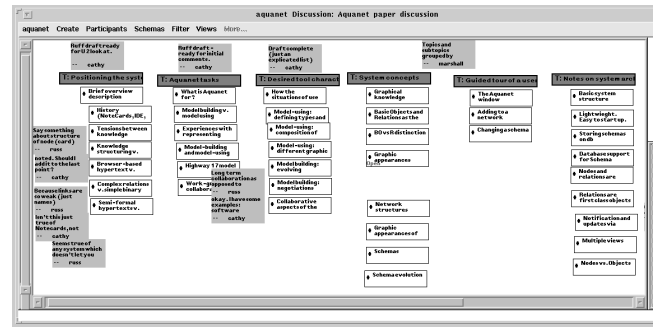


Figure 1: Example of spatial layout from Aquanet

outlined rectangles represent subtopics (and the underlying paragraphs explicating them); and thicker gray rectangles are annotations, notes from one author to the others. In Aquanet, these visual properties are solely dependent on the object's type in the system. The other computational sources, however, did not have this notion of object type in the system but the authors had improvised, creating visually apparent object types using different fonts and colors for objects.

It is readily evident to human perception that there is a significant amount of structure in the spatial layout in Figure 1. Yet very little of this structure is expressed in such a way that it is accessible to the system in which the layout was created (even though the system provided users with a mechanism for expressing just such structure).

Figure 2 is a close-up of one of the structures in Figure 1. The arrangement includes a list with six similar elements (of type subtopic); it is a good example of the kind of structure people use to express categories or sets. This list is part of a higher-level structure that includes the list and its heading, which is of a different type. We refer to this kind of structure (one consisting of a regular pattern of different object types) as a *composite*. The list of annotations on the right side of the list refers to a portion of the list. Unlike the rest of the structure, without examining contents, this sort of reference is more idiosyncratic and remains ambiguous to human perception. In our analysis of the layouts for common structures, we resolved such ambiguities by looking at textual content and talking to the original authors.

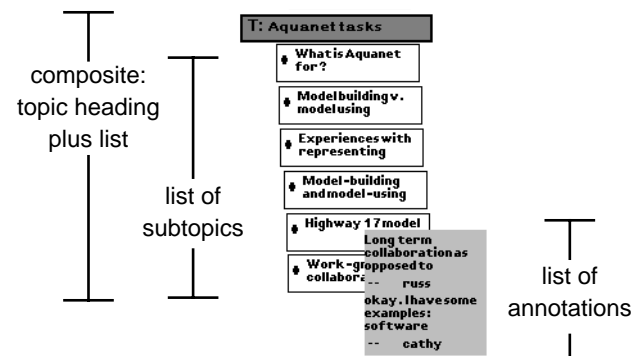


Figure 2: Close-up of structure with description

Referring back to Figure 1, we can also see that this list is part of a larger structure of six such lists, some of which are annotated, arranged horizontally across the space. It is a structure constructed in service of the larger activity of writing a paper.

Non-computational Example

Figure 3 shows a diagram of one of the non-computational layouts from our survey. This layout consists of around two hundred eighty 3"x5" cards, Post-Its, and other paper covering a good-sized wall. The arrangement was used to analyze consumer behavior for a product design in an industrial design firm.

In this non-computational arrangement, color (and kind) of paper and marker are used to visually reflect a notion of object type similar to that supported by Aquanet and improvised in the other systems. Objects shown in the diagram in Figure 3 have this information encoded in their shade of gray. While the majority of cards contain handwritten text, there are also some pictures and diagrams; these cards were assigned a different type because they were so visually distinct.

As in the prior example from Aquanet, there is an apparent structure to this diagram. The cards on the top and to the left act as labels resulting in an incomplete matrix of lists. A distinct horizontal row of cards divides the matrix at its center. Annotations take the form of cards and Post-Its attached to or next to cards they discuss.

In summary, our data analysis uncovered sufficient regularity to support the idea that automatic recognition of implicit spatial structure was possible, and that it could identify several useful kinds relations.

COMMON SPATIAL STRUCTURES

Based on our survey of spatial arrangements, we developed an initial set of primitive spatial structures: lists (aligned objects of the same type), stacks (overlapping objects of the same type), composites (repeated arrangements of alignment between objects of different types), and heaps (overlapping objects of different types.) These four primitives may be composed into higher-level structures,

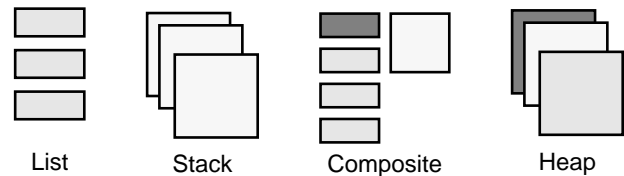


Figure 4: Examples of primitive spatial structures

like lists of composites. Figure 4 shows abstract examples of these four types of structure.

Lists. Lists are objects of the same type that are vertically or horizontally aligned; they were by far the most common spatial structure we identified in our survey, found in almost every example. Lists were used primarily for categorization: membership in a particular list indicated elements shared common features. Some of the lists in the diagrams included internal ordering principles, such as importance or temporal sequence. Also, some lists included groupings of more closely related items. Because these additional types of structure relied on analysis of object content, they are not represented in our list of primitive spatial structures.

Stacks. Stacks are objects of a single type that overlap significantly; they tended to be used for categorization when there was not enough space for lists. By arranging objects into stacks, the user has decided to favor compactness over ease of access since even partially obscured objects are more difficult to manipulate or view.

Composites. Composites are regular spatial arrangements of different types of objects; they were the most interesting kind of structure in our survey because they were used to denote higher level abstractions, relationships among types of objects. For example, in our survey, we found a repeated pattern of “implementor” objects above “system” objects; this pattern suggested an implicit “developed” relation. Similarly, labels often appeared above lists. In both cases, the spatial arrangement represents an implicit semantic relationship among the objects; it is a one-to-many relationship in the label/list case.

Heaps. Heaps are overlapping collections of different types of objects. They are usually created during the course of a

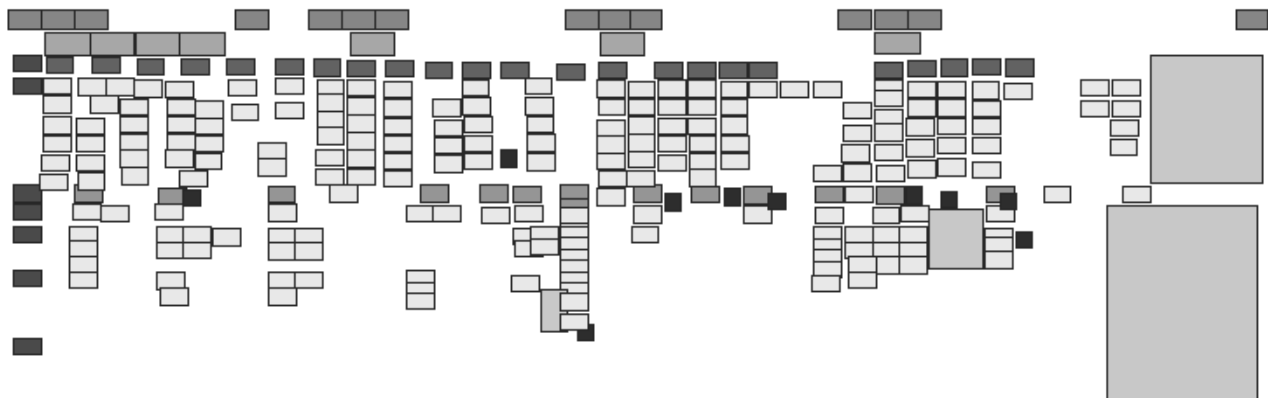


Figure 3: Diagram of physical layout of paper created during product design meeting.

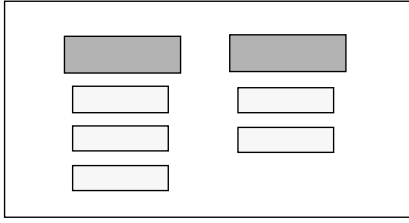


Figure 5: Sample graphic layout

sorting task as a strategy for managing a crowded space. People group dissimilar objects with the intention of spreading them out again once space becomes available.

RECOGNIZING SPATIAL STRUCTURE: AN EXAMPLE

We have implemented a spatial parser to recognize the common spatial structures identified in our observations. This section uses an example layout (shown in Figure 5) and its resulting parse to motivate our discussion of automatic recognition. The recognition architecture and its use will be described in later sections of this paper.

The layout shown in Figure 5, a simplified version of the arrangement in Figure 1, contains seven individual objects of two distinct types. The initial structures, described using the spatial primitives, are two vertical lists of smaller light gray objects. These lists, in turn, are part of two instances of a composite type that consists of a larger darker gray object over a list of light gray objects. Because these are instances of the same type of composite and they are aligned, a horizontal list with two elements is the next result of the parse.

Figure 6 shows the final parse tree of the spatial layout from Figure 5. Non-terminal nodes of the graph reflect the primitive and intermediate structures that may be identified through analysis of spatial layout, including Horizontal list, Vertical list, and Composite. Terminal nodes show graphic depictions of the original types from the layout.

The recognition algorithms were not tuned to any particular style of layout, and use only the location (x and y position), extent (width and height), and system-supplied type of the information objects. This limited set of characteristics was used so that the parser could be easily adapted to a variety of systems. For systems that do not include a notion of type, the visual characteristics of objects may be used to

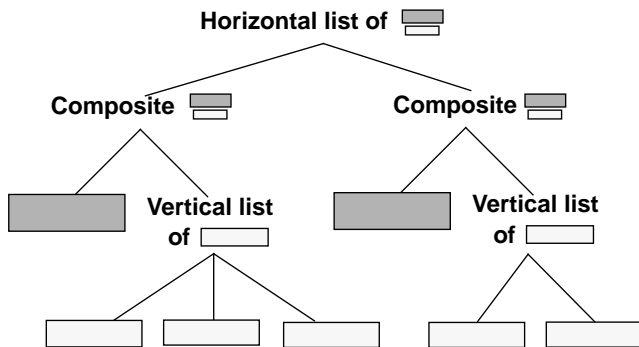


Figure 6: Spatial parse tree for sample layout

automatically assign them a type. This mapping of visual features to implicit object type is system dependent since different visual features are modifiable in different systems.

Using our survey examples as test cases, we found that this kind of parsing was fairly successful. At a low level, such as the labelled lists in Figure 1, the parser is quite accurate at determining what elements are part of the same structure; thus the small irregularities in alignment and spacing seem to be handled correctly. Ambiguous structures were not always identified as authors intended. For example, the fourth labelled list from the left in Figure 1 is identified as two separate lists (even though the author saw them as one) because of the large gap between the fragments.

Higher level structures were sometimes missed. For example, the six labelled lists in Figure 1 are parsed as two separate structures. In this example, the parser recognizes the first three lists as one structure, and the second three lists as another, because each of the lists on the left side of the diagram has an annotation directly above it. This is a second example of how layouts may also be ambiguous to humans without access to the semantics of the situation (if a reader did not realize the annotation functioned as such, he or she would make the same error as the parser). In most of our test cases, in spite of omissions and inaccuracies, the structure identified was often consistent with the authors' intent.

RECOGNITION ARCHITECTURE

We wanted to develop a flexible and easily extensible architecture that would allow us to tailor recognition performance to match different conventions, tasks, user preferences, and system-dictated layout constraints (such as gridding). As a result, the recognition architecture includes a reconfigurable pipeline of spatial structure *specialists* that share a blackboard of information and a recognition *strategist* that configures the specialists.

The strategist begins the recognition process by determining the order in which the pipeline of recognition specialists will be applied. This ordering is based on a statistical assessment of layout features. The specialists then begin a bottom-up parse of the layout; each specialist is responsible for identifying a particular type of structure. If the specialists define new types (as a result of encountering composites or heaps), they add these to the blackboard and recompute usage statistics to reflect the new structures. Figure 7 diagrams this process. The strategist, the blackboard, and recognition specialists are described in more detail below.

Strategist. The strategist performs some initial analysis of the overall space to determine the order in which the specialists will be applied. The strategist uses an initial calculation of overall object alignments to determine whether the arrangement seems to favor a vertical or horizontal orientation, or neither. According to these results, the specialists are ordered in the pipeline. (In the future work section, we discuss the potential for more complex or cooperating strategists.)

Blackboard. The blackboard contains global information shared by the specialists. This information includes user-

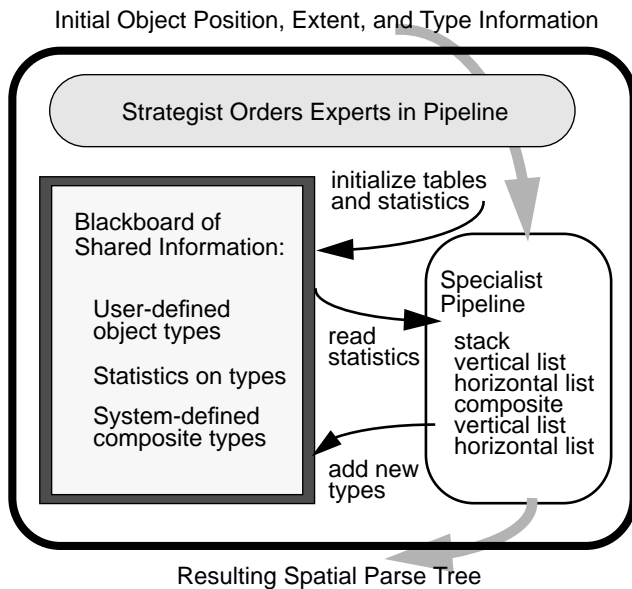


Figure 7: Architecture for spatial recognition

defined object types, system-defined composite types, and statistics on each type's use. Initial statistics are the average width and height for objects of a given type.

Specialists. Recognition of implicit structure is performed by a pipeline of computerized specialists. Each specialist is designed to recognize a single type of structure. Besides the information in the blackboard, the specialists are provided with the spatial display and its current parse. Because each specialist produces a new parse as output, the specialists can be pipelined together in any order and a single specialist can operate multiple times; this enables the recognition of complex structures like the horizontal list of composites used as an example in the last section.

To keep the algorithms system-independent, the specialists use only the visual and planar characteristics described earlier: object position, extent, and type. Specialists that identify heterogeneous structures (heaps of dissimilar objects and composites) generate new unique types for later specialists called by the parser; structures formed from homogeneous objects assume the type of their constituents. By repeated application of the recognition algorithms on parse results, higher levels of structure are recognized (like lists of composites, or composites which include lists).

SUPPORTING THE USER WITH RECOGNIZED STRUCTURE

Why is there a need to develop heuristic algorithms to find structure that is already perceived by the users?

First, they can help authors interact with ad hoc organization; the found structures can be used as the basis for supporting simple but repetitive information management tasks. Second, if a more formal knowledge base is a desired outcome of the task, recognizing structures is an important method for helping people notice and express the regular structure of their domain and maintain its consistency [15].

We are investigating these uses of the recognized structure in the context of VIKI [11]. With the addition of the spatial parser, VIKI assists authors in both interaction with ad-hoc structures and formalization of emerging structures. Before describing this support in detail we will give a brief overview of VIKI and its goals.

Brief Description of VIKI

VIKI is a spatial hypertext system that supports the emergent qualities of structure and the abstractions that guide its creation; the tool is intended for exploratory interpretation, making sense of a body of electronic information collected in service of a specific task such as analysis, design, or evaluation. Users manipulate graphical objects within a hierarchy of information workspaces (collections), using the visual properties of the objects along with a more formal types mechanism to express a highly nuanced interpretation of the collected materials.

VIKI gives users the ability to work with three kinds of elements: objects, collections, and composites. Each graphical *object* is a visual symbol that refers to an underlying piece of semi-structured information. Each *collection* is a subspace that can reflect semantic categorization or a task-oriented subset of the information space. *Composites* are two or more objects which are used together to make up a meaningful higher-level unit of structure. These three kinds of elements allow users to build up the same kinds of structures we observed in practice.

VIKI collections act as clipping regions so users can see information at multiple levels in the hierarchy of workspaces. Similar to Boxer [1], clicking on the border of a collection causes that collection to fill the VIKI window. Figure 8 shows before and after images of collection traversal in VIKI.

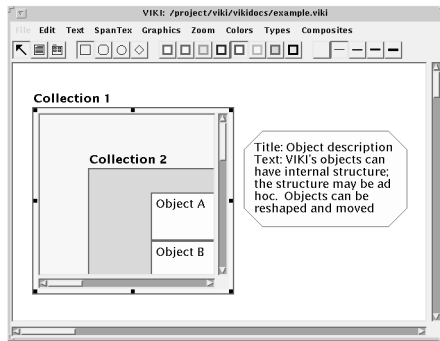
VIKI uses the results of spatial parsing two ways: it supports interaction based on implicit structure (where implicit structure remains undeclared) and it helps people use the object-collection-composite data model by supporting the transition from implicit to declared structure. Examples of each of these uses are described below.

Access to Implicit Structures

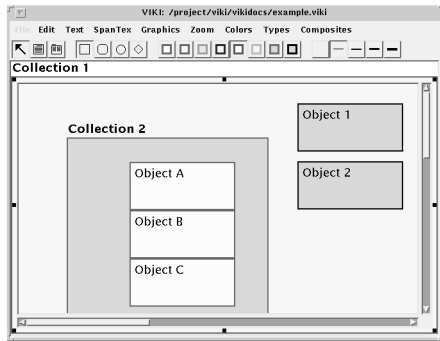
The spatial parser allow users to interact with implicit structure without requiring the structure to be formally defined or even anticipated. VIKI's hierarchic click-selection facility is a good example of this type of support.

Click-selection in VIKI works much the same way as it does in a text editor. In a text editor, a single click puts the cursor at a particular point; the next click selects the word; the next, the paragraph; the next the entire document; and the next returns to the single point of selection. VIKI uses a similar technique: each successive click selects the next level of hierarchical structure.

Figure 9 shows hierarchic click selection in action. The first click selects the individual object. The second click selects the list of three objects. The third adds the label for that list



a.) before traversal



b.) after traversal

Figure 8: Screen A shows a VIKI collection acting as a clipping region prior to traversal. Screen B shows the view after navigating into the collection.

and the fourth selects a similar labelled list above the first list.

Users realize two immediate advantages from hierarchic click-selection. First, users may select objects that are part of partially hidden structures without having to scroll or traverse from the current view. This is especially important in VIKI since collections act as clipping regions, displaying only portions of their contents. Thus, selecting a structure in one collection and moving it to another collection, a fairly frequent action, can be accelerated through use of click selection.

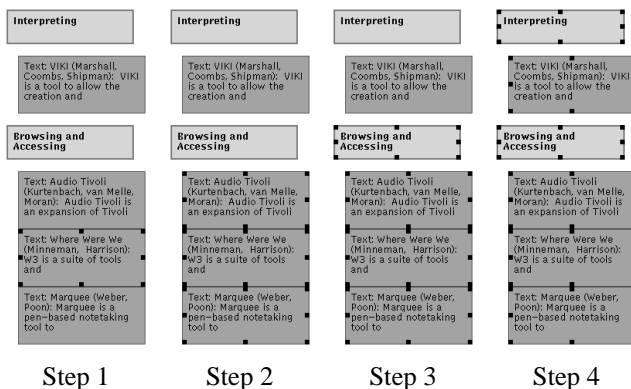


Figure 9: Four steps in VIKI's recognition-based hierarchic click-selection

Second, users may select partial structures in areas where objects are too densely packed for sweep-selection. In VIKI, such situations arise when users sort through large numbers of references to external documents--a frequent activity in the type of analysis tasks VIKI is designed to support.

In each of these cases, users tend to select objects this way for non-destructive operations, such as move, because the entire extent of the selection may not be visible.

Support for Creating Formal Structures

While undeclared structure can be the basis for certain kinds of interaction, formal or declared structure can facilitate other kinds of support.

Experiences with a variety of information management tools point to the difficulty users have in creating and using formal structure [16]. This difficulty led us to support a process of "incremental formalization", in which information is initially entered in an informal representation and later can be formalized when needed.

Incremental formalization aims, first of all, to eliminate the cognitive costs of formalization that inhibit user input. Secondly, it aims to reduce the burden of formalization by distributing it, and making it demand driven.

To further lower the cost of formalizing information, we are investigating techniques for using the recognized structure from the spatial parser to support incremental formalization. VIKI uses the results of the spatial parser to provide formalization suggestions to the user. This work builds on our experience with supporting incremental formalization based on the recognition of textual cues of inter-object relations in the Hyper-Object Substrate [17].

Collection Suggestion

By supporting incremental formalization VIKI helps users bridge the gap between their activities and the system's data model. For example, at a user's initiative, VIKI will suggest collections -- apparent subdivisions of materials for starting new subspaces. Collection suggestion is an accelerator: it greatly reduces author effort in creating new collections and moving existing, visually structured materials into them.

To suggest collections in VIKI, we look for the highest level of contiguous structure. These higher level structures correspond to the task-oriented workspaces we observed in our survey. In determining which top-level structures to suggest as collections, structures which greatly overlap in space (and thus would obscure one another if made into collections independently) are combined. We limit the number of extraneous small collections that VIKI suggests by requiring a minimum number of constituent objects.

Collection suggestion uses a standard spelling checker as a model of interaction. In our interface, shown in Figure 10, the user can iterate through the list of suggestions and accept those that are appropriate. VIKI displays these suggestions to the user by selecting all objects and collections that will become part of the new collection and outlining its extent

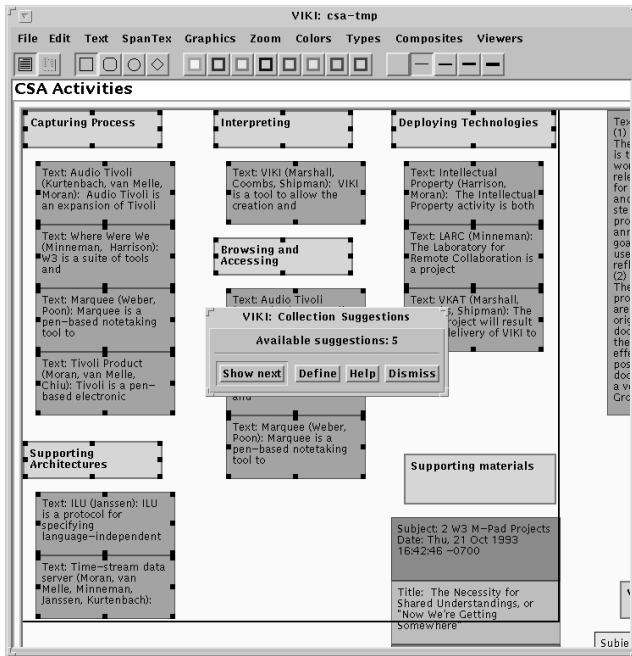


Figure 10: VIKI suggests a new collection based at the user's request

with a dark band. Users can modify suggestions by interactively changing which objects are selected and thus will be included in the new collection.

Composite Suggestion

Composite suggestion is an example of how VIKI can use recognition results to support not only the formalization of instances of relations, but also new classes of relations. The recurrent spatial patterns of object types that are identified by the spatial parser are used as suggestions for VIKI composite types.

Users can ask for VIKI to suggest potential composite types from within the composite definition dialog. Users may either accept the suggestions as is, modify them, or start from scratch to develop new composite types. Figure 11 shows the composite definition dialog with a suggested composite based on a recurrent pattern in the user's work. The suggestion appears as a set of abstract objects in an editable workspace with a composite name based on the names of the constituent objects and their arrangement.

RELATED WORK

Work on the recognition of spatial regularities has been a long-standing part of vision processing. But our work emphasizes the recognition of patterns in layouts of discrete, declared objects generated by purposeful human activities.

Document analysis and recognition shares our goal of identifying structures implicit in the layout of information. It differs, however, in its emphasis on the recognition of presentational structures common to known genres rather than identifying the more dynamic, idiosyncratic structures that evolve in the process of manipulating information

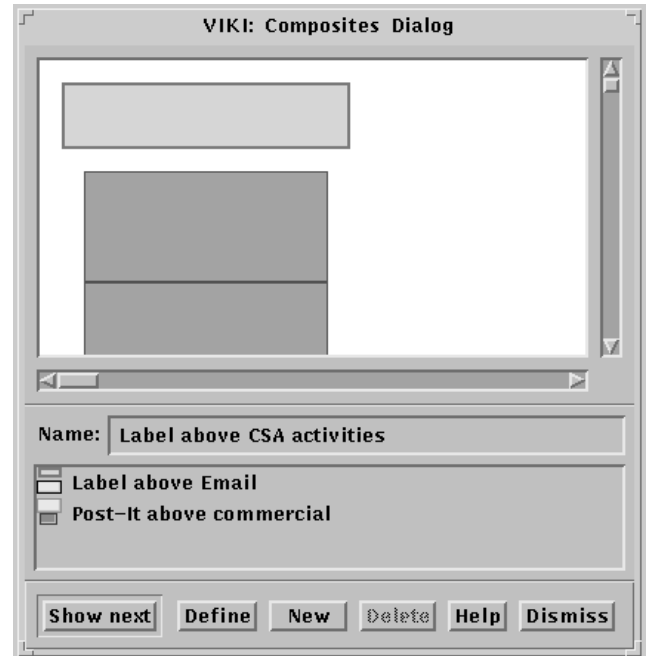


Figure 11: Composite definition dialog with suggestion based on the structure shown in Figure 10

objects. Thus, while some basic techniques may be shared, they are apt to diverge due to this crucial difference.

Spatial recognition and parsing is also found in work on visual languages, but has different goals and assumptions which influence the types of recognition algorithms produced. Unlike Lakin's visual language parsing in vmacs [5], our algorithms do not assume that we can unambiguously recover their underlying syntactic structure. Unlike Pictorial Janus [3], a visual programming environment that bases connectivity on assessments of inside, connected, or touching, our purpose is not to "debug" formal visual/spatial structures, but rather to tease out some implicit partially-framed structure.

Our goal of providing better support for users' already apparent desire to work with implicit structure has influenced the spatial parser's design from the start. Saund's perceptually-supported sketch editor [13] and Moran's support using implicit structure in Tivoli [12], although focussing on recognition in pen-based sketches, are based on similar goals.

FUTURE WORK

This work may proceed in several different directions to improve our understanding of the recognition and use of implicit spatial structure.

From our experience, it is apparent that new specialists are required for some kinds of recognition to proceed. Outlines, matrices and tables were found in our survey but are not parsed well by the current implementation.

Because of the idiosyncratic and ambiguous nature of the spatial structure, in cases where there is a high overhead for

incorrect parses, interaction will be important in guiding this kind of recognition. Up to now, we have emphasized uses of recognized structure that are tolerant to inaccurate parsing; uses where the results are lightweight and used to provide alternatives for actions already possible through other means. Future work on interaction methods for correcting incorrect parses would be required before error-intolerant uses of recognized structure can be evaluated in real-use situations.

More generally, experience with enabling and supporting incremental formalization in VIKI shows promising results. This work leads to more issues to be explored in both methods of producing and interfaces to suggesting possible formalizations. We need more experience before we can answer questions like when suggestions should be provided to the user and what good interfaces are for providing such suggestions.

CONCLUSIONS

Our survey showed that people use the visual and spatial characteristics of graphical layouts to express relationships between objects, icons, or other representations of underlying information. We discovered a small set of underlying primitives that capture the conventions illustrated by the layouts in our survey and implemented a set of heuristic algorithms to recognize these common structures. Because these structures are readily perceived by people, but remain implicit (and therefore opaque) to the systems in which they were created, we postulated that they would be a good basis for providing new system services.

Our extensible, tailorable architecture for performing spatial structure recognition uses a pipeline of specialists, configured by a strategist, that communicate through a blackboard. The specialists, which may be reapplied as many times as necessary, perform a bottom-up parse of a spatial layout to identify multiple levels of implicit structure. This architecture not only allows new heuristics to be added to the spatial parser, but also allows the parser to be modified to work with different systems or tools.

We have demonstrated the utility of this approach by integrating the parser with VIKI, a spatial hypertext system. In VIKI, we use spatial structure recognition as the basis for user interaction with implicit forms and as a means of suggesting more formal representations to the user at his or her initiative. Already the parser-based support provides advantages for VIKI users; as use continues, we believe that we will be able to enhance both the algorithms, the way they are configured, and the kinds of system features we can provide based on the results of their analysis.

ACKNOWLEDGEMENTS

We would like to thank Frank Halasz and Jim Coombs, who aided in the development of the ideas and systems described in this paper. This work was done while at Xerox PARC.

REFERENCES

[1] diSessa, A., Abelson, H. "Boxer: A Reconstructible Computational Medium." *Communications of the ACM*, 29, 9, (1986), pp. 859-868.

- [2] Halasz, F.G., Moran, T.P., Trigg, R.H. "NoteCards in a Nutshell." *Proc. of the ACM CHI + GI Conference*, (Apr. 5-9, 1987), pp. 45-52.
- [3] Kahn, K. "Concurrent constraint programs to parse and animate pictures of concurrent constraint programs." *Proc. of the International Conference on Fifth Generation Computer Systems*, June 1992.
- [4] Kidd, A. "The Marks are on the Knowledge Worker." *Proc. of CHI '94*, (Apr. 24-28, 1994), pp. 186-191.
- [5] Lakin, F. "Visual Grammars for Visual Languages." *Proc. of AAAI '87*, (July, 1987), pp. 683-688.
- [6] Malone, T.W. "How do People Organize their Desks? Implications for the Design of Office Information Systems." *ACM Transactions on Office Information Systems*, 1, 1 (Jan., 1983), pp. 99-112
- [7] Mander, R., Salomon, G., Wong, Y.Y. "A 'Pile' Metaphor for Supporting Organization of Information." *Proc. of CHI '92*, (May 3-7, 1992), pp. 627-634.
- [8] Marshall, C.C., Halasz, F.G., Rogers, R.A., Janssen, W.C., Jr. "Aquanet: a hypertext tool to hold your knowledge in place." *Proceedings of Hypertext '91*, (Dec. 16-18, 1991), pp. 261-275.
- [9] Marshall, C.C., Rogers, R.A. "Two Years before the Mist: Experiences with Aquanet." *Proc. of ECHI '92*, (Dec. 1-4, 1992), pp. 53-62.
- [10] Marshall, C.C., Shipman, F. M. "Searching for the Missing Link: Discovering Implicit Structure in Spatial Hypertext." *Proc. of Hypertext '93*, (Nov. 14-18, 1993), pp. 217-230.
- [11] Marshall, C.C., Shipman, F. M., Coombs, J.H. "VIKI: Spatial Hypertext Supporting Emergent Structure." *Proc. of ECHI '94*, (Sept. 18-23, 1994), pp. 13-23.
- [12] Moran, T.P., Chiu., P., vanMelle, B., Kurtenbach, G. "Implicit structures for pen-based systems within a freeform interaction paradigm." *Proc. of CHI '95*, (May 7-11, 1995).
- [13] Saund, E., Moran, T.P. "A Perceptually-Supported Sketch Editor." *Proc. of UIST '94*, 1994.
- [14] Shipman, F.M., Chaney, R.J., Gorry, G.A. "Distributed Hypertext for Collaborative Research: The Virtual Notebook System." *Proc. of Hypertext '89*, (Nov. 5-8, 1989), pp. 129-135.
- [15] Shipman, F.M. *Supporting Knowledge-Base Evolution with Incremental Formalization*. Ph.D. Dissertation, Technical Report CU-CS-658-93, Dept. of Computer Science, Univ. of Colorado, 1993.
- [16] Shipman, F.M., Marshall, C.C. "Formality Considered Harmful: Experiences, Emerging Themes, and Directions." Xerox PARC technical report ISTL-CSA-94-08-02, 1994.
- [17] Shipman, F.M., McCall, R. "Supporting Knowledge-Base Evolution with Incremental Formalization." *Proc. of CHI '94*, (Apr. 24-28, 1994), pp. 285-291.