Design Space Navigation: An Annotated Bibliography

Sheng-Fen Chien and Ulrich Flemming

EDRC 48-37-96

Design Space Navigation:
an Annotated Bibliography

Sheng-Fen Chien, Ulrich Flemming

Department of Architecture
and Engineering Design Research Center
Carnegie Mellon University
Pittsburg, PA 15213

March 7, 1996

Abstract. SEED, a Software Environment to Support the Early Phases in Building Design, allows designers to explore conceptual alternatives and to iterate over design concepts in their search for a most promising design scheme. But designers using SEED can easily get lost in the multiple *design spaces* they are able to generate and therefore need intelligent navigation aids. We review in this report the literature related to information navigation in electronic media, including the small subset of sources dealing specifically with *design space* navigation. The sources listed in the report cover two main subject areas: *information navigation* and *information visualization*.

This work has been supported by the U.S. Army Corps of Engineers Construction Engineering Research Center Laboratory and the Engineering Design Research Center, an NSF supported Engineering Research Center.

Contents

Introduction

SEED, a Software Environment to Support the Early Phases in Building Design, intends to encourage designers to explore conceptual alternatives and to iterate over design concepts in order to derive at a most promising design scheme. It supports the rapid generation of computable representations of alternatives and variants of alternatives based on explicitly stated requirements. Designers may modify dynamically requirements as their perception of the problem at hand evolves through work with SEED. They receive consistent feedback as SEED evaluates the current design after each design decision against the current requirements[1].

Designers work incrementally by starting with more abstract design versions, which they successively refine, or by adding pieces to a design at a given level of abstraction. Designers are able to pursue alternative paths, to return to temporarily abandoned design versions at a later time in the same design session; or to store any version persistently in a database, from which it can be retrieved later either by an explicit call or by using *case-based* search techniques.

The representation used by all parts of SEED is object-based, that is, both designs and requirements are represented as collections of related objects, which—in turn—are instances of classes from which they inherit attributes and behavior. Problems (collections of requirements) and their solutions are first of all distinguished in SEED by *type*. For example, the problem of laying out the spaces on a floor is distinct, on the one hand, from the problem of determining which spaces should be placed on that floor and, on the other hand, from the problem of placing partitions between the spaces on the floor. SEED offers specific solution generation operators for different types of problems. In the incremental design process supported by SEED, an object representing a problem or solution is derived from another object of the same class; that is, parent/child relations exist between problems on the one hand and solutions on the other hand. A problem and the solutions generated for it constitute a *design space*. The (intermediate or complete) solutions in this space are its *states*, which are related through parent/child relations. Design states can be nested within design states if they refine part of a solution generated at a more abstract level. For example, the problem of laying out the spaces on a floor refines a state that determines which spaces go on which floor.

It has been clear to the SEED developers from the outset and confirmed through experiments with potential users that designers using SEED can easily get lost in the multiple design spaces they are able to generate and populate with a potentially very large number of objects. It is common to call the features that distinguish nested design spaces and the states in any one of the subspaces the *dimensions* of the spaces. Users *navigate* this space by moving along any dimension, for example, by moving from a higher abstraction level to a lower one or vice versa; by revisiting states in a space; or by expanding a space through the generation of additional states or the modification of problems. Users need intelligent navigation aids if they are not to get lost on the way.

We review in this report the literature related to design space navigation. The report is self-contained; that is, it may be of interest to readers concerned with issues that differ somewhat from those encountered by the SEED developers. However, we do point out if a source sheds light on an issue of specific importance in the SEED context.

The sources listed in the report cover two main subject areas: *information navigation* and *information visualization*. Information navigation is further subdivided into the following subareas:

- Cognitive mapping: the cognitive theory of how humans find their way;
- Empirical studies of human cognitive processes during the execution of navigation tasks, such as navigating in a file system directory space or in a library database;
- Navigation frameworks: the overall concepts and/or metaphors employed in systems that support information space navigation, where the term *information space* is a metaphor and not to

[1] For an overview of SEED, see: Flemming, U. and R. Woodbury. “Software environment to support early phases in building design (SEED): Overview” Journal of Architectural Engineering 1 (December 1995), 147-152.

be confused with space in the physical world;

- Direct manipulation of visible information: a technique popular in the field of human-computer interaction that allows users to interact directly with data in an information space;
- Query, search and filter: utilities that allow users to focus on the data of interest in an information space.

Information visualization contains the following subareas:

- Visualization frameworks: overall concepts and systems set up to visualize an abstract information space;
- Visualizing hierarchical and other relational structures such as trees or networks;
- Visualizing multi-dimensional information, that is, information that can be categorized and retrieved according to multiple aspects or features;
- Multiscale displays: alternative views into the same information space with different resolutions of its data;
- Translucent / transparent displays: stacking multiple views in a limited screen area using translucent or transparent displays;
- 3D displays / effects: visualization using 3D objects or effects (such as perspective views);
- Visual metaphors: metaphors employed by systems to assist the visualization of information.

A source may be multiply listed if it covers several of the areas mentioned above. The selected sources cover research and applications in human-computer interaction, database management, hypermedia systems, World-Wide Web (WWW) search, computer graphics, and other areas related to information management. This review is by no means complete. However, the selected sources are state-of-the-art and representative of the areas they cover.

Information Navigation

Cognitive Mapping

Cognitive mapping has been the primary theory of human spatial cognition proposed in the literature. Three main cognitive mapping strategies have been identified: egocentric, fixed, and abstract systems of reference (see Hart and Berzok 1982). Researchers have demonstrated that both children (Down 1985; Hart and Berzok 1982; Mandler 1988; Pick and Rieser 1982; Somerville and Haake 1985) and adults (Gärling, Böök and Lindberg 1985; Mandler 1988) use these three strategies in various situations. They construct through these different strategies different cognitive maps. Lynch (1960) observes that a person may have different images of a city and that these images are "arranged in a series of levels, roughly by the scale of area involved, so that the observer moved as necessary from an image at street level to levels of a neighborhood, a city, a metropolitan region" (Lynch 1960, p. 86). Tversky (1993) refers to these images as cognitive collages which "are thematic overlays of multimedia from different points of views." (Tversky 1993, p. 15) Furthermore, landmarks are the essential elements in human cognitive maps or collages (Down 1985; Gärling, Böök and Lindberg 1985; Golbeck 1985; Hart and Berzok 1982; Lynch 1960; Mandler 1988; Pick and Rieser 1982; Somerville and Haake 1985; Tversky 1993); they "facilitate the encoding and retrieval of information about spatial location." (Golbeck 1985)

There appear to be clear distinctions between spatial maps of elements found in the physical environment and data in an information space; in the physical world, the landmarks stand out from a surrounding *ground*, which has structure of its own and is able, for example, to relate landmarks geographically to each other, not only in terms of proximity, but also in terms of more absolute directions (north / south etc.) so that people who know where they are in this ground can orient themselves generally in space. Information spaces have no such ground; that is, the pieces of information 'float' around all by themselves with no mediating 'matter'. We have not found any discussion of these differences in the literature and would be hesitant to apply findings from the realm of cognitive spatial maps directly to information space navigation. However, we list the following sources because we find them suggestive. For example, design space navigation could incorporate landmarks as general orientation points, but may have to allow users to select their own landmarks since—unlike in physical space—these landmarks are not preexisting. As another example, a navigation tool may be able to display design spaces at arbitrary levels of details and allow users to identify / store their abstraction settings. This would enable users to control the associations between their own cognitive maps and abstract views of design spaces.

Downs, R. M. "The representation of space: its development in children and in cartography." The Development of Spatial Cognition. Ed. Cohen, R. Hillsdale, NJ: Lawrence Erlbaum Associates, 1985. 323-345.

This paper presents a theory of the representation of spatial information in human cognition and relates its development to the evolution of map-making (cartography).

Gärling, T., A. Böök and E. Lindberg. "Adults' memory representations of the spatial properties of their everyday physical environment." The Development of Spatial Cognition. Ed. Cohen, R. Hillsdale, NJ: Lawrence Erlbaum Associates, 1985. 141-184.

The authors present an information processing model of adults' spatial memory. In this model, properties of the environment are cognitivly mapped. The mapped properties are places, spatial relations (spatial inclusion, metric spatial relations, proximity), and travel plans.

Golbeck, S. L. “Spatial cognition as a function of environmental characteristics.” The Development of Spatial Cognition. Ed. Cohen, R. Hillsdale, NJ: Lawrence Erlbaum Associates, 1985. 225-255.

The authors describe a list of environmental characteristics that influence spatial cognition. These characteristics are classified into two broad categories: structural features (landmark, containing feature), and organizational features (clustering, orientation, saliency).

Hart, R. and M. Berzok. “Children's strategies for mapping the geographic-scale environment.” Spatial Abilities: development and physiological foundations. Ed. Potegal, M. New York, NY: Academic Press, 1982. 147-169.

This paper describes three cognitive mapping strategies: egocentric (myself), fixed (landmarks), and abstract systems of reference (roads, orientations). Two broad categories of way-finding strategies are identified: sequential strategies (ordinal mapping, interval mapping, and accurate route mapping), and survey strategies (loose typological mappings, accurate simultaneous display strategies (like a real map)).

Lynch, K. The Image of the City. Cambridge, MA: The Technology Press & Harvard University Press, 1960.

Lynch introduces paths, edges (boundaries), districts, nodes (points, spots which observers can enter), and landmarks (a type of point reference) as image-forming elements in cities. Landmarks can be elements as different as a tower, a hill, a signpost, a tree, or a doorknob. A landmark can be easily identified if it has a clear form, contrasts with its background, or has a prominent spatial location.

The global image of the environment is a set of images, which more or less overlap and are interrelated. “They were typically arranged in a series of levels, roughly by the scale of area involved, so that the observer moved as necessary from an image at street level to levels of a neighborhood, a city, a metropolitan region.” (p.86)

The author refers to Warner Brown's study[1] of human way-finding behavior: “... subjects seemed to use at least three different kinds of orientation: a memorization of the sequence of movements, usually difficult to reconstruct except in correct sequence; a set of landmarks (rough boards, sound sources, rays of sunlight that gave warmth) which identified localities; and a general sense of orientation in the room space (for example, the solution might be imaged as a general movement around the four sides of the room, with two excursions into the interiors)” (p.131).

Mandler, J. M. “The development of spatial cognition: on typological and Euclidean representation.” Spatial Cognition: Brain Bases and Development. Eds. Stiles-Davis, J., M. Kritchevsky and U. Bellugi. Hillsdale, NJ: Lawrence Erlbaum Associates, 1988. 423-432.

Mandler identifies the importance of landmarks in human spatial cognition. The author observes that “… spatial primitives of landmarks, containment, support relations, and so forth seem to be basic at all ages, with metric information secondary. … Landmarks play important and distorting roles in the formation of cognitive maps at all ages. … In short, a number of non-Euclidean properties seem to be the most essential aspects of spatial knowledge at all ages. We code things as inside or outside of regions, nearby landmarks, on or under other things. Typically we do not process metric details because they are unnecessary for our usual commerce with the environment.” (p.430-431)

[1] Brown, W. “Spatial Integrations in a Human Maze.” University of California Publications in Psychology, V.5 (1932): 123-134.

Pick, H. L. and J. J. Rieser. "Children's cognitive mapping." Spatial Abilities: Development and Physiological Foundations. Ed. Potegal, M. New York, NY: Academic Press, 1982. 107-128.

The paper describes how Pulawat Islanders navigate in the Pacific Ocean. It again demonstrates the use of landmarks, reference systems, and spatial inference during human spatial tasks. The authors argue that the cognitive mapping process includes spatial inferences about relations between locations, as well as updates of the relation between one's own position and all other locations in space. They emphasize the process of 'mapping' rather than the static cognitive 'map' during human spatial tasks.

Somerville, S. C. and R. J. Haake. "The Logical Search Skills of Infants and Young Children." Children's Searching: the Development of Search Skill and Spatial Representation. Ed. Wellman, H. M. Hillsdale, NY: Lawrence Erlbaum Associates, 1985. 73-104.

This study demonstrates that children understand spatio-temporal sequences and know how to use them for searching purposes.

Tversky, B. "Cognitive maps, cognitive collages, and spatial mental models." COSIT '93, Marciana Marina, Elba Island, Italy, September 1993. Eds. Frank, A. U. and I. Campari. Springer-Verlag, 1993. 14-24.

The author proposes the spatial cognitive model of 'cognitive collages' and argues that it fits spatial behavior better than cognitive maps.

West, R. L. and C. W. Morris. "Spatial cognition on nonspatial tasks: finding spatial knowledge when you're not looking for it." The Development of Spatial Cognition. Ed. Cohen, R. Hillsdale, NJ: Lawrence Erlbaum Associates, 1985. 13-39.

This paper tries to demonstrate the usefulness of spatial representations (spatial visualization, spatial thinking) in scientific reasoning and many other domains.

Empirical Studies

The behavior of people navigating in information spaces has been observed by Akın, Baykan and Rao (1987) and O'Day and Jeffries (1993). These studies introduce a new kind of navigation tool, search, that is available in information spaces, but not in our physical environment. Search may be able to aid way-finding in design spaces while posing a minimal memory load on the user.

Akın, Ö., C. Baykan and D. R. Rao. "Structure of a directory space: a case study with a UNIX operating system." International Journal of Man-Machine Studies, 26 (1987): 361-382.

The paper presents an empirical study of the structure of directory spaces and of users' search behaviors in a UNIX operating system. Findings indicate that most directories are organized hierarchically, but with few levels. Furthermore, depth-first search characterizes both the organization of the directories and the behavior of the users. One of the findings, "users like to organize files by topic," indicates a possible organization of states in a design space. Several suggestions towards improving the UNIX operating system apply also to design space navigation, for example, providing tools for searching and retrieving desired files based on user-defined criteria.

O'Day, V. and R. Jeffries. "Orienteering in an information landscape: how information seekers get from here to there." INTERCHI '93 Conference on Human Factors in Computing Systems:

Bridges Between Worlds, Amsterdam, The Netherlands. Eds. Ashlund, S., K. Mullet, A. Henderson, E. Hollnagel and T. White. Addison Wesley, 1993. 438-445.

The authors present an empirical study of the uses regular clients of professional intermediaries (such as librarians) make of information retrieved by search. Tasks are performed for three different types of searches: (1) monitoring a well-known topic or set of variables over time; (2) following an information-gathering plan suggested by a typical approach to the task at hand; and (3) exploring a topic in an undirected fashion. The study identifies a set of common triggers and stop conditions for further search steps along with a set of common operations that clients use to analyze search results.

Navigation Frameworks

Navigation issues have been studied early on in connection with hypertext systems (see Balasubramanian (1994) for a review of several hypertext systems) and more recently with the WWW environment. In summary, four types of navigation frameworks have been discussed:

- Hyperlinks: navigation through sensitive objects in an information space, where the information may or may not be structured; WWW browsers (such as Netscape or Mosaic), KMS (Akscyn, McCracken and Yoder 1988), and InfoGrid (Rao et al. 1992) are of this type;
- Hierarchies or networks: navigation through nodes in an information space, where the information is organized into hierarchies (e.g., Rivlin, Botafogo and Shneiderman 1994) or networks (e.g., Thüring, Haake and Hannemann 1991) and views of the overall or partial information space are provided;
- Portals or wormholes: navigation through interactive zooming and panning in an information space; the zoom operation can activate a view (portal) which displays the selected information at different scales or in different formats (Perlin and Fox 1993), or a view (through wormhole) that displays additional data related to the selected information (Woodruff et al. 1994).
- Rooms (Clarkson 1991): navigation through rooms (information clusters) in a information space modeled as the interior of a physical space with rooms and doors displayed in 3D or 2D views.

Hierarchies are used in the current SEED-Layout implementation because design spaces in SEED-Layout are hierarchically organized through derivation relationships. For the general design space navigation problems in SEED, we may consider portals or wormholes, which provide flexible interactions at the overview level (where users see the structure of the design spaces) as well as the object content level (where users work in views of design states in a design space). In addition, this framework provides the capability to incorporate search mechanisms.

Akscyn, R. M., D. L. McCracken and E. A. Yoder. "KMS: a distributed hypermedia system for managing knowledge in organizations." Communications of the ACM, 31.7 (1988): 820-835.

KMS supports browsing in a hypertext-like environment using (1) a standard frame layout—its relative homogeneity makes it easier for users to perceive the components of a frame, interpret them, and make decision about what to do next, (2) large target selection, (3) fast backtrack commands, but no scrolling.

Studies of ZOG (the predecessor of KMS) showed that the multi-node views that are available in ZOG were rarely used. KMS, in contrast, provides only a breadth-first view of a frame, and the frame itself provides a bread-first view of the database.

KMS prevents disorientation by providing (1) hierarchical selection (hierarchical links are distinguished from cross-reference links), (2) special navigation commands, e.g. Goto, Home, Info, (3) flagging previous selections, and (4) fast response. It also supports annotation (as cross-reference) and search.

Balasubramanian, V. “State of the art review on hypermedia issues and applications.” Graduate School of Management, Rutgers University, Newark, New Jersey, 1994. <http://www.isg.sfu.ca/~duchier/misc/hypertext_review/index.html>.

This report consists of eight chapters dealing, respectively, with an introduction to hypertext; implementation issues; database issues; user interface issues; information retrieval issues; integration issues; applications; and a systematic approach to user interface design for a hypertext framework. The following chapters are the most interesting for our purposes.

Chapter 1 reviews several hypertext systems, such as Memex[1] (which allows for scanned-in handwritten marginal notes and comments), Xanadu[2] (a write-once system: once something is published, it is for the entire world to see forever; as links are created by users, the original document remains the same except for the fact that a newer version is created with references to the original versions), Intermedia[3] (provides three types of navigation tools: paths, maps and scope links), NoteCards[4] (contains four basic constructs: notecards that contain information contents such as text, graphics, images, voice, etc.; links that present binary relationships between cards; browsers that display node-link diagrams of portions of the network; and fileboxes that organize cards into topics or categories), KMS (see Akscyn, McCracken and Yoder (1988) on page 6.), and Guide[5] (supports four different kinds of links: replacement buttons, note buttons, reference buttons, and command buttons; navigation through replacement buttons initially provides a summary of the information; the degree of detail can be changed by the reader.)

Chapter 3 discusses efficient search and query mechanisms that could partly solve the “lost in space” problem experienced by users during navigations. There can be two kinds of queries: a structure query to retrieve a part of the network, and a content query to retrieve a specific node.

Chapter 4 summarizes several navigation approaches in hypertext systems:

1. Graphical browsers: help reduce disorientation by providing a 2D spatial display of the hypertext network. A drawback occurs when the network becomes too large.
2. Web views: global maps, local maps (focusing on the document of interest and its neighbors; which is most useful for WWW browsing), and local tracking maps. These web views are later improved to a path, a map, and a scope line. The combination of these three provides the spatial and temporal information.
3. Maps and overview diagrams (which seem to be the same as the web views in 2!).
4. Paths and trails: a path is a list of documents that the user has visited; it can be used to collect all interesting documents to form a single linear document.
5. Guided tours and tabletops: they are controlled navigation tools.
6. Backtracking, history lists, timestamps, and footprints, which are similar to paths and trails.
7. Arbitrary jumps, landmarks, and bookmarks
8. Embedded menus, which allow the user to select a word or item embedded within the text of a document and present the user with a list of links to go to.

[1] Nyce, J.M. and P. Kahn. Eds. From Memex to Hypertext: Vannevar Bush and the Mind's Machine. Boston, MA, Academic Press, 1991.

[2] Nelson, T. “Replacing the Printed Word: a Complete Literary System.” Information Processing '80. 1980.

[3] Meyrowitz, N.K. “Intermedia: the Architecture and Construction of an Object-Oriented Hypermedia System and Applications Framework.” OOPSLA '86. 1986.

[4] Halasz, F.G. “NoteCards: a Multimedia Idea Processing Environment.” Interactive Multimedia. Eds. S. Ambron and K. Hooper. Microsoft Press, 1988.

[5] Brown, P.J. “Turning Ideas into Products: the Guide System.” Hypertext '87, November 1987.

9. Fisheye views and 'spiders': An example of spiders, is the directed graph browser in Thoth-II[1] where a global map is created dynamically and expands in two-dimensional space creating spiders on the display as a user browses through linked nodes.

10. 'Roam' and zoom: Roaming is similar to conventional panning and scrolling operations, which provide users with a wire-frame box or rectangle to be dragged around the map window; they can thus roam around the information space (See “Examples of movable filters” on page 13.)

11. Conceptual space navigation[2]: useful for learning systems.

Chapter 5 argues that an effective access to information equals navigational access plus queries.

Clarkson, M. A. “An easier interface.” Byte. 2, 1991. 277-282.

The Information Visualizer provides 3D 'rooms' (see below) as multiple virtual workspaces, interactive objects for different visualization methods (such as ’cone trees’ or ’perspective walls’), and 3D navigation and manipulation (such as walking, touching, changing rooms and picking up objects). These elements as well as the application communicate through a 'cognitive coprocessor', which supports smooth animation and multiple asynchronous agents that present results to the user.

A 'room' is defined by a task-oriented clustering of information. Rooms are connected by 'doors'. “… Work is distributed throughout a collection of 3D / 2D rooms furnished with interactive objects such as walls and floating trees. To better understand the structure of the information that these objects represent, you can access embedded data and examine its structure from different angles by 'flying' around or through it.” Other operators on 3D trees are rotate, examine, prune, rearrange, etc.

“… These animated visualizations are designed to shift work to your perceptual system, freeing the conscious mind to work on larger problems.” Several filtering methods are introduced including abstraction and selective omission.

Mamrak, S. A. and C. S. O'Connell. “A framework for a comprehensive analysis of navigation in hierarchy topologies.” Technical Report, OSU-CISRC-4/91-TR12, Department of Computer and Information Science, The Ohio State University, Columbus, Ohio 43210, 1991.

The paper discusses three navigation strategies in a hierarchy topology: absolute, by name, and relative. Each of those can make use of three navigation mechanisms:

1. Basic: navigation strategies implemented in a direct, straightforward manner; an example of a basic absolute strategy in SEED would be the point-click method used in the problem hierarchy and design windows; the navigation buttons in the problem statement and design windows are examples of a basic relative strategy.

2. Enhanced: multiple views such as fisheye, overview and filtering; focusing; and memory support such as backtracking.

3. Redundant: different ways to do the same thing, such as the Home command.

The authors recommend the following guidelines for hyperbase software design:

1. Provide explicit navigation mechanisms;

2. Provide at least basic mechanisms for all the strategies;

[1] Collier, G. H. “Thoth-II: Hypertext with Explicit Semantics.” Proceedings of Hypertext '87 Conference, November 1987.

[2] Mayes, T., M. Kibby and T. Anderson. “Learning about Learning from Hypertext.” Designing Hypermedia for Learning, NATO ASI Series, Volume F67. Springer Verlag. 1990.

3. Provide enhanced and redundant mechanisms when resources allow for this.

The implementation should follow four guidelines: typographical cueing, redundancy, backtracking, and focus viewing. Examples from manuscript and mathematical applications are provided.

Perlin, K. and D. Fox. "Pad: an alternative approach to the computer interface." SIGGRAPH '93, Anaheim, CA. ACM Press, 1993. 57-64.

Pad is an infinite two-dimensional information plane. Objects are organized on that plane geographically: every object occupies a well-defined region on the plane's surface. 'Portals' are used for navigation; they act like magnifying glasses that can peer into and roam over different parts of the Pad surface. The screen itself is just a special 'root' portal.

Navigation is performed by zooming in and out and by moving around the surface. The underlying assumption for search is that all the information is there; to see more detail, you just have to take a closer look.

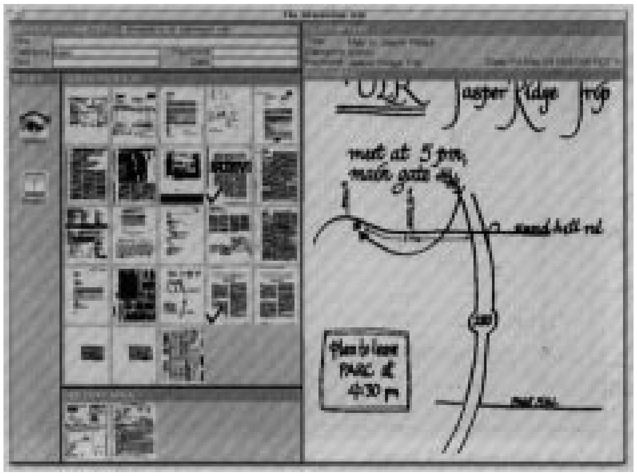
To facilitate the display, two techniques are used: 'semantic zooming' and 'portal filters'. Semantic zooming can be used to control the amounts or types of information to be displayed at certain scales. Portal filters control how information is displayed (e.g., in textual, tabular, chart or graphical format).

This navigation approach resembles closely that of the Tioga system (see Woodruff et al. (1994) on page 10).

Plaisant, C., D. Carr and B. Shneiderman. "Image browsers: taxonomy, guidelines, and informal specifications." Technical Report, CAR-TR-712/CS-TR-3282/ISR-TR-94-39, Human Computer Interaction Laboratory, University of Maryland, College Park, MD 20742, 1994.

This is a very comprehensive report on image browsing of objects in 2D space.

Rao, R., S. K. Card, H. D. Jellinek, J. D. Mackinlay and G. G. Robertson. "The Information Grid: a framework for information retrieval and retrieval-centered applications." UIST '92, Monterey, California. ACM Press, 1992. 23-32.



InfoGrid example: electric file cabinet

(Source: the original paper; reproduced with permission.)

The Information Grid (InfoGrid) is a framework for building information access (retrieval-centered) applications. It is based on the 'triple agent model' (developed at the Xerox Palo Alto

Research Center) consisting of the 'user', a 'dialogue machine', and a 'task machine'. Its domain is document services. The InfoGrid interface is divided into tiled areas interacting with the information access mechanism and document services provided by a particular application. These areas contain two kinds of objects: (1) 'property sheets', which display field-oriented information, and (2) InfoGrid buttons, which are objects that the user can manipulate. Some example InfoGrid applications are described, such as an 'electronic file cabinet,' a biography database, and an encyclopedia browser.

Rivlin, E., R. Botafogo and B. Shneiderman. “Navigating in hyperspace: design a structure-based toolbox.” Communications of the ACM, 37.2 (1994): 87-96.

The authors propose to organize hypertext links into hierarchical structures and provide algorithms to determine the root of a hierarchical structure.

Thüring, M., J. M. Haake and J. Hannemann. “What's Eliza doing in the Chinese Room?: incoherent hyperdocuments–and how to avoid them.” Technical Report, Arbeitspapiere der GMD 533, Gesellschaft für Mathematik und Datenverarbeitung MGH, 1991.

The authors claim that hyperdocument users fail to develop an understanding of the document as a “coherent entity” of closely related facts. They use it as an aggregation of loosely linked chunks of information because (1) inappropriately labeled nodes and links impede the understanding of semantic relations between nodes; and (2) the lack of higher-order units, which can represent part structures of the hypertext net, prevents the identification of different node contents. The report proposes a framework consisting of three parts: content (formed by 'design objects' which represent the information in the document), organization (represented by another type of design objects which encode relationships between documents), and presentation (different “views”).

Woodruff, A., P. Wisnovsky, C. Taylor, M. Stonebraker, C. Paxson, J. Chen and A. Aiken. “Zooming and tunneling in Tioga: supporting navigation in multidimensional space.” Technical Report, UCB//S2K-94-48, University of Berkeley, March 1994. <ftp://s2k-ftp.cs.berkeley.edu/pub/tech-reports/s2k/s2k-94-48/>.

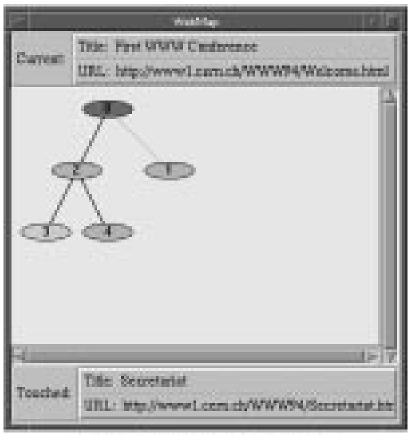
The authors propose the use of 'wormholes' to access spaces in different dimensions, an intuitively appealing approach. However, objects on either side of the wormhole are unrelated to each other, which makes this approach less applicable to navigation problems in design spaces because any two connected objects in a design space have some causal relation. Details, such as their 'elevation map' and 'elevation range', may be useful for design window navigations in SEED.

Direct Manipulation of Visible Information

Design space navigation tools should support direct manipulation / interaction with objects in design spaces, as well as facilitate view manipulations of these spaces, such as changing how the information is presented in a space (Dömel 1994), adding / removing information in the view (Gedy and Katz 1987; Henry and Hudson 1991), or changing aspects of views (Dömel 1994; Henry and Hudson 1991), e.g., by changing viewing axes (Hovestadt 1993)).

Dömel, P. “Webmap–a graphical hypertext navigation tool.” Electronic Proceedings of fhe Second International World-Wide Web Conference '94: Mosaic and the Web, Chicago, Fall 1994. <http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Searching/doemel/www-fall94.html>

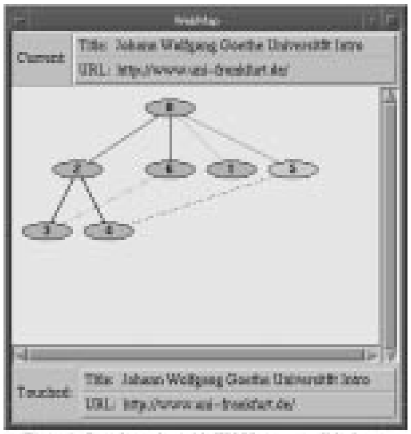
Webmap uses spanning trees as navigation aid. The paper includes algorithms and strategies for tree layouts and illustrates features and operations for navigation. The following navigation operations are provided by Webmap: next page, previous page, up one hierarchy level, and home to entry page. Webmap plans to support features such as multiple document views, automatic map creation using World-Wide Web search engines, operations to store and load several journeys (navigation sessions); it include time stamps in typology node statistics of the frequency of visits.



Current: Title: First WWW Conference
URL: http://www1.cern.ch/WWW94/Welcome.html

Touched: Title: Secretariat
URL: http://www1.cern.ch/WWW94/Secretariat.ht

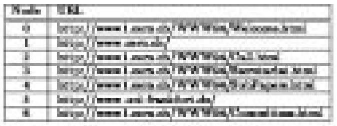
Figure 1: A sample session with WebMap.



Current: Title: Johann Wolfgang Goethe Universität Intro
URL: http://www.uni-frankfurt.de/

Touched: Title: Johann Wolfgang Goethe Universität Intro
URL: http://www.uni-frankfurt.de/

Figure 2: Sample session with WebMap: some clicks later.



Node	URL
0	http://www1.cern.ch/WWW94/Welcome.html
1	http://www.cern.ch/
2	http://www1.cern.ch/WWW94/Call.html
3	http://www1.cern.ch/WWW94/Secretariat.html
4	http://www1.cern.ch/WWW94/SubPages.html
5	http://www.uni-frankfurt.de/
6	http://www1.cern.ch/WWW94/Committees.html

Table 1: URLs of documents corresponding to the nodes shown in Figures 1 and 2.

Example screen images of WebMap

(Source: http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Searching/doemel/www-fall94.html; reproduced with permission.)

Gedye, D. and R. Katz. “Browsing the chip design database.” Technical Report, UCB//CSD-87-378, Computer Science Division, University of California at Berkeley, 1987. <ftp://tr-ftp.cs.berkeley.edu/pub/tech-report/csd/csd-87-38/>.

Multiple windows (in 2D) are used to display relationships between design objects, which may belong to one of three sets: composition, version, or equivalence. Since these relationships are orthogonal, they are displayed independently as a directed acyclic graph (DAG), a version tree, and an equivalence set, respectively. Users navigate by directly manipulating design objects (shown by their names) in each window.

Henry, T. R. and S. E. Hudson. “Interactive graph layout.” UIST '91, Hilton Head, South Carolina. ACM Press, 1991. 55-64.

The paper introduces three concepts to enable the user to dissect large graphs iteratively into manageable pieces and to lay out the portions in a manner that clearly reflects the user's current focus: (1) an architecture for building new simple graph layout algorithms out of existing algorithms, (2) parameterized graph layout algorithms to give the user control over the layout process, and (3) an interactive mechanism for selecting portions of the graph that match the user's current focus. The paper also introduces metagraphs to (1) convey the structure of the hierarchy of layout algorithms to the user, (2) allow the user to manipulate the hierarchy, and (3) provide an interface for the parameters of individual layout algorithms.

Hovestadt, L. "A4 digital building: extensive computer support for building design, construction, and management." CAAD Futures '93. Eds. Flemming, U. and S. Van Wyk. Elsevier Science Publishers, 1993. 405-421.

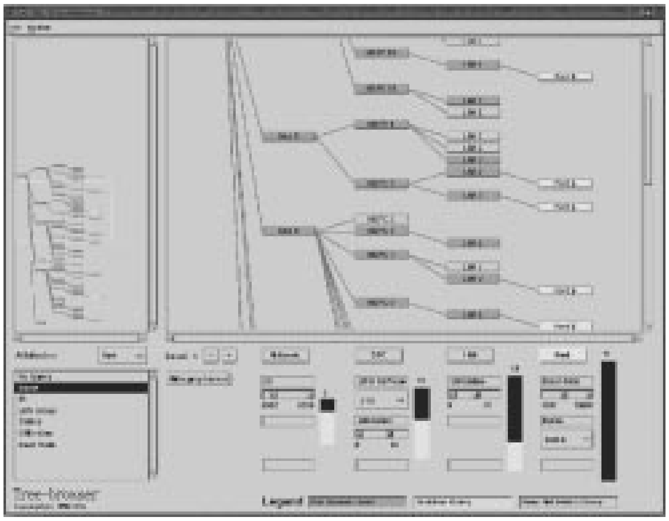
The paper proposes a method to structure the data needed in building design, construction and management. The dataspace is multidimensional, where each dimension represents a specific design feature. A4 includes as dimensions the x-, y-, z-coordinates, time, precision, morphology, size, aspect, alternatives, timetag, and user IDs. Each dimension is considered an axis in a multi-dimensional space modeled like a Cartesian coordinate space. Any design object is viewed as a point in that space whose coordinates are given by the location of the object's features on their respective axes. A strong underlying assumption is that the feature values are independent and can be linearized so that each feature value marks a unique point on the respective feature axis. Two commands, moveTo and Forget, are provided for navigation in the dataspace.

Compelling as A4 is because of its direct application to building design, its significance for SEED appears limited because features like the hierarchical relations ubiquitous in SEED cannot be modeled in a similar way.

Query, Search and Filter

Query, search and filter are navigation aids not available in the physical environment. Kumar, Plaisant and Shneiderman (1995) and Williamson and Shneiderman (1992) demonstrate these aids through dynamic queries for database applications. Stone, Fishkin and Bier (1994) present a user interface tool providing a filter utility. In general, design space navigation tools should provide search/filter facilities for users so that they can quickly find the desired information (see also "Empirical Studies" on page 5).

Kumar, H. P., C. Plaisant and B. Shneiderman. "Browsing hierarchical data with multi-level dynamic queries and pruning." Technical Report, CAR-TR-772/CS-TR-3474/ISR-TR-95-53, Human-Computer Interaction Laboratory, Department of Computer Science & Institute for Systems Research, University of Maryland, College Park, MD 20742, 1995.



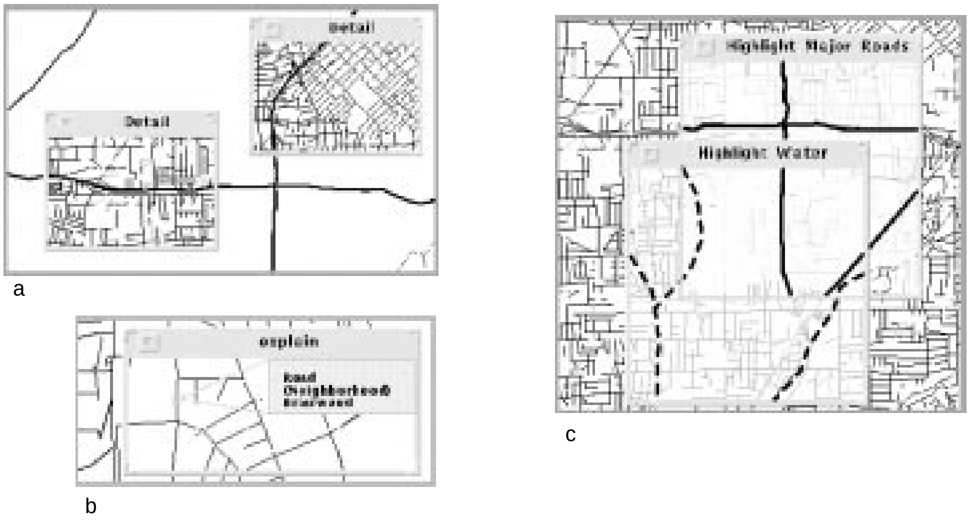
The PDQ Tree-browser interface

(Source: http://www.cs.umd.edu:80/TR/UMCP-CSD:CS-TR-3474; reproduced with permission.)

A good review section describes previous work related to visualizing large information spaces. A 'PDQ Tree-browser' is introduced with the following features: (1) two tightly-coupled node-link views of the tree (overview and detailed view), (2) a 'dynamic query environment' for users to customize their dynamic query panels, (3) dynamic queries at different levels of the tree, and (4) pruning of subtrees of uninteresting nodes to get more compact views. The underlying data structure has no limitation, but the UI design / representation is fine-tuned for a tree of depth 5.

Stone, M. C., K. Fishkin and E. A. Bier. "The movable filters as a user interface tool." CHI '94 Human Factors in Computing Systems: Celebrating Interdependence, Boston, Massachusetts. Eds. Adelson, B., S. Dumais and J. Olson. Addison Wesley, 1994. 306-312.

'Magic lens' filters employ an analogy with physical lenses; they show a modified view in the context of the original view. The modified view can be a detail view, a coordinated alternate view (view of other types of information), or a visual macro (filter).



Examples of movable filters

(Source: http://www.xerox.com/PARC/istl/gir/94CHIFilters.html; reproduced with permission.)

a. Lenses showing detail in two different locations.
b. The lens shows the text tags for the indicated segment of road.
c. Composing lenses to show waterways (dashed lines) and major roads (bold lines).

Williamson, C. and B. Shneiderman. "The dynamic HomeFinder: evaluating dynamic queries in a real-estate information exploration system." Technical Report, CAR-TR-602/CS-TR-2819, Human-Computer Interaction Laboratory & Department of Computer Sciences, University of Maryland, College Park, MD 20742, 1992.

The authors observe that users, in general, prefer the Dynamic Query (DQ) interface over traditional database query interfaces because it is faster, fun to work with (the animated graphical nature of the DQ interfaces may have motivational power), easy to use, and makes it easy to see the result. They summarize the benefits and drawbacks of dynamic queries over traditional database queries.

Information Visualization

This section cites again some sources introduced in preceding sections; but this time, the visualization method itself is the object of interest.

Visualization Frameworks and Techniques

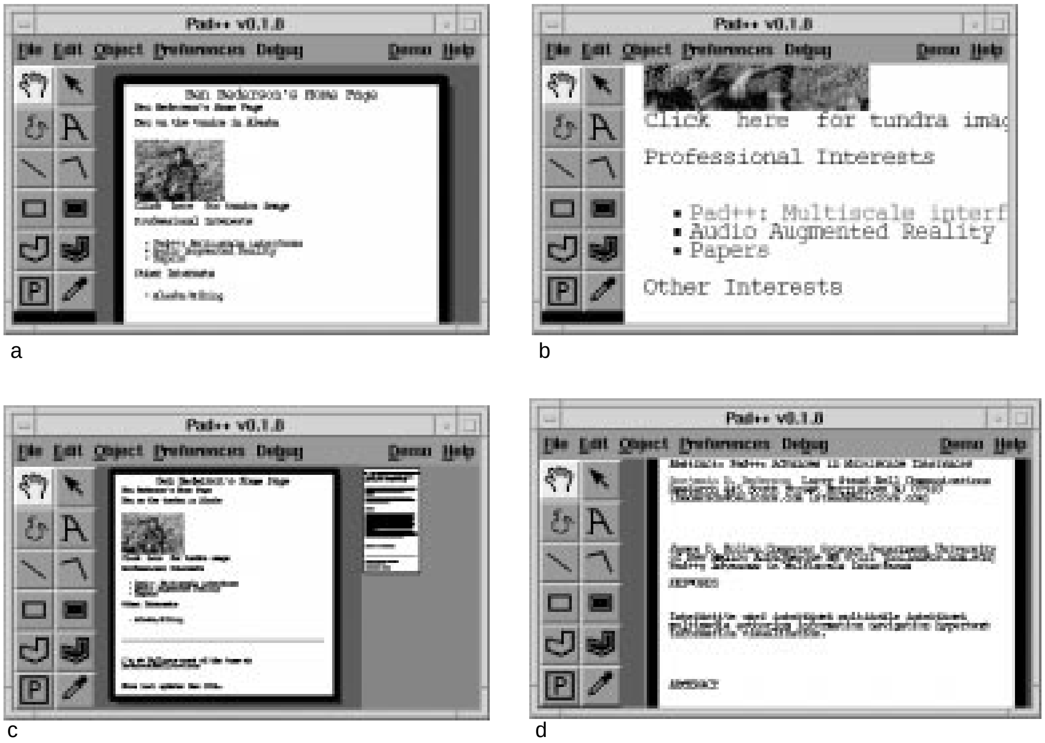
The visualization frameworks or techniques proposed in the reviewed literature can be categorized into four types:

- Traditional methods: used in visualizing statistical information, such as tables, panels, graphs, scatter plots, and maps (Chimera 1991; Plaisant, Carr and Shneiderman 1994; Zarmer and Chew 1992);
- Nodes and links: used for visualizing information with hierarchical or network structures, where nodes represent data sets and links between nodes depict relationships between data sets; the visualization usually takes the form of 2D or 3D trees or networks, 3D cone trees, or structured hyperlinks (Mukherjea and Foley 1995);
- Multiscale views: the information is presented at different scales to show the data of focus (in the largest scale) as well as contextual information; for example, the Spiral Calendar (Mackinlay, Robertson and DeLine 1994) and Pad++ (Bederson and Hollan 1994) show the history of object interactions at successive scales;
- Perspective views: used to present multi-attribute information 3-dimensionally, for example, the Perspective Wall in Mukherjea and Foley (1995), or the Time Lattice in Mackinlay, Robertson and DeLine (1994).

Since the information in SEED design spaces is highly structured, the nodes and links framework is used predominantly in our current implementation. We plan to investigate this framework further, especially the use of 3D displays such as 3D cone trees. In addition, multiscale views may help to reduce clutter in design space displays by arranging objects at different scales and resolutions.

Bederson, B. B. and J. D. Hollan. “Pad++: a zooming graphical interface for exploring alternate interface physics.” <u>ACM Symposium on User Interface Software and Technology</u>, Marina del Rey, California. ACM Press, 1994. 17-26.

The Pad++ system uses a multiscale view to present the user with an overview of a selected set of objects. The objects are history-enriched and stored along with the interaction events that comprise their uses. The display of a 'history-enriched object' shows a graphical abstraction of the accrued histories as part of the object itself (see the figure below).



Ben Bederson's Home Page

Click here for tundra imag

Professional Interests

- Pad++: Multiscale interf
- Audio Augmented Reality
- Papers

Other Interests

Example screen images of Pad++

(Source: http://www.cs.unm.edu/pad++; reproduced with permission.)

a. A screen snapshot showing Pad++ displaying an HTML document.
b. A zoomed-in view of the document. Hotwords are shown in blue; positioning the pointer over a hot-word changes its color to red.
c. When you follow a link, the relevant document is loaded into Pad++ and placed on the surface to the right of the original document, at a smaller scale.
d. Simply zoom-in on the sub-document to read it.

Chimera, R. “Value Bars: an information visualization and navigation tool for multi-attribute listings and tables.” Technical Report, CAR-TR-589/CS-TR-2773, Human-Computer Interaction Laboratory, University of Maryland, College Park, MD 20742-3255, 1991.

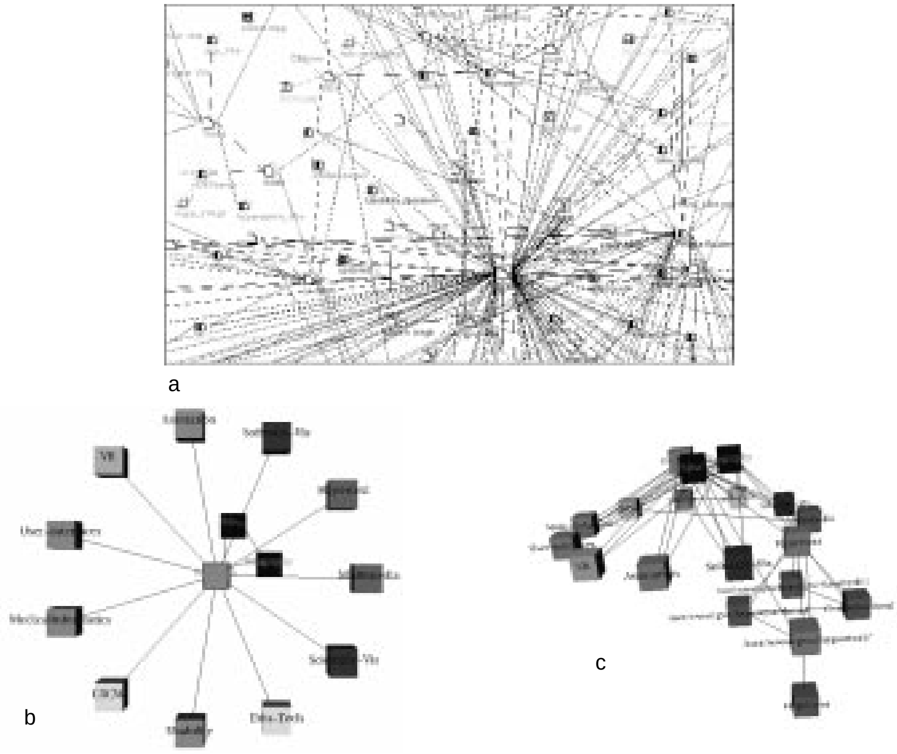
The Value Bars display attributes representing size or quantity measures as linear scales.

Mackinlay, J. D., G. G. Robertson and R. DeLine. “Developing calendar visualizers for the Information Visualizer.” ACM Symposium on User Interface Software and Technology, Marina del Rey, California. ACM Press, 1994. 109-118.

This paper explores two methods of using 3D graphics and interactive animation to design and implement visualizers that improve access to large masses of time-based information: the Spiral Calendar for rapid access to an individual's daily schedule; and the Time Lattice for analyzing the time relationships among the schedules of groups of people.

Mukherjea, S. and J. D. Foley. "Visualizing the World-Wide Web with the Navigational View Builder." Computer Networks and ISDN System, Special Issue on the Third International Conference on the World-Wide Web '95, Darmstadt, Germany. April 10-14, 1995. <http://www.igd.fhg.de/www/www95/proceedings/papers/mukh/mukh.html>.

This paper describes the Navigational View Builder, a tool to assist users visualizing information. Several strategies are used to structure the information: binding (data<->visual properties, e.g. color), clustering (find files that are closely related), filtering (allow user control), and hierarchization. The Navigational View Builder provides several visualization methods: network overview (with different bindings), table-of-contents, 3D trees, 2D trees, cone trees, treemaps, and perspective walls.



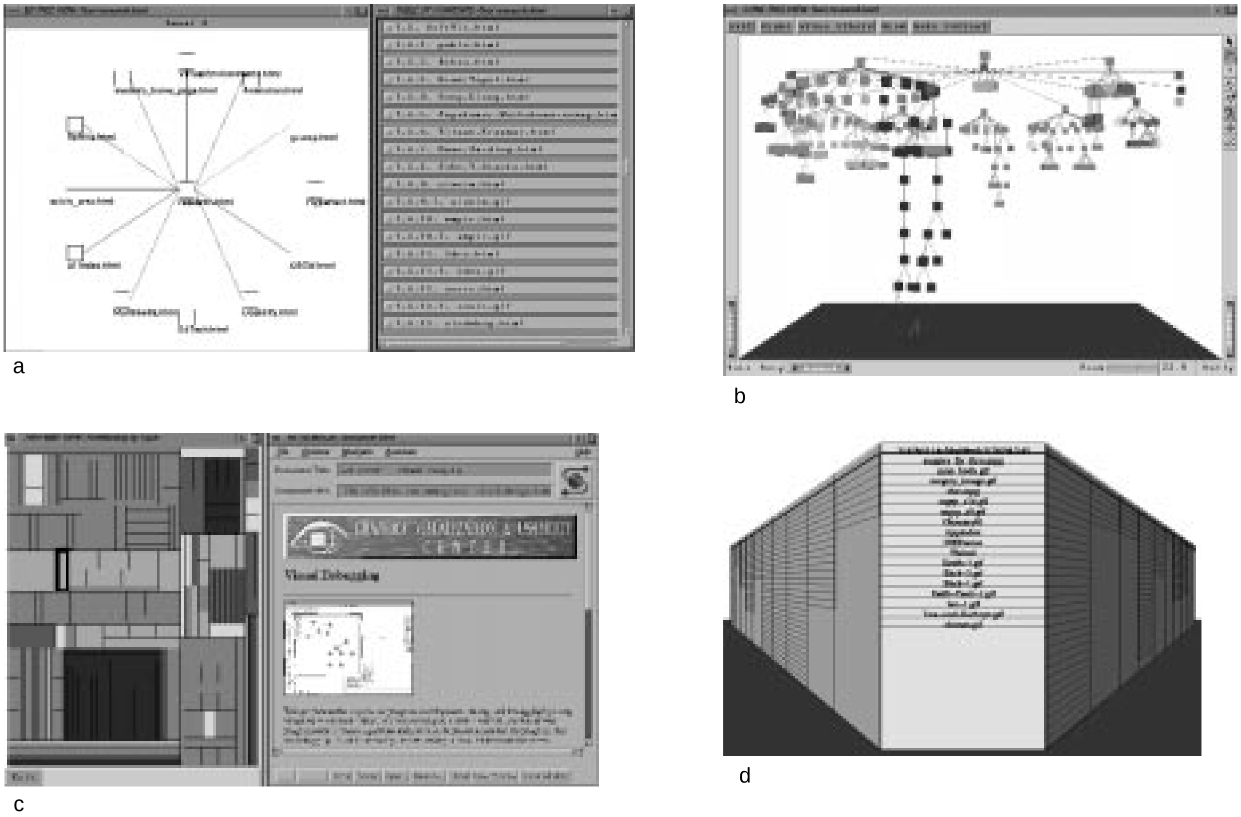
Example views of the Navigational View Builder (1)

(Source: http://www.cc.gatech.edu/gvu/people/Phd/sougata/Nvb.html; reproduced with permission.)

a. An example of a view of the GVU WWW pages with various visual properties bound to information attributes;

b. A view where the user wanted to see the details of the research pages. The *x-y* plane shows the details of each particular layer; the layers are arranged in the *z* dimension with the most detailed view (of the Research pages) in front;

c. A top view of the information space after the user wanted to see details of the hypertext research area.



Example views of the Navigational View Builder (2)

(Source: http://www.cc.gatech.edu/gvu/people/Phd/sougata/Nvb.html; reproduced with permission.)

a. The left-hand screen shows the top level of the default hierarchy created for the data by the Navigational View Builder algorithm. The right-hand side shows a view of a section of this hierarchy, where the nodes are listed like a table of contents of a book.
b. A 3d tree view of this hierarchy.
c. The left-hand screen shows a Treemap view of a hierarchy; the corresponding WWW page is shown in the right-hand screen.
d. A perspective wall view of a linear structure sorted by the last-modified-time.

Plaisant, C., D. Carr and B. Shneiderman. "Image browsers: taxonomy, guidelines, and informal specifications." Technical Report, CAR-TR-712/CS-TR-3282/ISR-TR-94-39, Human Computer Interaction Laboratory, University of Maryland, College Park, MD 20742, 1994.

This is a very comprehensive report on image browsing of objects in 2D space.

Zarmer, C. L. and C. Chew. "Frameworks for interactive, extensible, information-intensive applications." UIST '92, Monterey California. ACM Press, 1992. 33-41.

This paper describes a set of application frameworks called 'visual formalisms' that provide widgets, data structures, editing operations, and layout algorithms for visual presentations such as tables, panels, graphs, outlines, plots and maps. The architecture of the formalisms contain (1) a C++ run-time type facility, (2) a protocol for asking objects about their operations (e.g. name/class query), and (3) UI classes that support event notifications.

Visualizing Hierarchical Structures

Hierarchical Structures are often visualized using 2D tree or network diagrams, (see, e.g., Pitkow and Bharat 1994). However, traditional tree or network views become unmanageable when the number of nodes increases. The 'focus+context' scheme (Brown, Meehan and Sarkar 1993; Lamping and Rao 1994) addresses this problem by enabling users to focus on the information of interests while maintaining the relationships between these focused data and the overall context.

Complex information, such as high-dimensional hierarchical structures, is difficult to visualize in a 2D tree or network diagram. Koike (1993) adds the third dimension to 2D trees and networks to provide users with additional information about the hierarchical structure; Pitkow and Bharat (1994) address this issue by using visual encodings through shapes, colors, and labels.

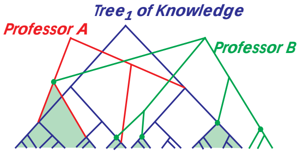
The design spaces in SEED are growing continuously at run-time, and it may be beneficial to employ the focus+context scheme. In addition, the overall organization of design spaces is potentially multidimensional. A 3D tree or network visualization of design spaces conveys more information to users than a traditional 2D view. But these two methods require high computational power to support interactive manipulations, and design space navigation should remain a supporting tool in SEED so that users or the system are not be bogged down by it.

Brown, M. H., J. R. Meehan and M. Sarkar. “Browsing graphs using a fisheye view.” INTERCHI '93 Conference on Human Factors in Computing Systems: Bridges Between Worlds, Amsterdam, The Netherlands. Eds. Ashlund, S., K. Mullet, A. Henderson, E. Hollnagel and T. White. Addison Wesley, 1993. 516.

The authors present 'fisheye views' (for example, see figure “An example view of a hyperbolic space” on page 19), a scheme to visualize large hierarchies by showing the area of interest with greater detail and the rest of the structure with successively less detail.

Furnas, G. W. and J. Zacks. “Multitrees: enriching and reusing hierarchical structure.” CHI '94 Human Factors in Computing Systems: Celebrating Interdependence, Boston, Massachusetts. Eds. Adelson, B., S. Dumais and J. Olson. Addison Wesley, 1994. 330-336.

The paper introduces a new type of structure for representing information, 'multitrees', a class of directed acyclic graphs with the unusual property that they have large, easily identifiable substructures that are trees. Multitrees are DAGs; therefore, a node in the structure can have multiple parents. Several ways of visualizing multitrees are presented.



An example multitrees structure

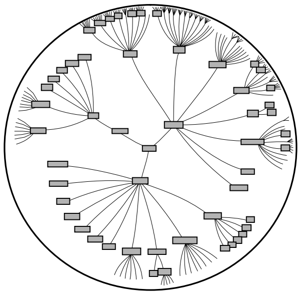
(Redrawn from original.)

Koike, H. “The role of another spatial dimension in software visualization.” ACM Transactions on Information Systems, 11.3 (1993): 266-286.

The paper demonstrates a 3D visualization framework, where objects (nodes) are 3D objects implemented using a 3D graphics library, and results are displayed in perspective views. Since the displayed objects are actually modeled, the viewing angle can easily be changed.

Lamping, J. and R. Rao. “Laying out and visualizing large trees using a hyperbolic space.” <u>ACM Symposium on User Interface Software and Technology</u>, Marina del Rey, California. ACM Press, 1994. 13-14.

The authors present a new 'focus+context' (fisheye) scheme for visualizing and manipulating large hierarchies.



An example view of a hyperbolic space

(Redrawn from original.)

Pitkow, J. E. and K. A. Bharat. “WebViz: a tools for World-Wide Web access log analysis.” <u>The First International World-Wide Web Conference</u>, Geneva, Switzerland, May 25-27, 1994. <http://www1.cern.ch/PapersWWW94/pitkow-webvis.ps>.

The paper presents a layout algorithm for overviewing WWW networks (see figure (a) of “Example views of the Navigational View Builder (1)” on page 16).

Visualizing High-Dimensional Objects

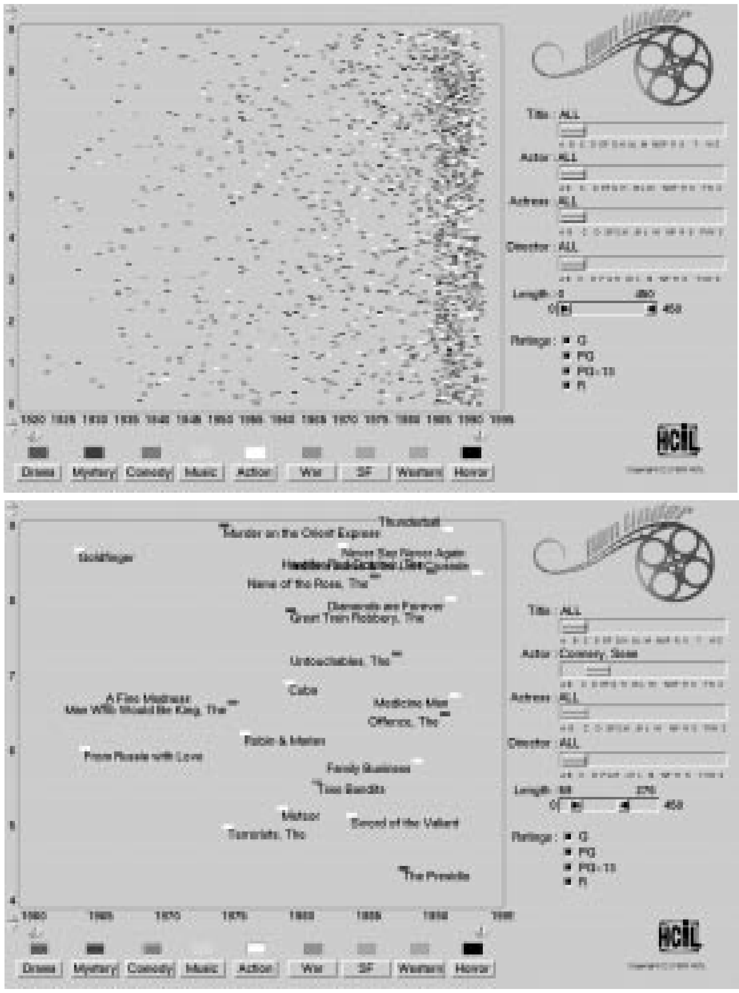
The common approach to visualize high-dimensional objects (that is, objects with more than 3 dimensions of interest) is to present the two primary dimensions in 2D, and then to supply visual cues (such as colors, patterns, and texts) to distinguish additional dimensions. For example, Ahlberg and Shneiderman (1994) use 2D scatter plots along with colors to present filtered information from a database. Feiner and Beshers (1990) introduce a 'worlds within worlds' metaphor which presents higher dimensions embedded in 3D space (see figure “An example worlds within worlds display” on page 21). Unlike the previous approaches, which compress/filter high-dimensional objects and display them in 2D or 3D space, Furnas and Buja (1994) propose a new technique, prosection, to visualize high-dimensional spaces, for instance, a 3D object in a 6D space.

The organization of design spaces in SEED is high-dimensional (see “Introduction” on page 1). However, the notion of *dimensions* in the SEED design space differs from those mentioned in the literature, which emphasizes individual attributes that can often be mapped on linear axes or 'dimensions'; such attributes may exist in SEED-generated objects, but the basic dimensions in the SEED design space are the different hierarchical structures indicating abstraction levels, constituent or derivation relations, where especially the latter are non-linear by nature. Furthermore, we expect users to interact directly with the objects in a design space. Nevertheless, some of the approaches relying on 2D or 3D displays seem to have promise for our purpose. Techniques like the 'worlds within worlds' or prosection, which require high computing power, may not be feasible because of the highly interactive nature of SEED design spaces.

Ahlberg, C. and B. Shneiderman. “Visual information seeking: tight coupling of dynamic query filters with starfield displays.” <u>CHI'94 Human Factors in Computing Systems: Celebrating</u>

<u>Interdependence</u>, Boston, Massachusetts. Eds. Adelson, B., S. Dumais and J. Olson. Addison Wesley, 1994. 313-317.

This paper introduces the 'starfield display', a 2D scatterplot of a multidimensional database, where every item from the database is represented as a small colored glyph whose position is determined by its ranking along ordinal attributes laid out on multiple axes.

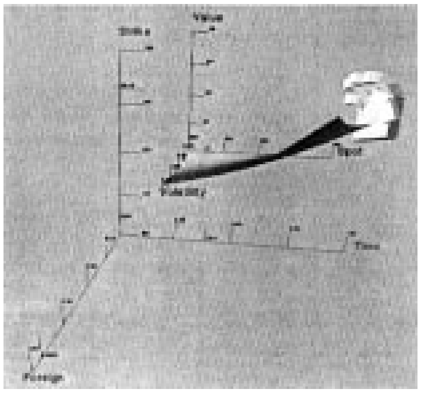


Example starfield displays

(Source: http://www.cs.chalmers.se/SSKKII/ivee-dumps/filmfinder.html; reproduced with permission.)

Feiner, S. and C. Beshers. "Worlds within worlds: metaphors for exploring n-dimensional virtual worlds." <u>Third Annual Symposium on User Interface Software and Technology</u>, Snowbird, Utah, USA. ACM Press, 1990. 76-83.

This paper describes the 'worlds within worlds' metaphor, which reduces the complexity of a multidimensional space by holding one or more of its independent dimensions constant to decrease the number of dimension to three and then embeds in this 3D world another 3D world that represents three additional dimensions. The position of the embedded world's origin relative to the containing world's coordinate system specifies the value of three of the inner world's variables that are held constant. This process can then be repeated by further recursive nesting of heterogeneous worlds to represent the remaining dimensions. The example application shown below is in the financial visualization domain.



An example worlds within worlds display

(Source: the original paper; reproduced with permission.)

The outer world has axes of time to maturity, strike price, and foreign interest rate. The domestic interest rate had been held constant and is not assigned to an axis. The position of the inner world determines the time to maturity, strike price, and foreign interest rate used in evaluating the function.)

Furnas, G. W. and A. Buja. "Prosection views: dimensional inference through sections and projections." Journal of Computational and Graphical Statistics, 3(4). 1994: 377+. <http://community.bellcore.com/gwf/home-page.html>.

This paper presents some basic properties of projection and section (which are general techniques for constructing views of high-dimensional objects) and introduces a new technique called 'prosection', a composition of section and projection. Algorithms of projection, section, and prosection are provided in detail. Applications of prosections are demonstrated in statistical graphics systems such as displaying a 3-dimensional torus in 6-dimensional space and plotting 7-dimensional physics data.

Multiscale Displays

There are 2 types of multiscale displays: focus+context (Brown, Meehan and Sarkar 1993; Lamping and Rao 1994; Sarkar et al. 1993) and pan+zoom (Bederson and Hollan 1994; Bier et al. 1993; Lieberman 1994; Perlin and Fox 1993; Rennison 1994). A focus+context scheme displays information at continuous scales, while a pan+zoom interface uses discrete scales. However, the pan+zoom interface provides possibilities for many scales overlaying at one time, while the focus+context scheme provides a single view of the information space.

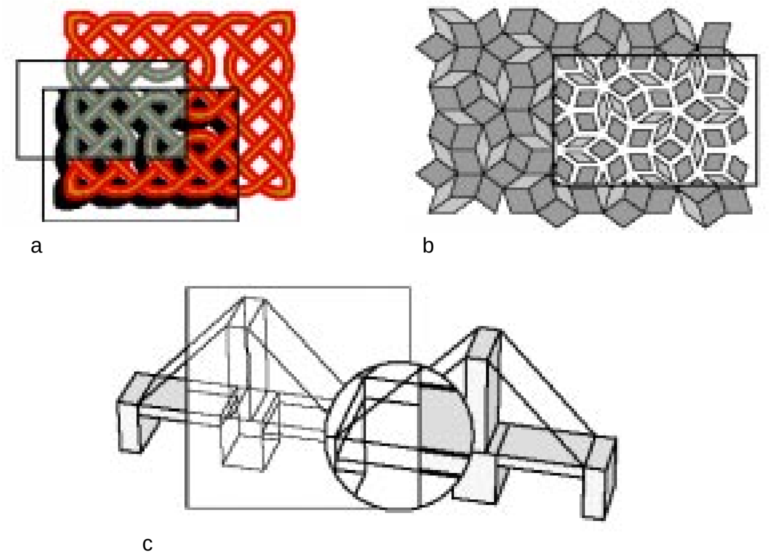
Since design spaces can contain large amounts of information, multiscale displays may be desirable. However, the focus+context scheme may not suffice for our needs because it supports the display of 2D structures only, while the 'zoom' operation in the pan+zoom scheme allows information filtering and thus can potentially support views of other dimensions.

Bederson, B. B. and J. D. Hollan. “Pad++: a zooming graphical interface for exploring alternate interface physics.” ACM Symposium on User Interface Software and Technology, Marina del Rey, California. ACM Press, 1994. 17-26.

The Pad++ system uses a multiscale view to present the user with an overview of a selected set of objects. The objects are history-enriched and stored along with the interaction events that comprise their uses. The display of a 'history-enriched object' shows a graphical abstraction of the accrued histories as part of the object itself.

Bier, E. A., N. C. Stone, K. Pier, W. Buxton and T. D. DeRose. “Toolglass and magic lenses: the see-through interface.” SIGGRAPH '93, Anaheim, CA. ACM Press, 1993. 73-80.

This paper introduces a new style of graphical user interface called the 'see-through interface'. It includes semi-transparent interactive tools, called 'toolglass widgets' that appear on a virtual sheet of transparent glass, the 'toolglass sheet', placed between the application and a traditional cursor. These widgets provide customized views of the application underneath them using 'magic lens' filters. Each lens is a screen region coupled with an operator such as “magnification” or “render in wireframe,” performed on the objects viewed in the region. Examples are demonstrated below for clipboards, previewing lenses, selection tools, grids, and visualization.



Examples of magic lenses

(Source: the original paper; reproduced with permission.)

a. An achromatic lens over a drop shadow lens over a knotwork (knotwork by Andrew Glassner);
b. The local scaling lens (tiling by Doug Wyatt);
c. A bridge made of shaded, 3D blocks showing a 3D wireframe lens and a 2D magnifier.

Brown, M. H., J. R. Meehan and M. Sarkar. “Browsing graphs using a fisheye view.” INTERCHI '93 Conference on Human Factors in Computing Systems: Bridges Between Worlds, Amsterdam, The Netherlands. Eds. Ashlund, S., K. Mullet, A. Henderson, E. Hollnagel and T. White. Addison Wesley, 1993. 516.

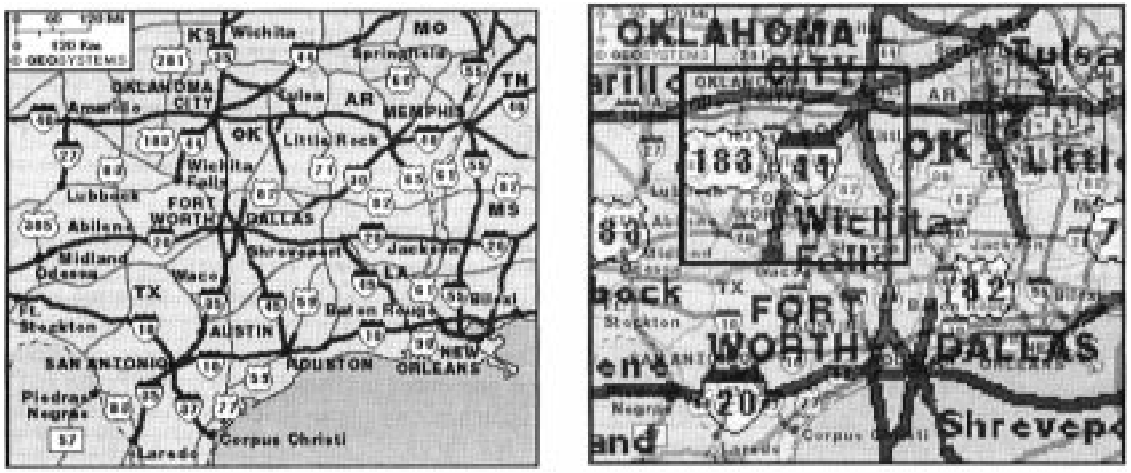
The authors present 'fisheye views', a focus+context scheme to visualize large hierarchies by showing the area of interest with greater detail and the rest of the structure with successively less detail.

Lamping, J. and R. Rao. "Laying out and visualizing large trees using a hyperbolic space." ACM Symposium on User Interface Software and Technology, Marina del Rey, California. ACM Press, 1994. 13-14.

The authors present a new 'focus+context' (fisheye) scheme for visualizing and manipulating large hierarchies.

Lieberman, H. "Powers of ten thousand: navigating in large information spaces." ACM Symposium on User Interface Software and Technology, Marina del Rey, California. ACM Press, 1994. 15-16.

The paper proposes 'macroscope', a technique based on zooming and panning in multiple translucent layers.



Example displays of translucent layers with the macroscope

(Source: the original paper; reproduced with permission.)

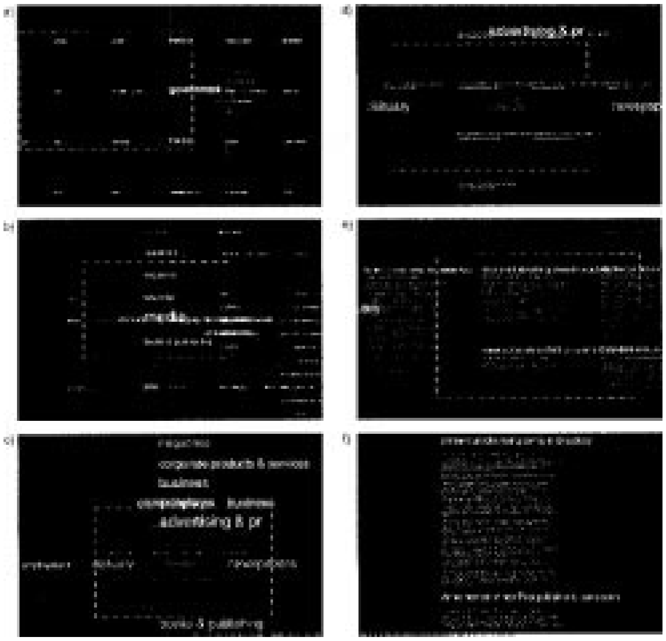
Perlin, K. and D. Fox. "Pad: an alternative approach to the computer interface." SIGGRAPH '93, Anaheim, CA. ACM Press, 1993. 57-64.

Pad is an infinite two-dimensional information plane. Objects are organized on that plane geographically: every object occupies a well-defined region on the plane's surface. 'Portals' act like magnifying glasses that can peer into and roam over different parts of the Pad surface. The screen itself is just a special 'root' portal.

To facilitate the display, two techniques are used: 'semantic zooming' and 'portal filters'. Semantic zooming can be used to control the amounts or types of information to be displayed at certain scales. Portal filters control how information is displayed (e.g., in textual, tabular, chart or graphical format).

Rennison, E. "Galaxy of News: an approach to visualizing and understanding expansive news landscape." ACM Symposium on User Interface Software and Technology, Marina del Rey, California. ACM Press, 1994. 3-12.

At the heart of the Galaxy of News is an engine that constructs an associative relation network (potentially multidimensional) that automatically builds implicit links between related articles. Although the information space contains relationships that are multidimensional, not all elements of the relational hierarchy are visible at a single glance; rather, only elements that are relevant to the user's present view are shown in the pan+zoom visualization scheme.

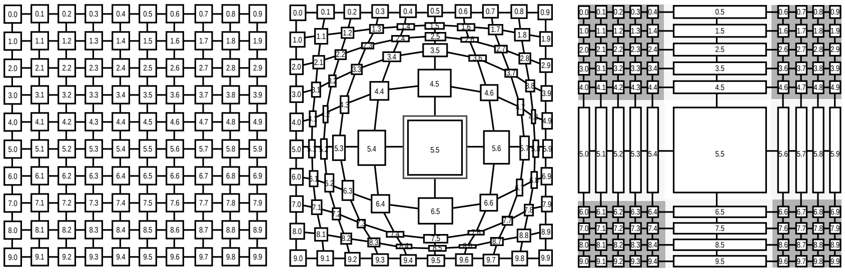


Example displays of the Galaxy of News system

(Source: the original paper; reproduced with permission.)

Sarkar, M., S. S. Snibbe, O. J. Tversky and S. P. Reiss. "Stretching the rubber sheet: a metaphor for viewing large layouts on small screens." UIST '93, Atlanta, Georgia. ACM Press, 1993. 81-91.

The authors propose the metaphor of rubber sheet stretching for viewing large and complex layouts within small display areas. As the user stretches an area, a greater level of detail is displayed there. This technique is similar to the fisheye view. However, the method contains mechanisms to stretch arbitrary (orthogonal or polygonal) regions and multiple foci, which fisheye views do not support.



Examples of the rubber sheet metaphor

(Redrawn from originals.)

Translucent/Transparent Displays

This technique is used mostly in conjunction with the pan+zoom scheme (See “Multiscale Displays” on page 21) to allow users to receive as much information as possible while keeping track of the surrounding context. It is analogous to an architects’ tracing paper and can be used, for example, in the SEED-Layout Design Window, where each layer of tracing paper can display a problem decomposition level in the problem specification.

Bier, E. A., N. C. Stone, K. Pier, W. Buxton and T. D. DeRose. “Toolglass and magic lenses: the see-through interface.” SIGGRAPH '93, Anaheim, CA. ACM Press, 1993. 73-80.

This paper introduces the 'see-through interface', which includes semi-transparent interactive tools, called 'toolglass widgets' that appear on a virtual sheet of transparent glass, the 'toolglass sheet', placed between the application and a traditional cursor. These widgets provide customized views of the application underneath them using 'magic lens' filters. Each lens is a screen region together with an operator, such as “magnification” or “render in wireframe,” performed on objects viewed in the region (see figure “Examples of magic lenses” on page 22). Examples are demonstrated for clipboards, previewing lenses, selection tools, grids, and visualization. Management of sheets is discussed where three approaches are presented.

Kramer, A. “Translucent patches–dissolving windows.” ACM Symposium on User Interface Software and Technology, Marina del Rey, California. ACM Press, 1994. 121-130.

This paper presents motivation, design and algorithms for using and implementing translucent, non-rectangular patches as a substitute for rectangular opaque windows. The underlying metaphor is closer to a mix between the architect's yellow or trace paper and white boards than to rectangular opaque paper in piles and folders on a desktop. The paper contains a literature survey regarding translucent and transparent displays.

Lieberman, H. “Powers of ten thousand: navigating in large information spaces.” ACM Symposium on User Interface Software and Technology, Marina del Rey, California. ACM Press, 1994. 15-16.

The paper proposes 'macroscope', a technique based on zooming and panning in multiple translucent layers.

Rennison, E. “Galaxy of News: an approach to visualizing and understanding expansive news landscape.” ACM Symposium on User Interface Software and Technology, Marina del Rey, California. ACM Press, 1994. 3-12.

At the heart of the Galaxy of News is an engine that constructs an associative relation network (potentially multidimensional) that automatically build implicit links between related articles. Although the information space (relationships) is multidimensional, not all elements of the relational hierarchy are visible at a single glance; rather, only elements that are relevant to the user's present view are shown. The system employs a transparent display technique to show elements in different scales (see figure “Example displays of the Galaxy of News system” on page 24).

3D Displays/Effects

Many information visualization environments add a third dimension in the display to convey more information. Although 3D displays usually require specialized computer hardware and software (Koike 1993), simple 3D effects may achieve the same result (Staples 1993).

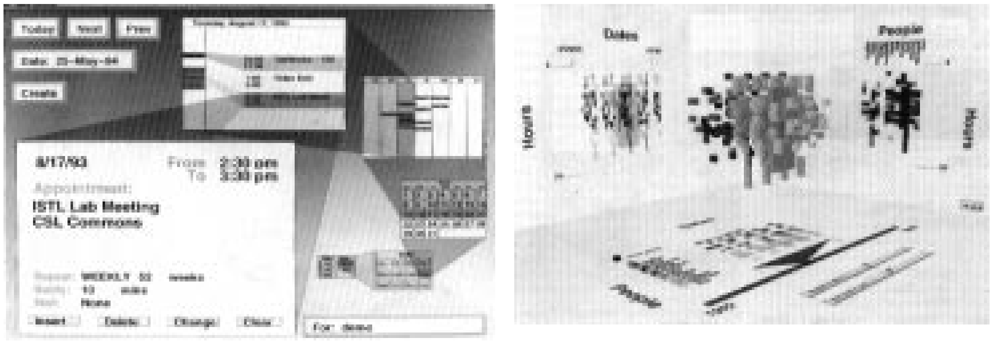
In SEED, 3D visualization is desirable, but the system response time should remain reasonable to support interactive object manipulations.

Koike, H. "The role of another spatial dimension in software visualization." ACM Transactions on Information Systems, 11.3 (1993): 266-286.

The paper demonstrates a 3D visualization framework. Objects (nodes) are 3D objects implemented using a 3D graphics library, and results are displayed in perspective views. Since the displayed objects are actually modeled, the viewing angle can easily be changed.

Mackinlay, J. D., G. G. Robertson and R. DeLine. "Developing calendar visualizers for the Information Visualizer." ACM Symposium on User Interface Software and Technology, Marina del Rey, California. ACM Press, 1994. 109-118.

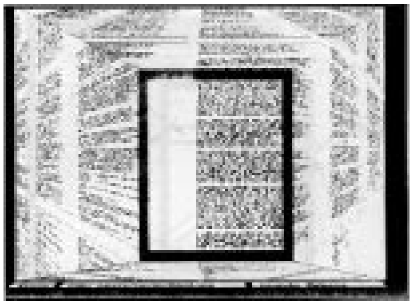
This paper explores two methods of using 3D graphics and interactive animation to design and implement visualizers that improve access to large masses of time-based information: the Spiral Calendar for rapid access to an individual's daily schedule; and the Time Lattice for analyzing the time relationships among the schedules of groups of people.



8/17/93 From 2:30 pm
To 3:30 pm
Appointment:
ISTL Lab Meeting
CSL Commons

Examples of the Spiral Calendar and Time Lattice
(Source: the original paper; reproduced with permission.)

Robertson, G. G. and J. D. Mackinlay. "The Document Lens." UIST '93, Atlanta, Georgia. ACM Press, 1993. 101-108.

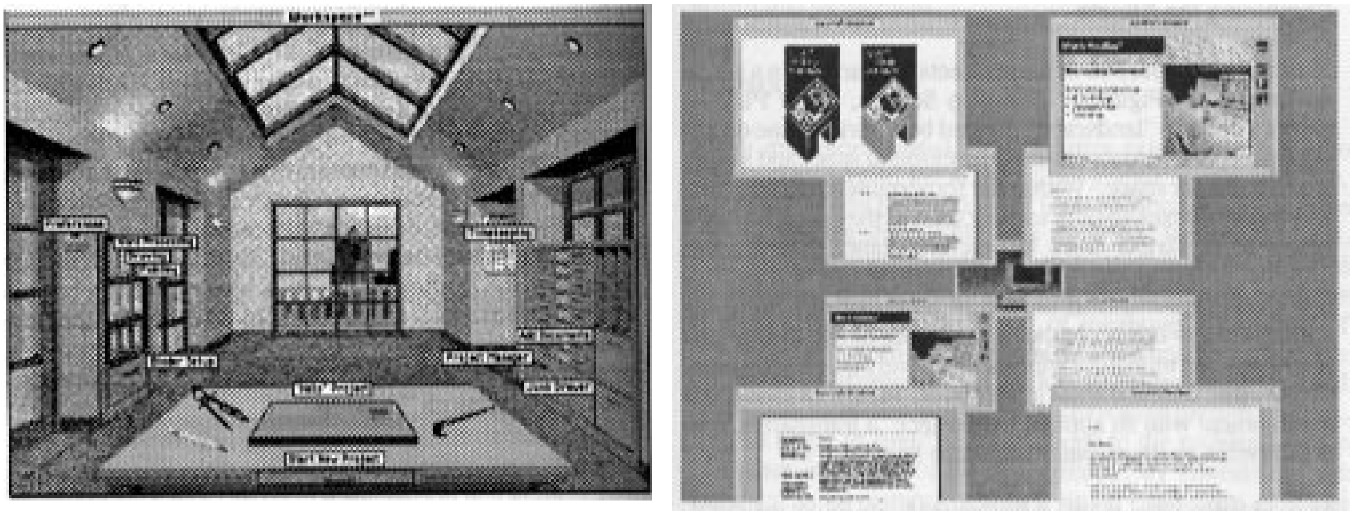


A Document Lens example
(Source: the original paper; reproduced with permission.)

The Document Lens is a 3D visualization of large rectangular presentations that allows the user to quickly focus on a part of a presentation while continuously remaining in context. This is a focus+context display and is similar to the fisheye view technique. However, the Document Lens displays information in a truncated pyramid fashion.

Staples, L. "Representation in virtual spaces: visual convention in the graphical user interface." INTERCHI '93 Conference on Human Factors in Computing Systems: Bridges Between Worlds, Amsterdam, The Netherlands. Eds. Ashlund, S., K. Mullet, A. Henderson, E. Hollnagel and T. White. Addison Wesley, 1993. 348-354.

This paper explores two ways to use 3D rendering effects in a GUI desktop–one point perspective and light effects from the history of art.



Two example displays using 3D rendering effects

(Source: the original paper; reproduced with permission.)

Visual Metaphors

A prominent visual metaphor is the physical object-in-space metaphor. For example, Clarkson (1991) uses the buildings/rooms metaphor, and Nabkel and Shafrir (1995) use cities/streets/landmarks. Feiner and Beshers' (1990) worlds-within-worlds metaphor introduces infinite entries into different dimensions and may combine with the pan+zoom visualization scheme (See "Multiscale Displays" on page 21) to show multiple worlds in one view. In addition, the rubber sheet metaphor (Sarkar et al. 1993) provides the opportunity to also display multiple scales in one view within a focus+context visualization scheme.

Any visual metaphor, if used in SEED, should be coordinated with the navigation scheme. From our initial study, the buildings/rooms and cities/streets/landmarks metaphors are worth investigating since these physical objects constitute some basic roles in human cognitive mappings (See "Cognitive Mapping" on page 3).

Clarkson, M. A. "An easier interface." Byte. 2, 1991. 277-282.

The Information Visualizer provides 3D 'rooms' as multiple virtual workspaces, interactive objects for different visualization methods (such as cone trees or perspective walls). A room is defined by a task-oriented clustering of information. Rooms are connected by 'doors'. "… Work is distributed throughout a collection of 3D/2D rooms furnished with interactive objects such as walls and floating trees." These visualizations are "designed to shift work to your perceptual system, freeing the conscious mind to work on larger problems."

Nabkel, J. and E. Shafrir. "Blazing the trail: design considerations for interactive information pioneers." SIGCHI Bulletin, 27.1 (1995): 45-54.

This article describes two products: HP UVE Help (hypertext style) and Access HP (WWW information service, URL: http://www.hp.com). Recognizable metaphors (e.g., street signs indicate directions) are used to help users create cognitive spatial map of the information space. The paper illustrates the iterative process of the metaphor design.

Feiner, S. and C. Beshers. "Worlds within worlds: metaphors for exploring n-dimensional virtual worlds." Third Annual Symposium on User Interface Software and Technology, Snowbird, Utah, USA. ACM Press, 1990. 76-83.

This paper describes the 'worlds within worlds' metaphor, which uses a common approach to reduce the complexity of a multidimensional space by holding one or more of its independent dimensions constant to reduce a world's dimension to 3D, then embeds in the 3D world another 3D world that represents three additional higher dimensions. The position of the embedded world's origin relative to the containing world's coordinate system specifies the value of three of the inner world's variables that are held constant. This process can then be repeated by further recursive nesting of heterogeneous worlds to represent the remaining dimensions. The example application is in the financial visualization domain.

Sarkar, M., S. S. Snibbe, O. J. Tversky and S. P. Reiss. "Stretching the rubber sheet: a metaphor for viewing large layouts on small screens." UIST '93, Atlanta, Georgia. ACM Press, 1993. 81-91.

The authors propose the metaphor of rubber sheet stretching for viewing large and complex layouts within small display areas. As the user stretch the area, a greater level of detail is displayed there. This technique is similar to the fisheye view. However, this paper describes mechanisms to stretch arbitrary (orthogonal or polygonal) regions and multiple focuses, which fisheye view does not support.