InfoSpace™ - A Conceptual Model for Interacting with Information in a Three-Dimensional Virtual Environment

James Leftwich, IDSA – April 1993

Orbit Interaction
131 Hawthorne Avenue Suite F
Palo Alto, California 94301-1062

ABSTRACT

With the advent of workstations and visualization devices capable of supporting three-dimensional stereoscopic virtual environments, it is possible to develop a system of information interaction that exists in such a space. This paper will describe InfoSpace, one conceptual form of such a system, its various components, and its evolutionary relationship to existing information systems and graphical user interfaces. In addition, the physiological, psychological and cognitive aspects of such an informational interface will be addressed.

1. INFOSPACE OBJECTIVE

(Note - June 2020: Sections 1, 2, and 3 and associated figures 1-20 were written and developed between November 1988 and February 1991)

InfoSpace is a model that combines a three-dimensional visual interface with a three-dimensional virtual environment to create a new mode of data organization and human/information interaction.

InfoSpace is a multi-dimensional information and data format that can serve as a three-dimensional virtual environment for the creation, manipulation, and access of many forms of storable data. Among these forms are text, audio, two-dimensional graphics, video, multimedia, as well as the various kinds of three-dimensional applications. A few examples of three-dimensional applications might be, but are not limited to, architectural or industrial design plans, molecular/medical models, mathematical illustrations/animations, statistical or scientific visualization. These might be static, animated, or user-interactive, depending on the type and purpose of the information and application.

One of the advantages of such an interface is that it makes it possible for the user to develop a three-dimensional spatial relationship to the information with which he or she is interacting. Since we live in a three-dimensional world, it seems only natural that we should develop those tools that can help us to organize and perceive information and informational relationships in that way. Historically, the impact of the development of two-dimensional communication technology, i.e. a writing surface and the subsequent development of the various forms of writing, on the human ability to organize thoughts and information was extremely profound. Technological advances throughout the ages, such as the creation of paper, the advent of printing, the typewriter, and the modern computer have all contributed to a more efficient means of utilizing that two-dimensional form of communication, but have not broken from it in any fundamental way. Now, combining the rapidly increasing power of computation with the advances in the understanding of human/computer interaction, there exists the possibility of creating a fundamentally new level of information interaction.   It is accurate to term this a new level and not a completely new form because none of the previous forms of communication and information interaction need be abandoned, but rather can be incorporated within this new interlinked, three-dimensionally organizing superstructure.

Such a method of organizing and manipulating information may well have an equally important impact upon human thought, communication, learning and the interpretation of information, since it will allow the three-dimensional visualization of informational relationships in ways heretofore impossible or impractical. This gives rise to the possibility of utilizing spatial memory and other psycho-physiological means to absorb, process, and synthesize information. Just as an individual organizes, relates to, and interprets his or her physical environment in ways far more complex than can be represented on paper or similarly with the two-dimensional metaphors of the desktop utilized in many of today's visual interfaces, *InfoSpace* creates an informational environment in which these higher orders of interaction can take place. *InfoSpace* is an attempt to harness these higher orders of interaction and apply them in an informational universe that parallels our physical environment dimensionally, yet goes far beyond it in terms of its potential for simulation, manipulation and interaction by way of its virtual nature.

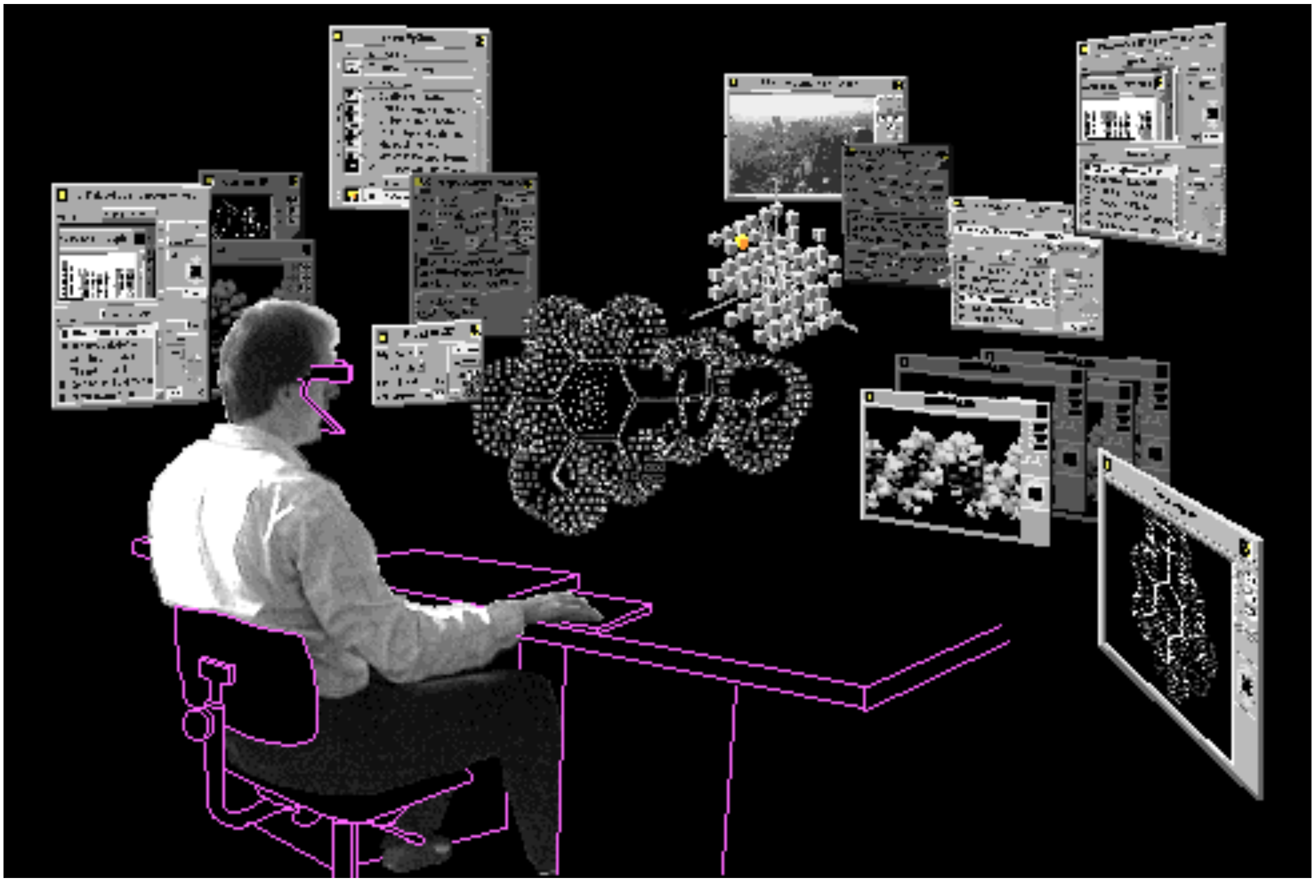


Figure 1: In this illustration we are “looking in” on the user as he perceives *InfoSpace*. The user is wearing *stereoscopic* eyephones that allow him to see the three-dimensionally projected array of *screens*, *control* and *status panels* and other elements. At this particular level, the user has *activated-to-3D* a three-dimensional virtual model of a molecule, which is now floating in front of him. He may wish to tumble or otherwise *manipulate* it, or scale it up and navigate through it. Navigation might be especially appropriate in a three-dimensional virtual architecture model that the user might enlarge to full scale so that it surrounds him and the currently deployed *screens* and *control panels*. He may also opt to “put away”, dim , or close down some of the deployed *screens*, *control panels*, and *directories* to floating *icons*.

2. INFOSPACE COMPONENTS

2.1 Space

The environment in which the *InfoSpace* concept operates is the virtual representation of infinite, three-dimensional space called *space*. This is the three-dimensional canvas upon and within which all information interaction takes place and parallels in three dimensions a sheet of paper in two dimensions or with the flat, scrollable computer screen or window. This is accomplished with the use of *stereoscopic* headset displays or eyephones. Each image is split into two and digitally reprojected to each eye, thus simulating the three-dimensional view we have of the real world.

The user, seated at a workstation, will don eyephones or the appropriate headset and will launch *InfoSpace* (see Fig. 1). This will be accomplished through a simple hardware and software startup. *InfoSpace* differs somewhat from other virtual environment interaction scenarios that concentrate heavily on physical interaction via a dataglove, which the user wears and uses to virtually touch and *manipulate* elements in the virtual environment. While this interaction device may be extremely useful and appropriate in certain applications, such as remote manipulation, virtual sculpting or building, etc., it may prove to be awkward and inefficient for rapid, repetitive command and control of an informational system. In order for an interactive informational universe to function effectively and naturally, it needs to be "driven." This will require a physical interaction system capable of accepting the users physical, interactive input in a variety of ways.

While the mouse is a very effective interaction device in windowing workstation configurations, it lacks many degrees of control that would be useful in a virtual three-dimensional information space. Ergonomically-designed, multi-functional hand controllers in addition to a keyboard for textual input will serve much better. These controllers may include a joystick or other three-dimensional interaction device for applications where that type of interaction would be appropriate. Foot pedals may also be helpful in providing yet another channel of control and physical interaction. The eyephones or headset may also incorporate some form of eye tracking to indicate the current center of attention. Linked together with the physical interaction

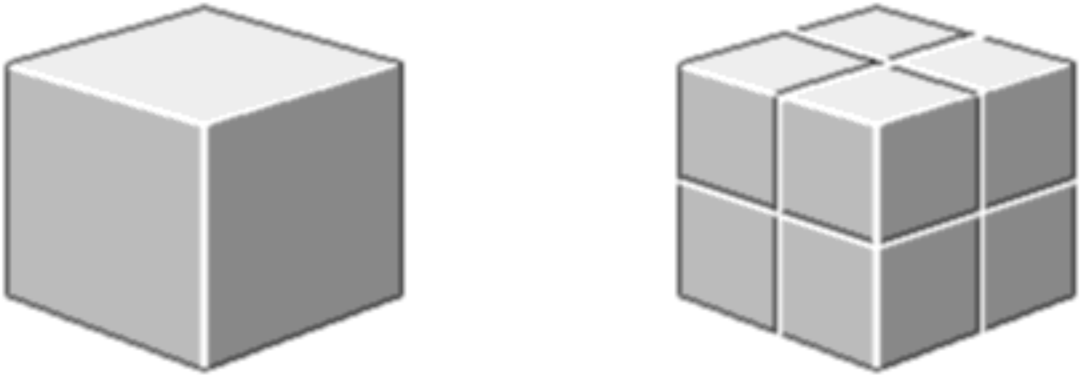


Figure 2: **Blocks and SuperBlocks** Functioning as a leaf node, the *block* (left) contains information, but does not contain other *blocks* or *matrices*. The *superblock* (right) contains other *blocks* or *matrices*. These are the primary informational components of *InfoSpace* and are analogous to the document and folder in two-dimensional graphical user interfaces.

controls, this could control the movement of a cursor or crosshairs that the user would aim here and there within the *space*, locking on various targets (*screens*, *control panels*, etc.) and move them around, forward and backwards in *space*, etc.

The *physical interface* controls may be represented in the virtual *space* as outlines to provide spatial feedback to the user. Nomenclature or labeling may also be displayed on these virtual representations of the *physical interaction controls*. This could be very appropriate in that many of the *physical controls* could be context-sensitive, or perform different functions in different situations. This would be very helpful to many users in complex environments, especially novice or infrequent users. The system may also track the user's hands, etc. and display representations of them for further visual spatial feedback, much in the same way the dataglove is projected virtually into three-dimensional space in other virtual environment applications.

To aid the user in managing spatial layout and arrangement and avoiding a non-orderly clutter of *screens* and other elements, *InfoSpace* will employ various *sweep-up* and *arrange routines* that the user can *activate*. These will consist of standard default routines as well as user-defined preferences.

Depending on the processor's speed and ability to project and animate multiple dimensional elements within *InfoSpace*, high-resolution rendering may be limited to the current element of attention or current focus. Therefore, as the user focuses on an element, it is instantly rendered in high-resolution, while all other elements are represented in low-resolution "cartoon" form. Elements may also be restrained to spatial movement along specific tracks or axis' in order to simplify spatial computation and projection.

2.2. Blocks, SuperBlocks and Matrices

The principal organizational unit of data or information within *space* is the *block* (see Fig. 2). A *block* is essentially a three-dimensionally projected cube in *space*. This *block* may represent any possible form of data or group of related data. *Blocks* may be displayed together in various ways within an organizing *matrix*. Just as a *block* may be a part of a *matrix*, that *matrix* may represent one *superblock* of a *supermatrix* (higher level *matrix*). Also potentially existing within any *matrix* can be one or more *superblocks*. A *superblock* represents a *submatrix* (lower level *matrix*) (see Fig. 2). The metaphor used here is that of super-systems, systems and sub-systems, or structures within structures within structures, etc. Information is made available to the user via the *uptrace panel* (see Fig. 9) to indicate where within the data structure he or she is and where it is possible to go *matrix*- or *block*-wise.

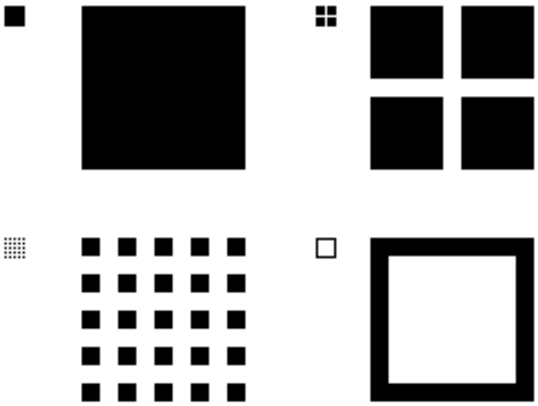


Figure 3: **Identifier Glyphs** Identifier glyphs representing the four major informational levels within *InfoSpace* (shown small and large). Block (top left), superblock (top right), matrix (lower left), and system (lower right). In addition to acting as identifiers of informational files or units within directory lists, etc., these glyphs can be activated to flash the corresponding object in the user's Current Space.

At certain times (while interacting with extremely complex or large *matrices* for instance), the user may wish to alter the display of the *matrix* or *blocks* by any of several ways in order to see and work with it more effectively. The entire *matrix* can be scaled up or down in size. The *blocks* can be scaled up and down in size (down to points) while the *matrix* retains its size. The *blocks* can

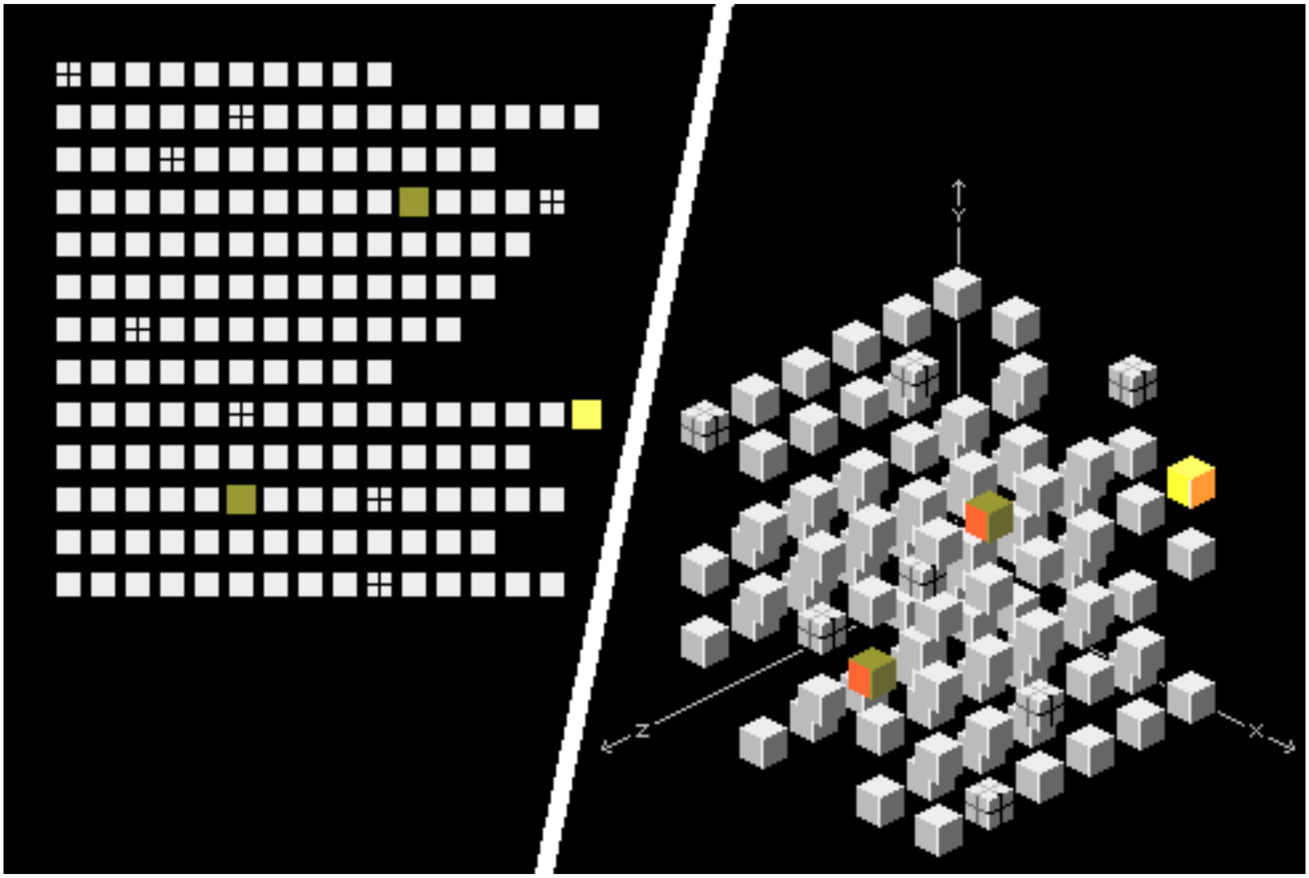


Figure 4: **2D and 3D Matrix Configurations** Shown above are two ways the user may choose to display the *blocks* and *superblocks* that exist on any particular level. On the left is the two-dimensional *storyboard configuration*. This may be useful and/or desirable when the user has no particular need for visualizing a three-dimensional spatial relationship among the elements. Two-dimensional relational layout, ordering and sorting will be available with the *storyboard configuration* in addition to simple non-relational ordering. On the right is the three-dimensional *matrix* configuration. In this configuration the *blocks* and *superblocks* are projected three-dimensionally into *InfoSpace*. Spatial layout of the *matrix* involves assigning a three-dimensional address to each of the elements within. In addition to simple non-relational layout and ordering, each of the elements can be relationally ordered and *addressed* according to *matrix* parameters, which are orders or values assigned to each of the three *matrix axis'*.

remain the same size and the *matrix* spacing increase or decrease. All of these display settings can be adjusted independently or in combination with each other. There will also be user-changeable default display settings for the creation of new *blocks* and *matrices*. In addition to viewing the *matrix* in a three-dimensional configuration, the user will be able to reconfigure all the elements in any or all levels to a two-dimensional or *storyboard configuration* (see Fig. 4). This spatial configuration may be more appropriate for some users or information systems, especially those without the need for visualizing three-dimensional spatial and cognitive relationships.

2.3 Systems

All the *submatrices*, *matrices*, and *supermatrices* that are linked together become a *system*. Separate *systems* can be linked together and access and information from them can be gained while the user is within one particular *system*. The *system* the user is currently in will be displayed at the top of the *system uptrace panel* (see Fig. 9) and will have in the left margin of its label, an *identifier glyph* for "*system*" (see Fig. 3).

2.4 Directories

Directories are floating lists that contain the names and information about the contents of a *system*, *matrix*, *superblock* or *block* (see Figs. 6 and 7). If a user wishes to know what information is contained in a *block* or *superblock*, he or she selects it and a summons a *block* or *superblock directory*, which appears and lists all pertinent information about the selected element. Each element listed in a directory has in its left margin an *identifier glyph* indicating what type of element it is (*block*, *superblock*, etc.) Obtaining *directory* information in this sequence is called the *explore mode* (see Fig. 6).

If the user wishes to locate a specific *system*, *matrix/superblock* or *block* of information, he or she will summon *system directories*, *matrix/superblock directories*, or *block directories* directly.

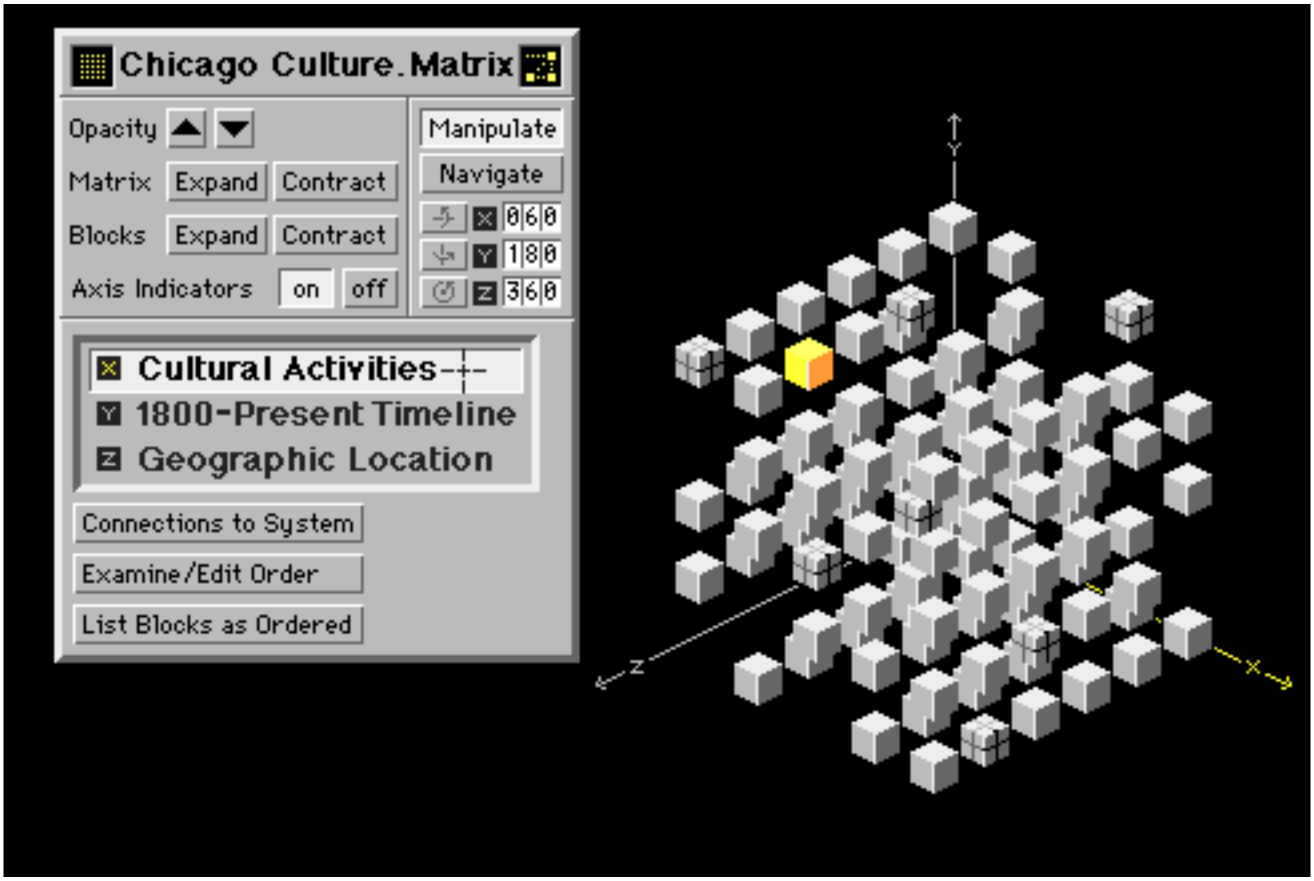


Figure 5: **Matrix Configuration Control** A possible *matrix* configuration. The *matrix* is arranged by assigning an address to each of the *blocks* or *superblocks*. This address is based on a comparative value or order for each of the three spatial axis', X, Y & Z. e.g. if the *matrix* represents information about city of Chicago, Illinois, the X-axis may represent cultural information ordered in some fashion (1 - Art, 2 - Music, 3 - Theatre, etc.). The Y-axis may represent a time line ordered by dates going from the past to the present (1 - 1800, 2 - 1820, ...the present), and finally, the Z-axis might represent geographical location (1 - inside the downtown Loop area, 2 - along Lake Michigan, 3 - within the city limits, 4 - suburban locations, etc.). As different value scales or orderings are attached to the axis', the whole configuration of the *matrix* will change. This will utilize animation to show the *blocks* and *superblocks* changing position and may involve some leaving the *matrix* or being called in from the *system database* if specified by the user. In addition to manipulating the *matrix* (rotating or tumbling), and *navigating* ("flying" through the matrix), the user will also be able to expand or contract a *matrix*, vary the size of the *blocks* and *superblocks* within a *matrix*, and perhaps vary the opacity of nearer *blocks* to see into the matrix. These abilities will enable the user to better visually understand what is contained within a *matrix* and its interrelationships. It may be desirable to vary the shape of the *blocks* and *superblocks* to indicate different kinds of informational files (i.e. text, audio, video, 3D, etc.), but in this paper only one form is shown for simplicity. Selection is indicated by the brightened or colored *block* or *superblock* and, along with opening and *activation*, is accomplished by actions similar and analogous to a click and double click (or menu command) as utilized in two-dimensional graphical user interfaces.

These will provide lists of all the *systems*, *matrices*, *blocks* or *superblocks* from which the user can find a specific area or piece of information. While in a *system directory*, the user may summon a *matrix directory*, and so on until he or she has located the target information. This process is called the *find mode* (see Fig. 7).

2.5 System UpTrace Panels

Going up and down through the various levels of a *system* would be very confusing if there were no way for the user to keep track of where he or she is and where he or she has been. This information will be displayed on the *system uptrace panel* (see Fig. 9). The *system uptrace panel* will show the highest level *system* that the user is within and will list the *matrices* in order of descending levels down to where the user is currently focused.

The user will also have access to *path histories* that will show in adjustable detail where he or she has been. In addition to displaying this *path history* as a list, with each element selectable so the user can go directly to a former spot on the *path history* if desired, the history may also be "played back" with forward and reverse controls. The user will see *screens* and *matrices* being opened, manipulated and closed, etc. in the same linear order as he or she originally controlled.

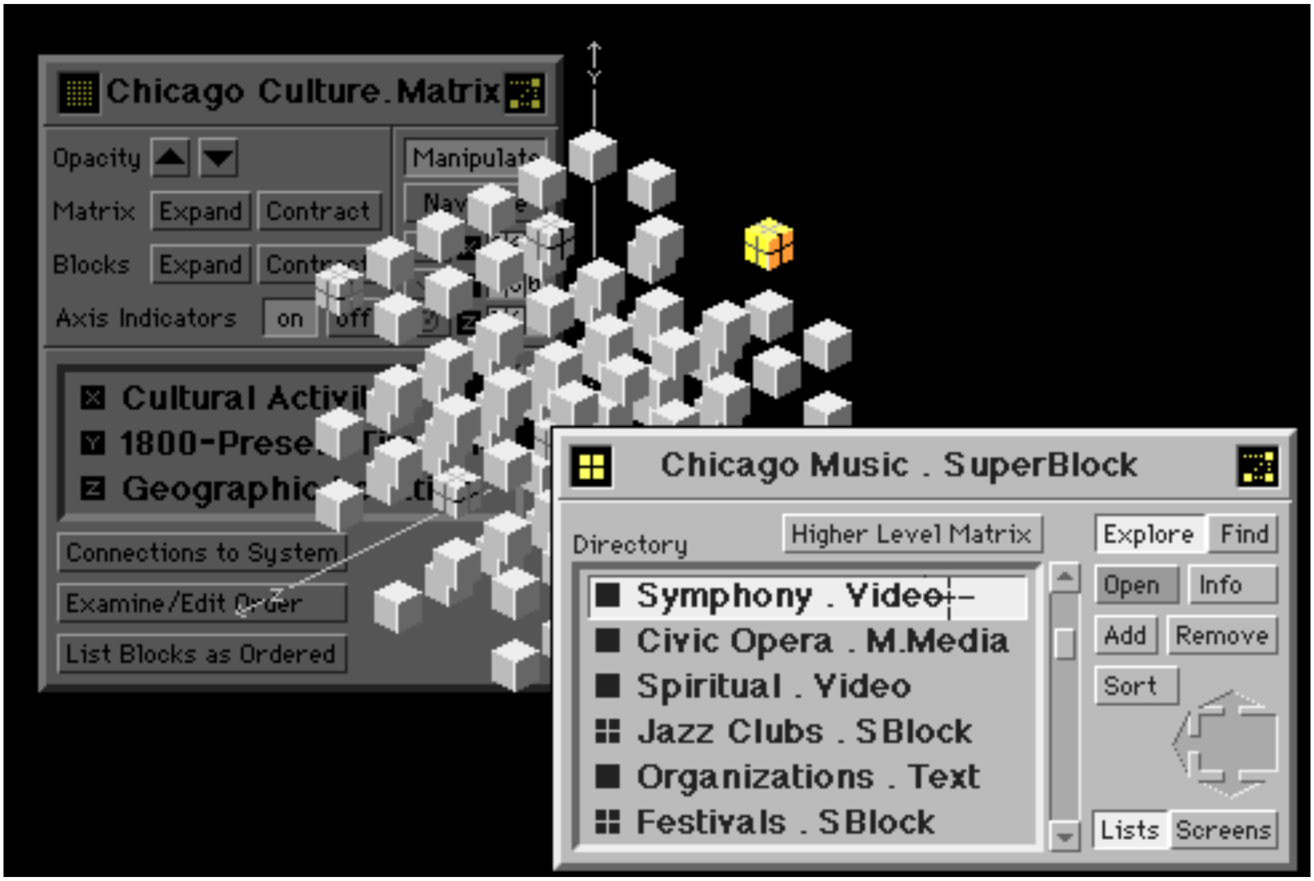


Figure 6: **Matrix/SuperBlock Directory** When a selected *superblock* is opened, a *directory* listing its contents is displayed. This *directory* can be moved about to suit the user's visual or organizational needs. *Blocks* and *superblocks* contained are denoted by an *identifier glyph* along with a filename and file type suffix. Interaction with this *directory* would follow methods used in two-dimensional graphical user interfaces regarding selection or opening of an item and in the redundant options for accomplishing an action such as directly or through a button. In the case of a *superblock*, which is in actuality a *submatrix* itself, it is possible to travel deeper into the *system* by replacing the current level *matrix* with the lower level *matrix* of the selected *superblock*. With the aid of the *system uptrace panel*, animation, and *review forward* and *review backward histories*, the user will travel both deeper and higher in the *system* without becoming confused or disoriented. Note how inactive or non-current *screens* or *panels* are dimmed.

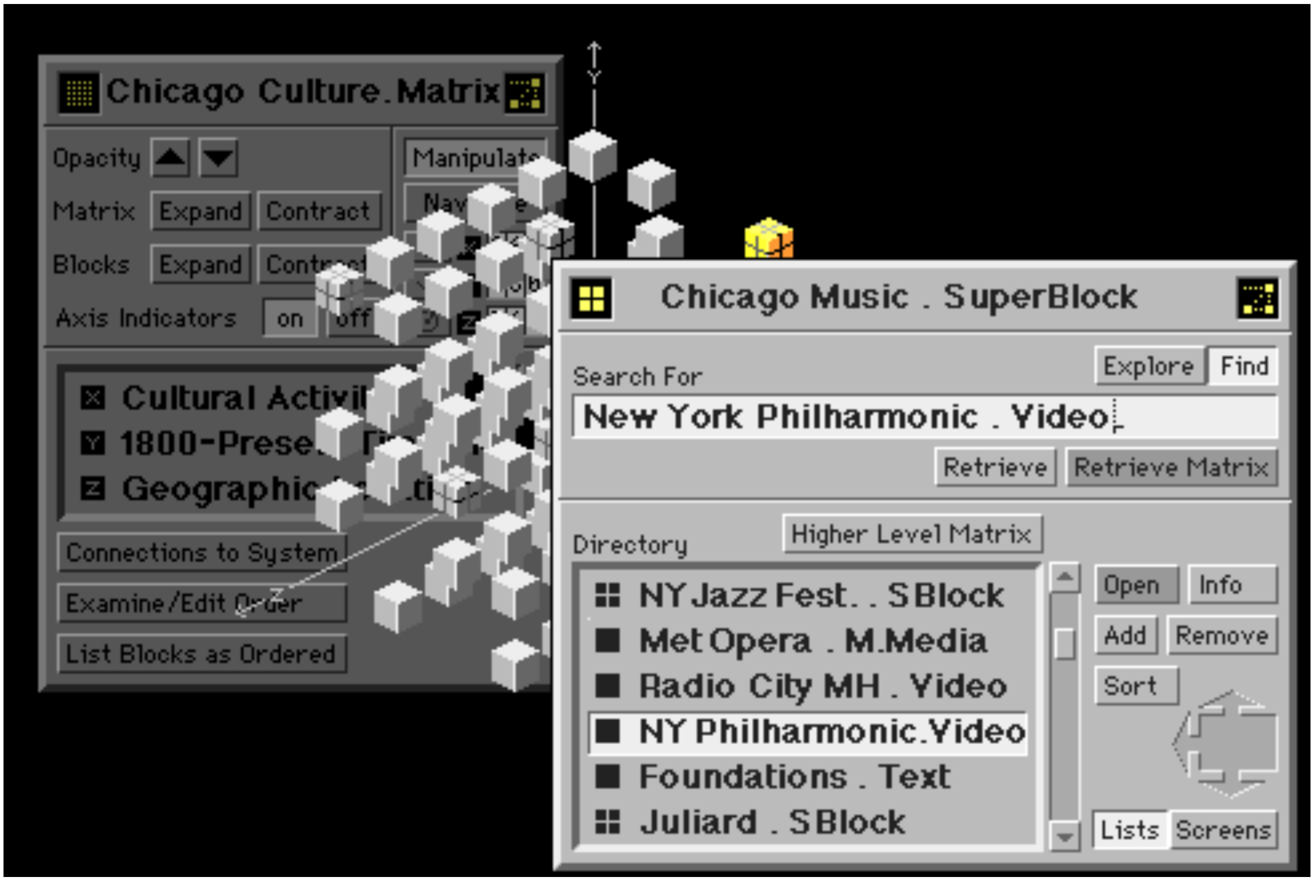
This feature may be especially useful when "canning" tours or walkthroughs for later presentation. With several people plugged into the same *InfoSpace*, this could be a remarkable teaching tool, with one user acting as the "driver" and others experiencing *InfoSpace* passively or with limited control options. "Canning" tours or sequences while allowing this partial user control may make better use of *hyper-connected* educational packages, as "wandering off" the hyper-path can often be problematic.

2.6 Screens

Screens are analogous to three-dimensional but flat-screen monitors that float in the *space* in front of the user (see Fig. 8). The purpose of a *screen* is to display two-dimensional information (Text, Two-dimensional graphics, video, etc.), or the two-dimensional representation of three-dimensional information [2D(3D)] (see Figs. 10, 11, 12 and 14). A screen appears when a *block* is opened by means of a cursor the user operates within the *space*. Regardless of the user's orientation to the *matrix*, the *screens* always appear so that the user is looking at them from the front.

Screens may have attached *control panels* that pertain to controlling and/or manipulating the data displayed on them. Screens may have scrollbars and can be resized and moved about (up, down, left, right, forward, backward) within the space. If the screen is 2D(3D) the user has two options. The 2D(3D) *block* can be viewed, manipulated within its 2D *Screen*, or the 2D(3D) *screen* can be activated to 3D so that the *space* is replaced by the *subspace* of the 3D *block*.

In the latter case the *space* and the *matrix* that the user is working with is replaced by the 3D *space* of the *block*, and the user finds him- or herself within the 3D environment of the *opened block*. While within this *subspace*, the user still has access to the *block/matrix* display, organizational and retrieval capabilities of *InfoSpace* due to its interlinked nature.



Search For: New York Philharmonic . Video

- NY Jazz Fest. . S Block
- Met Opera . M.Media
- Radio City MH . Video
- NY Philharmonic.Video
- Foundations . Text
- Juliard . S Block

Figure 7: **Find Mode** In this *directory* the user has chosen the find mode via the exclusive choice buttons, *Explore/Find*, near the top on the right side. Choosing the *find mode* enlarges the *directory* to include a text-entry field in which the user can type or otherwise input the name of an element to search for. The sought-after element can be retrieved either by itself with the *Retrieve button*, wherein it will appear as a *screen* if it is a *block* or a separate *directory* if it is a *matrix/superblock*, or as a *matrix* or element within one and be listed below in the *directory*. In the example above the user has entered New York Philharmonic.Video for retrieval. Upon *hitting* the *Retrieve Matrix button* (shown dimmed to denote the default selection), the *block* is retrieved as a selected element along with the other elements in its *matrix*. By *hitting* the *Info button*, the user will get an informational panel describing the location of this element. The user is then free to open the retrieved *block* or another in its *matrix* list, or open a *superblock* in its list thus going deeper into that *system*. Or, by hitting the *Higher Level Matrix button* located above the list, the user can go higher into that *system*. As with the previous *directory* in Fig. 6 and many other *InfoSpace* elements, the user may choose to view the retrieved *block* or *matrix* as *screens* and *directories* instead of as a list. This is accomplished via the exclusive choice buttons *Lists/Screens* at the lower right corner of the *directory*. When the *screens mode* is chosen, the *swap-space glyph* becomes active and the user can exchange the *space* within the *directory* with that of the *current space*.

2.7 Manipulators and Navigators

Within the *InfoSpace* environment, the user has access to controls known as *manipulators* and *navigators* to interact with the *matrices* or opened 3D *blocks*. If the user wishes to remain stationary in space and tumble or otherwise move the *matrices* or opened 3D *blocks*, this is known as *manipulation*. If, however, the user wishes to move about, through and around the *matrices* or opened 3D *blocks*, this form of interaction is known as *navigation*. This is accomplished through the use of a physical interface of hand and voice controls, including a joystick for *navigation*. Foot controls might be incorporated as well to take advantage of the entire body's coordination and participation.

All *physical controls* will be visualized (or hidden according to user preference) within the *Space*. This allows the user spatial feedback when using the *physical controls* and will aid in coordination and control of the interface.

Manipulation mode will deal with a manipulatable elements within the *current focus*. This might be a three-dimensional object on a *screen* or a three-dimensional object that is in the *current space*. *Manipulation* may also apply to the *matrix*, which may be tumbled or rotated in order for the user to better visualize its structure and the spatial relationships of its contents.

Navigation mode will apply mostly to three-dimensional elements within the *current focus* that can be moved through, as one would move through a *space*. This may be a very powerful method of understanding three-dimensional models or spatial arrangements such as *matrices* or certain applications like architectural models

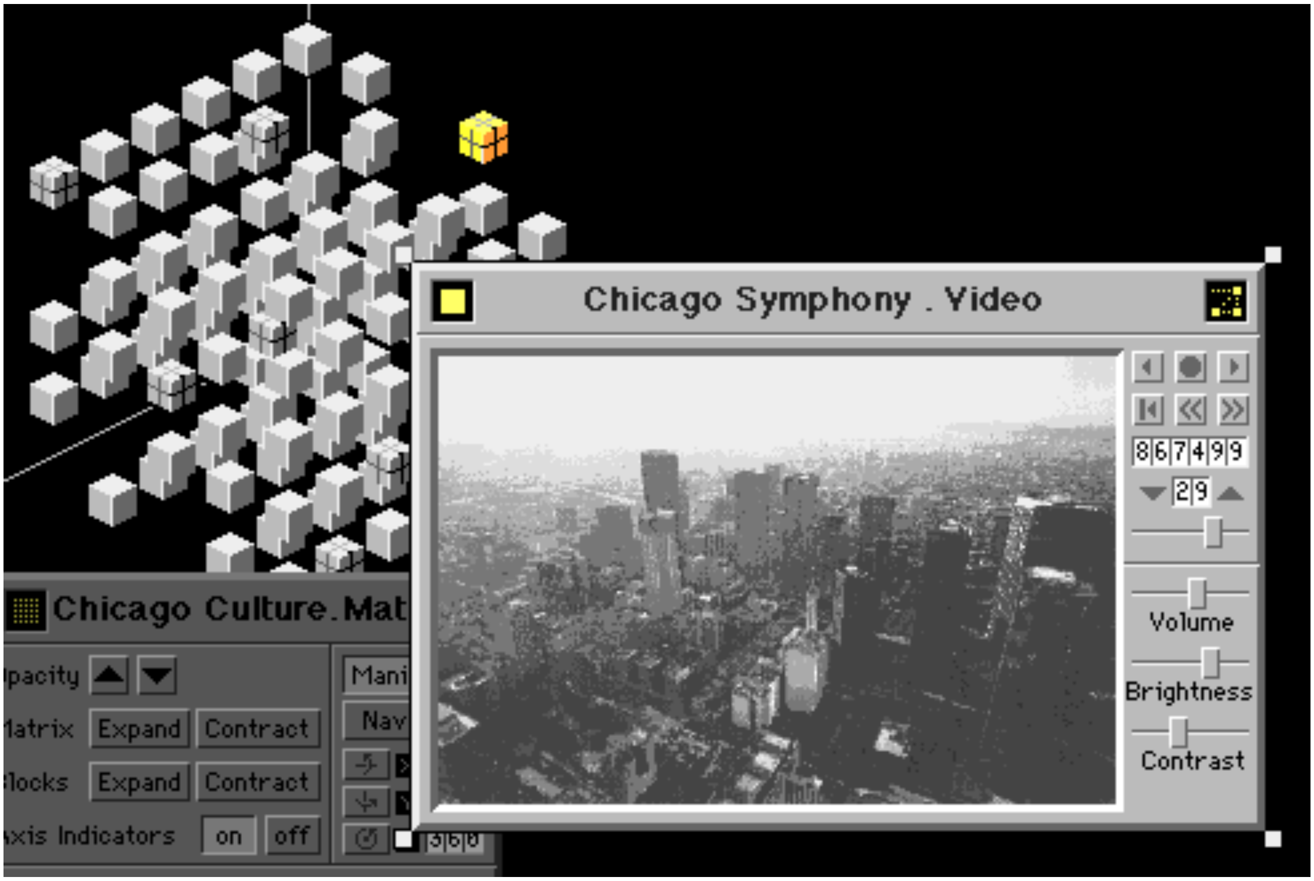


Figure 8: **Video Screen** When a *block* is opened, either from a *directory* or directly from the *matrix*, it appears as a *screen*. In the illustration above, the *block* opened contains a video file, so the *screen* reflects the type of *control panel* needed to operate a video. Similar appropriate *screens* would appear for text files, audio files, 3D files, multi-media files, etc. It's very easy to imagine including controls to cut, paste, and otherwise manipulate and build other files included on these *control panels*.

or microscopic models or environments. This functionality is one of the most cognitively effective aspects of this interface. Not only will the user be able to visualize the informational database in spatially-related, three-dimensional, virtual form, but will be able to interact physically and cognitively with individual 3D models and environments in the same manner.

2.7.1 Manipulation and Navigation Control Panels

User control of *manipulation* and *navigation* will also be available via *control panels* (see two examples: Fig. 5 - a *matrix control panel*, and Fig. 11 - a 3D *object control panel*). These *control panels* will be operable via the user's *physical interface* and will provide information and status on control options and visual interactional feedback.

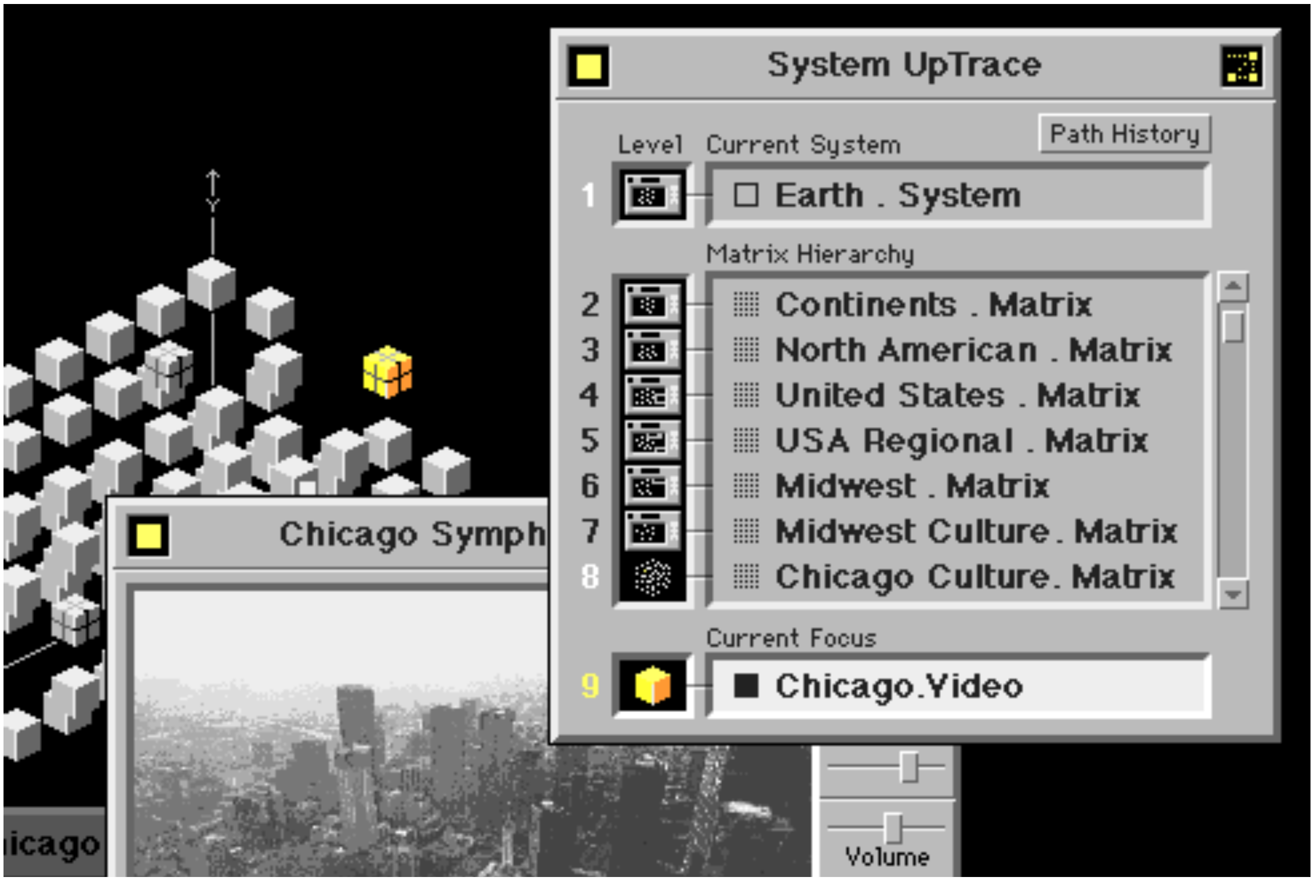
2.8 Hyper-Connections / Hyper-Connection Panels

Within *InfoSpace*, relational, trans-spatial connections can be made between elements. These connections will allow any element, be it a simple *block* or an entire *system*, to be connected to any other element or elements. These *hyper-connections* are recorded and accessible via *hyper-connection panels*.

When element "A" establishes a *hyper-connection* "out" to element "B", it is recorded as an *out-to hyper-connection* in the *hyper-connection panel* of element "A". When this *hyper-connection* is established, an *in-from hyper-connection* is simultaneously recorded in the *hyper-connection panel* of element "B." Access to this panel is gained by activating the *hyper-connection glyph* in the upper right corner of the *header panel* of most *screens*, *panels*, and *directories*.

A *hyper-connection panel* is available for any element that has established *hyper-connections*. The panel divides *hyper-connections* into two lists or groups: *out-to hyper-connections* and *in-from hyper-connections*. With the panel the user can easily see, establish or disconnect *hyper-connections.* The *hyper-connection panel* also operates similarly to directories, so the user can traverse *InfoSpace* in a similar fashion as the find mode in order to establish a *hyper-connection*.

Hyper-connections are an integral part of *InfoSpace*. Through them, the user is free to create and perceive vast neurally-intertwined networks of information. With the combination of dimensionally-projected tree diagrams and *system uptrace panels*, the user should be able to quickly and easily travel around the



System UpTrace

Path History

Level Current System

1 □ Earth . System

Matrix Hierarchy

2 Continents . Matrix

3 North American . Matrix

4 United States . Matrix

5 USA Regional . Matrix

6 Midwest . Matrix

7 Midwest Culture . Matrix

8 Chicago Culture . Matrix

Current Focus

9 ■ Chicago.Video

Figure 9: **System Up-Trace Panel** The *system uptrace panel* displays fields areas of information. The top field lists the *current system*, the middle field lists the hierarchy of *matrices* from highest level down to the lowest currently open, and the bottom field lists the *current focus.* The user can go instantly to any one of these listed elements or levels by *activating* the *icon* (the miniature representation of a *screen*, *matrix*, or *block*) to the left of the desired destination. The user may also choose to access the *Path History* function by hitting the *Path History button* near the top of the *system uptrace panel* just below the *header panel*.

systems and maintain a constant visualization of his or her location and spatial orientation.

2.9 Briefcases

As the user travels through *InfoSpace*, he or she may wish to "pick up" certain elements and bring them along with them. For instance, in building a multi-media presentation or file, the user may travel through many levels of many systems to gather elements for inclusion in the final piece. For this purpose the *briefcase* (see Figs. 17 and 19) serves as a method of collecting and carrying elements from one place to another. It is somewhat analogous to the clipboard or scrapbook found in two-dimensional graphical user interfaces, but differs in that there can be multiple briefcases and they can be transformed from or to *matrices*, either as copies or altogether.

This *briefcase* might be thought of as somewhat analogous to the clipboard in two-dimensional graphical user-interfaces, although in *InfoSpace* the user can have several *briefcases* and the *briefcases* can be *hyper-connected* to one another. A *briefcase* is in some respects a portable mode of a *matrix*. *Matrices* can be turned into *briefcases*, copied as a *briefcase* and, conversely, *briefcases* can be changed into *matrices* and inserted at any level into a *system*.

3. COMPONENT SPECIFICATIONS, BEHAVIOR, AND INTERRELATIONSHIPS

3.1 Blocks

3.1.1 Block Shape and Light Source Shading

Blocks, the principal information units in *InfoSpace* are represented by cube-shaped objects that are projected three-dimensionally into *space*. A simulated fixed light source serves to shade a *block* or group of *blocks* in order to provide the user a sense of spatial orientation.

In this conceptual model of *InfoSpace*, all blocks are in the generic form of a cube, however alternate shapes would most certainly

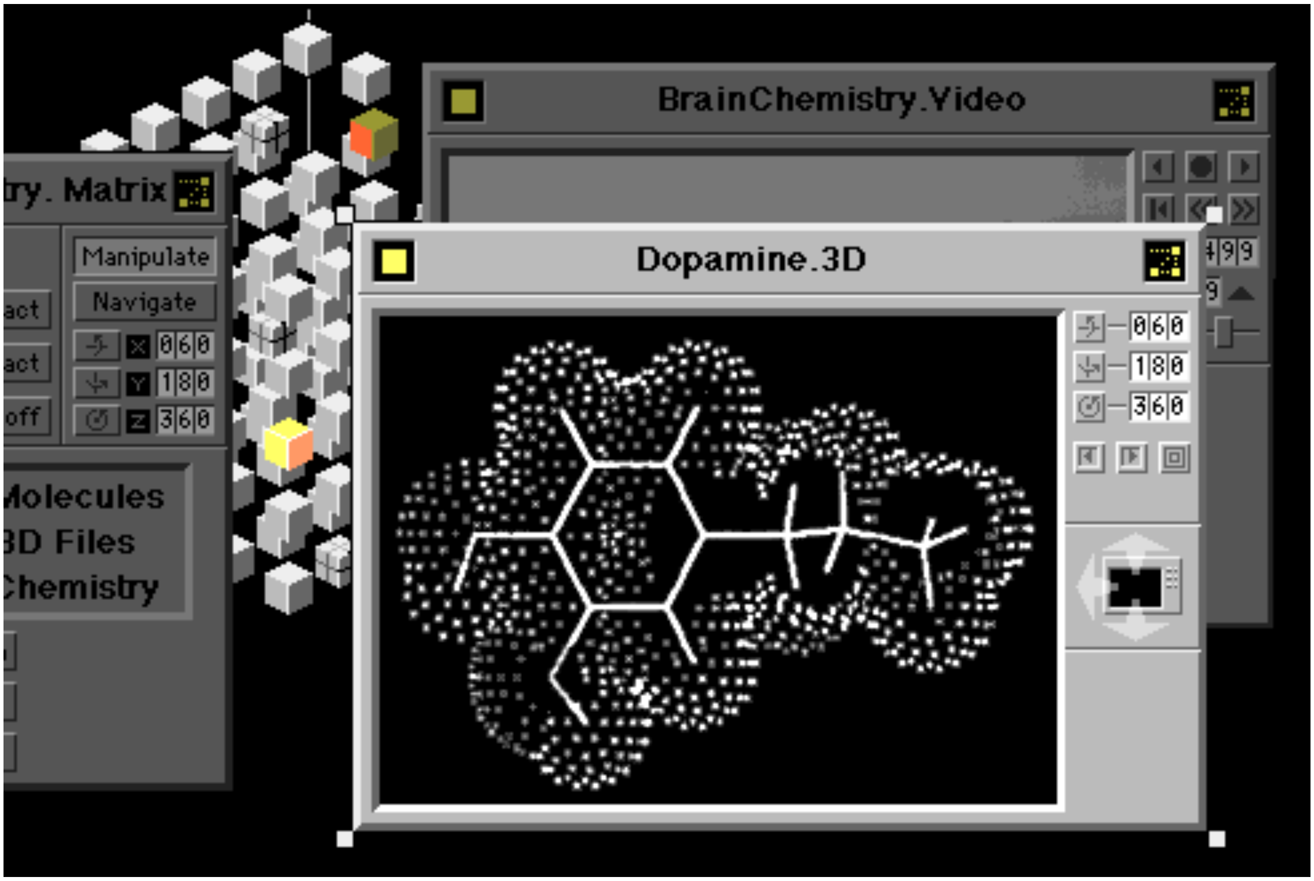


Figure 10: **3D File Screen** Three-dimensional files will be opened as *screens* much like videos and may even have video-like animation routines attached. The three-dimensional file will be able to be *manipulated* by the user with *axis* controls similar to those used to *manipulate* the *matrix*. The image will be stereoptically three-dimensional as well, with the user seeing into the screen. By *activating* the *"Swap-Space" glyph* on the *screen*, the *space* of the 3D file will replace the *space* of the *matrix* and the *matrix* will appear as a 3D image on a *screen* within the new *space* (see Fig. 11).

aid the user in distinguishing one type of *block* from another. Depending on computational power, speed and rendering ability, these *blocks* could be projected as tiny, detailed objects or simple geometric shapes. The user may actually want to switch back and forth between miniature representations of the real object or file and geometric representations in different situations. This flexibility is an important part of the nature of *InfoSpace* and greatly enhances the intuitiveness and effectiveness of the interface among many levels of user types and preferences.

3.1.2 Selecting and Opening Blocks

A *block* that contains information, but does not contain other *blocks* or groups of *blocks* (*matrices*) is referred to as simply a *block*. A *block* is rendered as a simple cube with smooth sides, as opposed to a *superblock* described further on. Selecting a *block* involves the user placing the Cursor over the desired *block* and signaling with the *physical interface* by some physical or voice means (most likely a single click or equivalent signaling method). Upon selecting a *block*, a floating *directory* can be summoned and will appear in front of the user listing the type and name of the information contained within the *block*.

Opening a *block* is the method by which the user gains access to the information within a selected *block*. This can be accomplished by either of two methods. First, the user may select and open a *block* simultaneously by placing the cursor over the desired *block* and *activating* it. Depending upon the type of *physical interface* utilized by *InfoSpace*, this may be through a double click of a control or some equivalent physical or voice signal. If however, the user is *exploring a matrix* of *blocks* by randomly or methodically *selecting* them, he or she may *open* a *block* through the floating *directory*. This would be accomplished in the same manner as if the user were directly *opening* a *block* with the exception that the cursor would now be placed over the name of the *block label* displayed in the *directory*.

3.1.3 Screens and Screen Control Panels

When a *block* is opened by the user, a *screen* appears. This *screen* will have a *header bar* that will contain a *source glyph*, the name of the *block* and *block* type, and a *hyper-connection glyph*. The visual appearance and configuration of the rest of this screen will be determined by the *block's* type or application. Hence, a *block* representing a video file would be opened as a video screen with a *control panel* appropriate to its function and control (see Fig. 8).

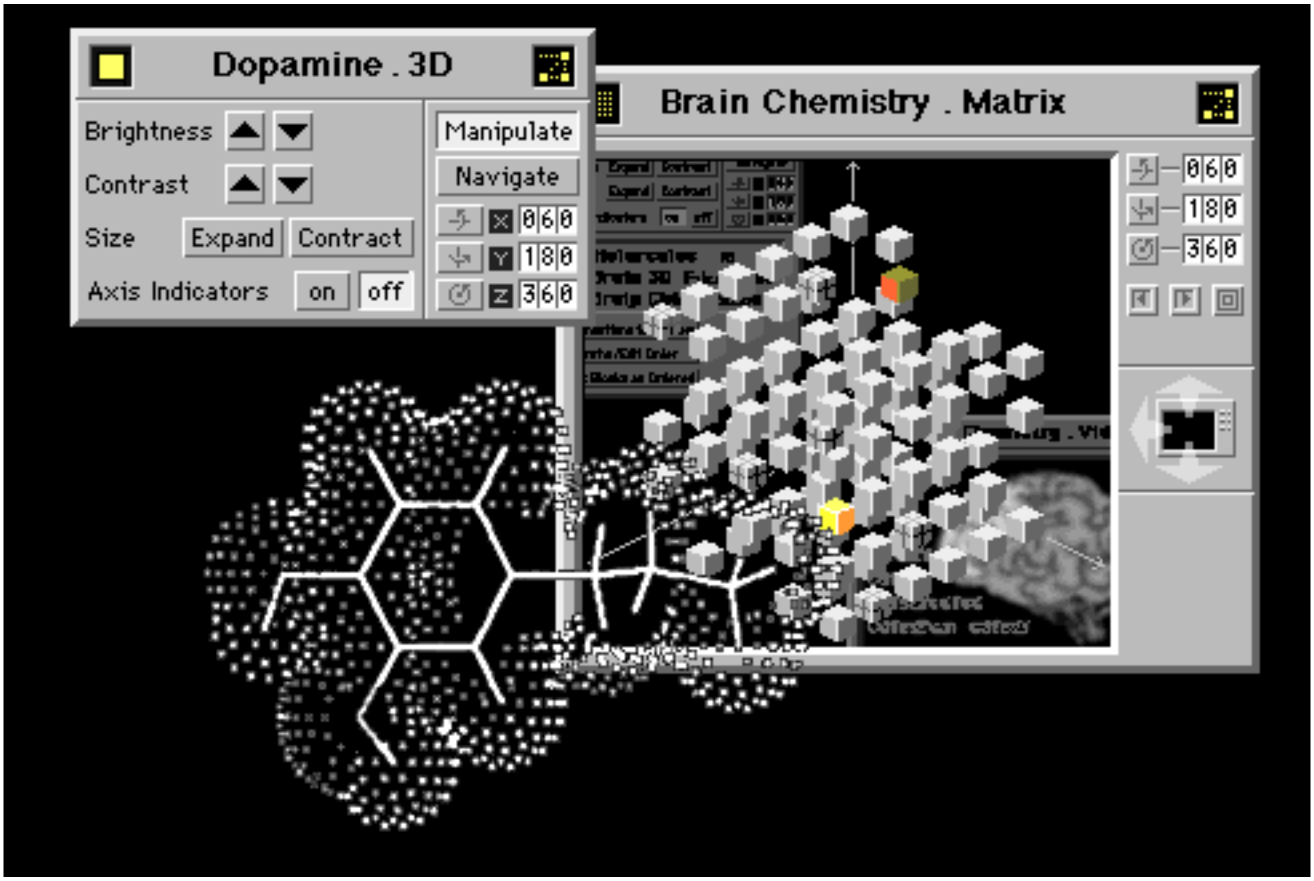


Figure 11: **Swap-Space** In this illustration, the object in the 3D file and its *space* have, through the *activation* of the *swap-space glyph*, become the *primary* or *current space* for the user, with the *matrix* and its *space* transferred to a *screen*. The *matrix* is now *manipulated* by a *control panel* on its *screen* and the 3D file is *manipulated* by a *control panel* that appears when its *space* becomes the *primary space*. *Stereopsis* will aid in distinguishing between the 3D objects and *screens*. If the user *hits* the colored *identifier glyph* in the upper left corner of the *3D object control panel,* its corresponding *block* will flash on and off in its *matrix* (which is now visible in the space of the Brain Chemistry.Matrix screen). If the user *activates* the *identifier glyph*, the *control panel* and the 3D object will disappear and return to the *block* in its *matrix*, whereupon the user can then *activate* the *swap-space glyph* and return everything in the *screen's space* to the *current space*. This swap can also take place without the 3D object being closed down to its *block*, leaving the user where he or she began (see Fig. 10).

In *blocks* that represent 3D objects, upon opening to a *screen*, the *control panel* will enable the user to *manipulate* or *navigate* the 3D object or model. In an open 3D screen, the contained object or model will appear *stereoscopically* through the *screen's* window. These *screens* will also have an active *swap-space glyph* that will allow the user to replace the *current space* with that of the 3D object's or model's *space*. When this occurs, the *3D control panel* will appear as a separate *control panel* in the *current space* along with the object or model (see Fig. 11). This *control panel* can then be repositioned by the user for optimum access and visibility.

3.2 SuperBlocks

3.2.1 SuperBlock Shape and Light Source Shading

A *Block* which represents a *matrix* of *blocks* (this would be anywhere from two upwards) is designated as a *superblock*. A *superblock* is rendered as a cube with each of its six sides divided into four equal areas by a horizontal and vertical line. As resolution permits, these lines are rendered as to look chiseled into the surface of the *superblock* and are highlighted by the *light source*.

As was stated about the shape of the *blocks*, there may be different shapes, textures, or colored *superblocks* in order to give the user additional visual clues about their identity and contents without having to open and examine them.

3.2.2 Selecting and Opening Superblocks

The methods used for selecting and opening *superblocks* are the same as for regular *blocks*, however, the results are different. In selecting a *superblock*, the user gets a *directory* listing the names of the *blocks* in the *matrix* contained within. For each element in this list there is also an *identifier glyph* located in the left margin denoting if it's a *block* or another *superblock* (see Figs. 6 and 7).

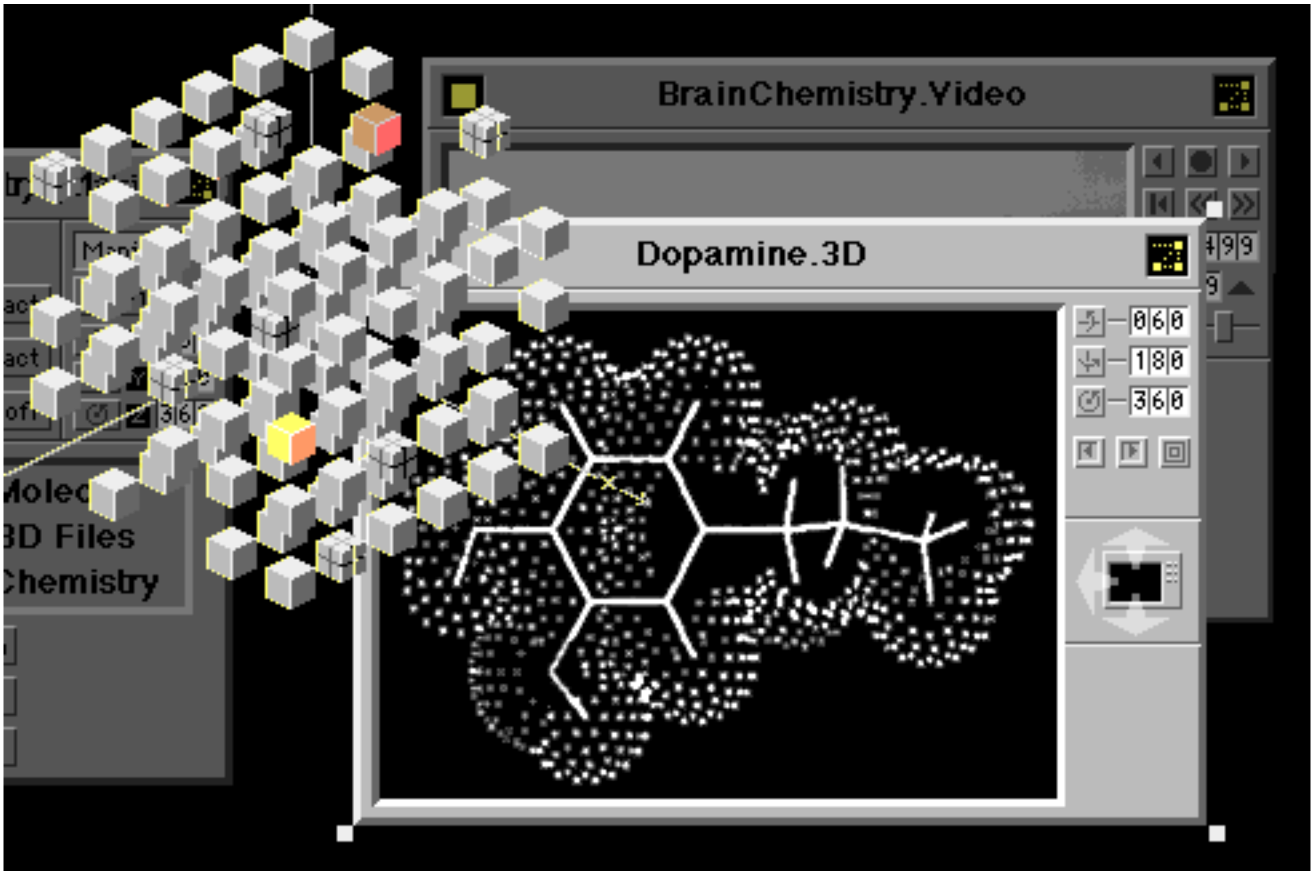


Figure 12: **Arranging Elements in the Current Space** Elements within the *space* can be brought forward or backward in *space* for easier visualization. In this illustration, the *matrix* has been brought to the front.

After selecting a *superblock*, the user may opt to open one of the contained *blocks* or *superblocks* directly from the *directory*. If the user opens a *block* in this manner, a *screen* appears with the information contained in the *block*.

If the user opens a *superblock*, a *screen* appears with a 2D(3D) representation of the *matrix* contained within that *superblock*. The user then has the option of *activating* that *screen* to 3D wherein the *space* of the *superblock's submatrix* will replace the *matrix* the user was previously in. The user will now find him- or herself in a new *space* or what was formerly the *subspace* within the former *matrix* or what is now the *supermatrix*.

Note that the prefixes of Sub- and Super- do not refer concretely to any particular *matrix* or *block*, but are always used in a relative context according to where the user finds him- or herself within the *InfoSpace* environment.

3.3 Matrices

Matrices are the organizing structures that *blocks* fit into within a *space*. *Matrices* are formed by arranging *blocks* and *superblocks* within them according to three axis' or addresses based on an X, Y and Z location in the *space*. This means that the values or orders for the three *parameters* will determine the location or address of a *block* or *superblock* within the *matrix*. These three *parameters* can be changed to reorder and hence reshape the *matrix*. *Parameters* can be almost any form of classification that can be valued or ordered in some manner. All *blocks* and *superblocks* within a *matrix* must contain *values* or *orders* for the *parameters* to be used in determining addresses.

There will be utilities available for assigning *orders* or *values* to all *blocks* and *superblocks* in a *system* or set of *systems*. As individual *blocks* and *superblocks* are created, *screens* will appear that will ask the user to define all *orders* and *values* used currently within that *system*.

Information will appear on the user's system *uptrace panel* (see Fig. 9) that will list the *matrix level* of the *system* that is currently being viewed and allow the user to see where he or she is within *InfoSpace*, where he or she has been (by utilizing the *path history function*), and allow travel to *hyper-connected* points and levels.

Matrix directories will also be available to the user in case he or she wishes to see a listing of all the *blocks* and *superblocks* in a *matrix* or in a *system* and *find* them rather than *explore* for them. *Finding* refers to the process of searching for a *block* or *matrix* that the user can *summon* by means of a *block*, *matrix* or *system directory*. *Exploring* refers to the sequence of getting information on selected *blocks* or *superblocks* with a *summoned directory*.

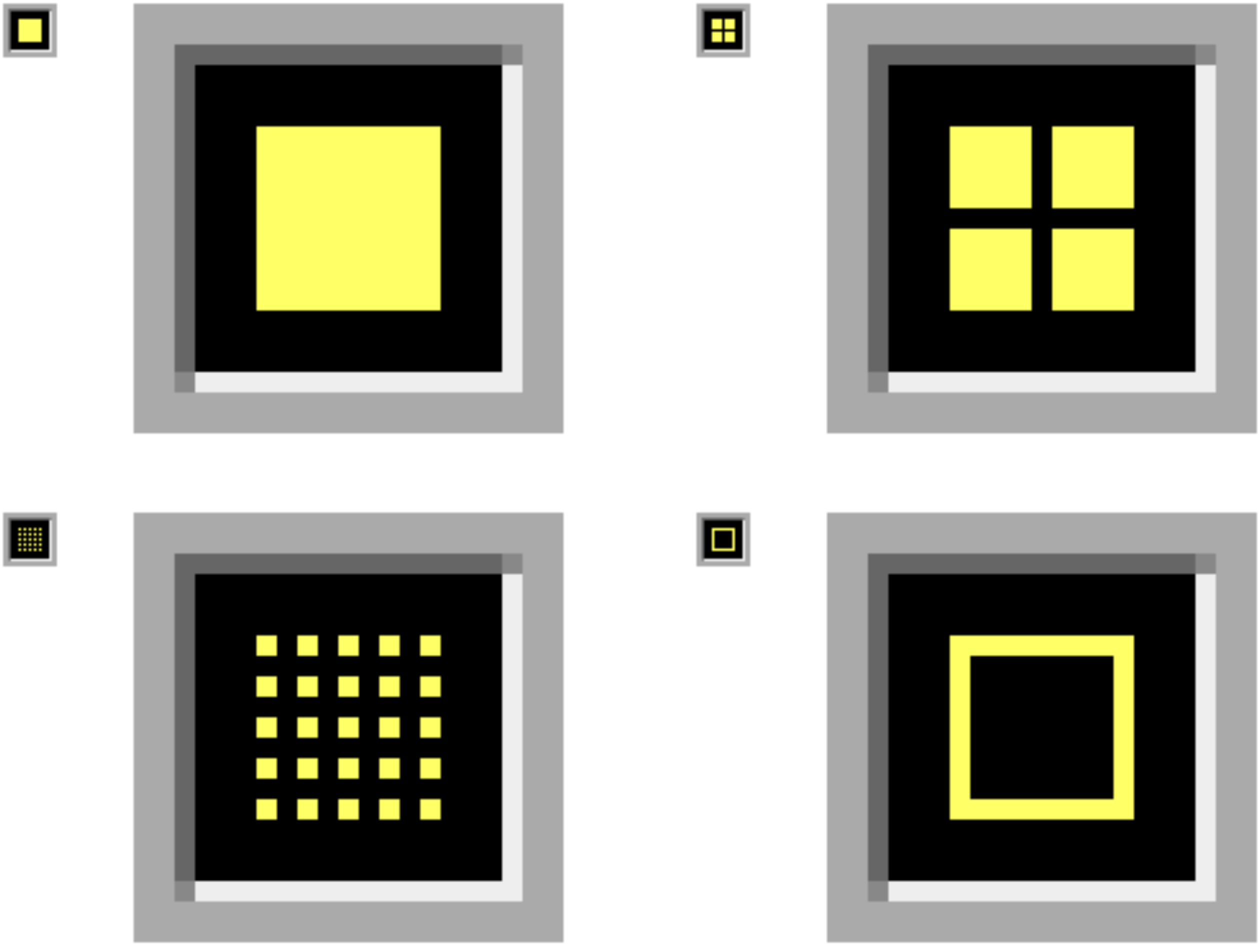


Figure 13: **Source Glyphs** The four variations of the *source glyph* found in the *header panel* of *screens*, *informational panels*, *control panels*, and *hyper-connection panels.* The *identifier glyphs* for *block*, *superblock*, *matrix*, and *system* indicate the source type from which the element displaying this *source glyph* came, i.e. (counter-clockwise from top, left: a *block*, a *superblock*, an open *matrix*, or an entire *system*. This *glyph* functions in two ways. The first function is accessed by selecting it, (i.e. analogous to clicking on and holding it) the *glyph* will flash and its corresponding owner, either a single element, or set of elements, will also flash. This will aid the user in determining what is representing what, such as in finding the corresponding *block* for an open screen, *control panel*, *directory*, etc. The second function of the *source glyph* is accessed by *activating* it (i.e. double clicking or similar interaction) and the *glyph* then closes the *screen*, *informational panel, control panel*, etc. to its source. This closing will be accomplished with a visual animation of the *screen* simultaneously shrinking and returning to its source and will be the reverse of an opening animation. The *source glyph* can also contain a *briefcase* or *screen/directory symbol* (see Figs. 17, 18, 19, and 20).

3.3.1 Matrix Control Panels

When a *matrix* is opened by the user, a *matrix control panel* appears (see Fig. 5). When this occurs, the *3D control panel* will appear as a separate panel in the *current space* along with the *matrix* (or *storyboard*, if so configured). This *control panel* can then be repositioned by the user for optimum access and visibility. This *control panel* will have a *header panel* that will contain a *source glyph*, the name of the *matrix*, and a *hyper-connection glyph*.

A *matrix control panel* will be visually rendered and spatially configured similar to other *InfoSpace* elements. The *header panel* will include a *source glyph*, the name of the *matrix*, and a *hyper-connection glyph*. the rest of the *matrix control panel* will be divided into three sections.

The first section pertains to the opacity and scaling of the *matrix* and as well as that of its contents. The opacity can be controlled so that *blocks* or *superblocks* nearer to the user are rendered more transparently, thus allowing the user to see "into" a dense *matrix* more effectively. The *matrix* can also be scaled up or down in size or have its elements expanded or contracted in relation to one another. The size of the *blocks* and *superblocks* within the *matrix* can also be scaled up until they touch one another or down to points, without changing the spacing among them. The last control in this section are exclusive choice buttons for turning the *axis indicators* on or off.

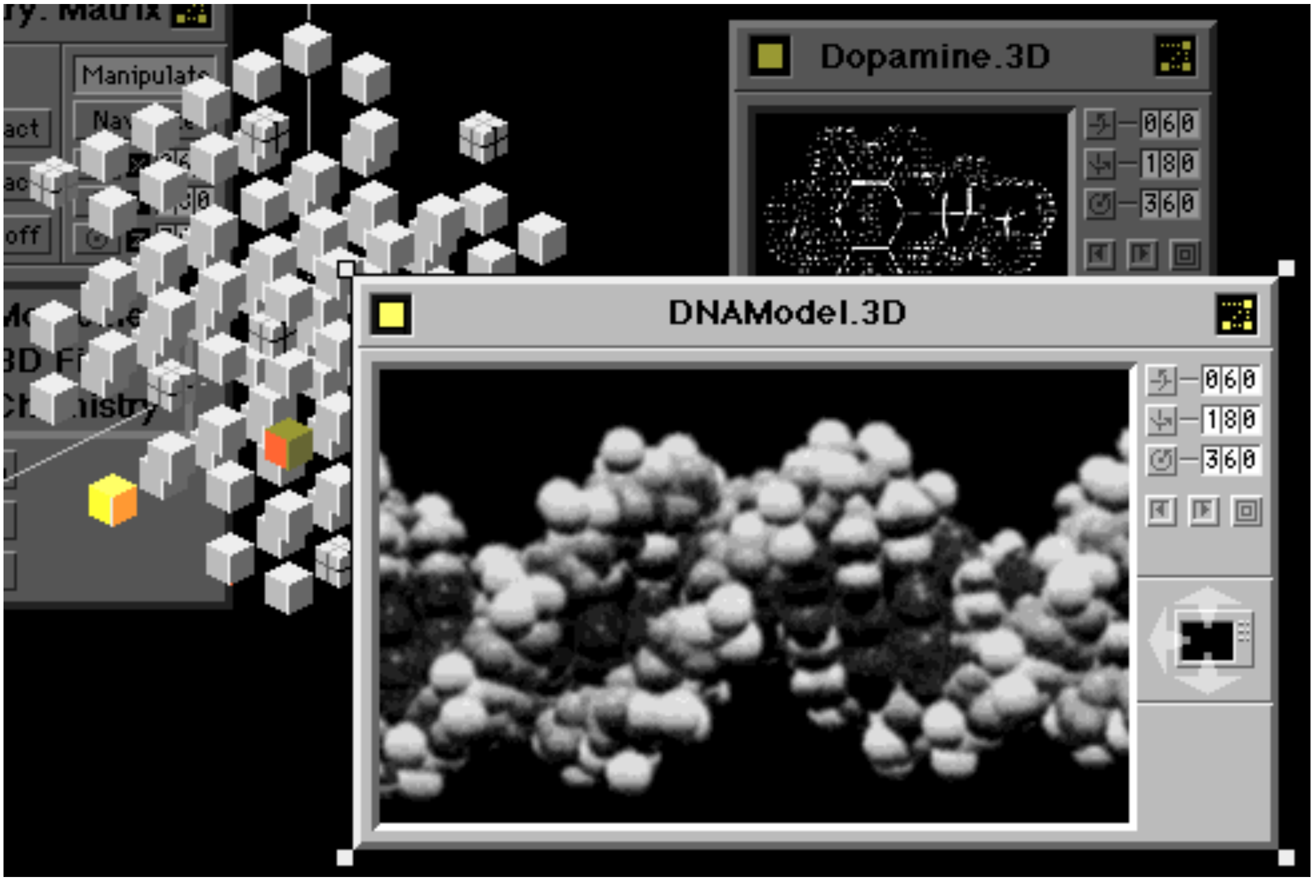


Figure 14: **Resize Squares** *Screens* can also be resized using the *resize squares* displayed at the four corners of current *screens*. In addition to the ability to be moved back and forth (X-axis) and up and down (Y-axis) in *space*, *panels*, *screens*, and other elements within a *space* can be moved fore and aft on a Z-axis perpendicular to the user's viewpoint and orientation. To avoid visual confusion and the necessity to continually re-position peripheral *screens* and other *panels* desired open, it will be possible to maintain their orientation to the user using *navigation* of a *matrix* or another three-dimensional file. Psychologically, this will be important to avoid confusing these elements with the *matrix* or three-dimensional file the user is *navigating*. This will avoid such unnecessary issues such as "What does the back of a *directory panel* look like?"

The second control section is located to the right of the first and controls the *manipulation* or *navigation* of the *matrix*. The *manipulation* or *navigation mode* is set via exclusive choice buttons. While in the *manipulation mode*, the user may tumble the *matrix* in *space*, thus allowing him or her to visualize the spatial relationships more effectively. *Navigation* will allow the user to "fly" in and around the *matrix*, thus giving a different type of spatial understanding about the *matrix*. This might be especially interesting in extremely complex *matrices* containing many elements.

3.4 Directories

3.4.1 Directory Visual Configuration and Behavior

Directories are floating lists that contain the names and information about the contents of a *system*, *matrix*, *superblock* or *block*. They appear as flat, but three-dimensionally rendered, with a *header panel* that will contain a *source glyph*, the name of the related element and element type, and a *hyper-connection glyph*. The *directory* will display an informational description (when representing a *block*) or a list of contents (when representing a *matrix/superblock*) in the area below the *header panel* (see Figs. 6 and 7).

In addition to the *physical interface* controls to which the user has access, there will also be *cursor operable controls* available on the *directories* through which the user may *open*, *save* or otherwise interact with a *directory* item.

A *directory* can be viewed in two ways, either as a list of the contents or as screens. The user will indicate which mode via two exclusive-choice buttons labeled *Lists* and *Screens*.

The *List mode* is very straightforward. It consists of a scrolling list of the contents. *Identifier glyphs* in the left margin indicate the type of element listed (see explanation Fig. 2). List items can be selected and opened singularly or in groups. This can also be done via the button controls to the right of the scrolling list.

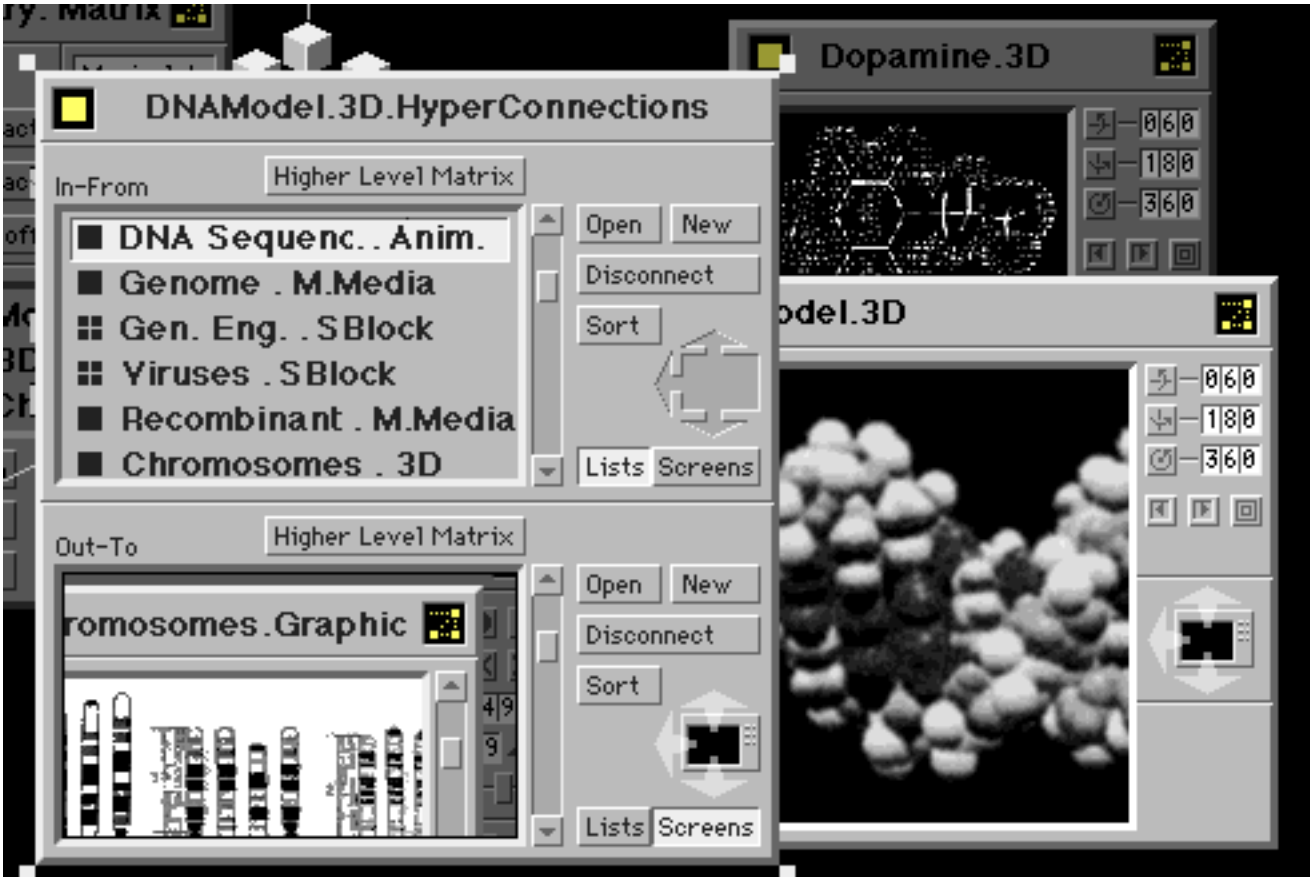


Figure 15: **Hyper-Connection Panels** The *hyper-connection panel* above belongs to the 3D *block* DNAModel.3D (shown as a *screen* directly behind the *hyper-connection panel*). It has been *summoned* through the user *hitting* the *hyper-connection glyph* located in the upper right corner of the 3D *screen*. In the top section of the *hyper-connection panel*, being viewed in the *list mode*, are *hyper-connections* to the DNAModel.3D *block* that are coming *in-from* other elements in *InfoSpace*. If the user were to open any of these files to the *current space*, or travel to it, and then *summon* its *hyper-connection panel*, DNAModel.3D would be listed as an *out-to hyper-connection*. The bottom section shows DNAModel.3D's *out-to hyper-connections* and is being viewed in the *screens mode*, thus the *swap-space glyph* is active.

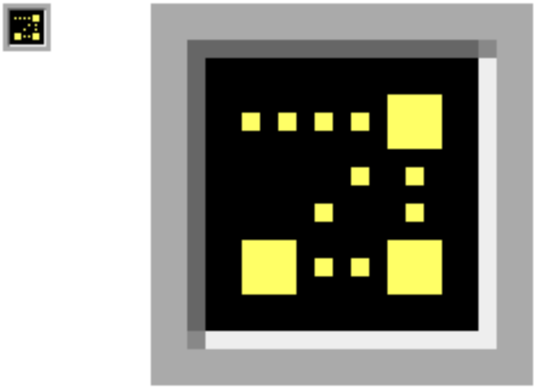


Figure 16: Hyper-Connection Source Glyph The *hyper-connection glyph*. When the user hits this *glyph*, located in the upper right corner of *hyper-connectable screens*, *control panels*, and *directories* in the *header panel*, the *hyper-connection panel* for that element will appear.

These buttons, in addition to giving the user another method of opening or *activating* an item also allow for the addition or removal of an item from the list. The *Info button* invokes a *screen* giving information that has been stored for the elements currently selected. Sorting of the list in a variety of ways is also possible via the *sort button* which will bring up a multi-choice list of sorting alternatives or can be set to function automatically via a user-configurable default setting. While the *directory* is in the *list mode*, the *swap-space glyph* is inactive, as it relates to the switching of the screen *space* (active and visible in *screen mode*) with that of the current space (see Figs. 8 & 9).

In the *screens mode*, the *directory* will display its contents in the form of *screens* in *space*. If the contents contain one or more *superblocks*, they will be displayed as *directories* themselves. *Screens* will be displayed in stacked order or can be spatially positioned differently by the user. While in the *screens mode*, the *sort* functions will operate similarly to the *sort* function in the *list mode*, however, rather than merely reordering a list, *sorting* will affect the spatial arrangement or stacking of the *screens* and/or directory contents in their *space*.

In addition to having the capability of resizing the *directory* to allow for easier viewing, while in the *screens mode*, the *directory's swap-space glyph* is active. This will allow the user to swap the *space* of the *directory* with that of the *current space* (see Figs. 10 & 11).

3.4.2 Explore Mode

Directories can be utilized in two different ways within the *InfoSpace* model, by operating them in the *Explore Mode* or the *Find Mode*. The user chooses one of these modes via an exclusive choice buttons labeled *Explore* and *Find* and are located at the top of the right-side controls on the *directory*. The first mode is the *explore mode* (see Fig. 5). If a user wishes to know what information is contained in a *block* or *superblock*, he or she selects it and a *block* or *SuperBlock Directory* appears that lists all pertinent information about the selected element. This mode is particularly helpful to the user when information is needed about various elements being viewed.

3.4.3 Find Mode

The second method or mode operates in reverse fashion (see Fig. 6). If the user wishes to locate or retrieve a specific *system*, *matrix*, *superblock* or *block* of information, he or she will *summon system directories*, *matrix directories*, *block* or *superblock directories* directly. These *directories* will provide lists of all the *systems*, *matrices*, *blocks* or *superblocks* from which the user may find a specific area or piece of information. A particular target is found through the process of elimination. The user starts by *summoning* a *directory*. Unless a specific type of *directory*, i.e. *system*, *matrix*, *block* or *superblock*, is requested, the *directory* that corresponds with the *current level* will appear. In either the *explore mode* or *find mode* the user can then use *cursor operable controls* on that *directory* to either *summon* a *Higher Level directory* or a *Lower Level directory*. By *activating* the *Higher Level Matrix button* above the *directory* list or window, the list or window will change to show the contents of the next higher level *matrix*. Going lower in level is as simple as selecting and opening any *superblock* listed in the *directory list* or window. Thus, the user may essentially zoom out to get a macro view or listing of the systems, or zoom in to get a micro view or listing of the information in a single *block*.

When in the *find mode*, a user knows exactly what he or she is wanting to find or retrieve. The name can be typed into a field on a *directory* and using the *cursor operable controls* the user can go right to it or its level. If partial information about a specific level or *block* is entered into this field, a *directory* listing possible matches will appear and the user can narrow it down from there using the process described above.

3.5 Hyper-Connections / Hyper-Connection Panels

3.5.1 Inter-Element Hyper-Connections

Within *InfoSpace*, the user may establish trans-spatial relationships or hyper-connections between elements. Any *InfoSpace* element, from a single *block* to an entire *system*, may be *hyper-connected* to any other element or elements. Information about and control of this feature is handled via a *hyper-connection panel* (see Fig. 15). Every element that can have *hyper-connections* will have access to a *hyper-connection panel* via the *hyper-connection glyph* (see Fig. 16) located at the upper right corner of *screens* and *panels*. By *hitting* this *glyph*, the *hyper-connection panel* for that element will appear as the *current focus*. As with other *InfoSpace screens* and *panels*, it is resizable and can be positioned spatially for optimum viewing.

Through *hyper-connections* the user can travel to any *hyper-connected* element, or *summon* any *hyper-connected* element to the *current space*. As with all moves through *InfoSpace*, this will be recorded in the *path history* and the user will be able to travel easily back and forth as desired without becoming disoriented or lost.

3.5.2 Hyper-Connection Panels

The *hyper-connection panel* is structured, rendered, and operated in a similar fashion to a *directory*. This panel will have a *header bar* that will contain a *source glyph*, and the name of the element it's representing and its type. The area below the *header panel* is divided into two sections. The top section is labeled *In-From* and displays information pertaining to *hyper-connections* to the element of *current focus* coming *in-from* another element or elements. Like a *directory*, this display can be viewed either as a sortable listing or as a collection of *screens* and *superblock directories*, or as a *matrix*. When *in-from hyper-connections* are viewed as *screens* or as a *matrix*, the *swap-space glyph* becomes active and the user may exchange the *space* of the *in-from* section with that of the *current space*.

The *control panel* area to the right of the *in-from hyper-connections* allows the user to open any existing *hyper-connected* element, establish a new *hyper-connection*, disconnect any existing *hyper-connection*, and *sort* the list or collection in ways similar to the *sort* function in *directories*. The *Higher Level Matrix* button also functions similarly to the same button found on *directories*, allowing the user to move up through the *system* levels in order to find a desired *block* or level. Also similar to the method used in directories is moving down through *system* levels. In both *directories* and *hyper-connection panels* the user accomplishes this by simply selecting and opening any element listed as a *superblock* or displayed in the *screens mode* as a *matrix* or *matrix directory*.

The bottom section is labeled *Out-To* and displays information pertaining to *hyper-connections* from the element of *current focus* going *out-to* another element or other elements. Just as in the *in-from* section, this display can be viewed either as a sortable listing or as a collection of *screens* and *superblock directories*, or as a *matrix*. When *out-to hyper-connections* are viewed as *screens* or as a *matrix*, the *swap-space glyph* becomes active and the user may exchange the *space* of the in-from section with that of the *current space*.

The *control panel* section of the *out-to* section functions identically to the *control panel* section of the *in-from s*ection, and actions performed in this section will result in concurrent changes in the *hyper-connection panels* of affected elements, just as occurs in performing *in-from hyperconnection* actions.

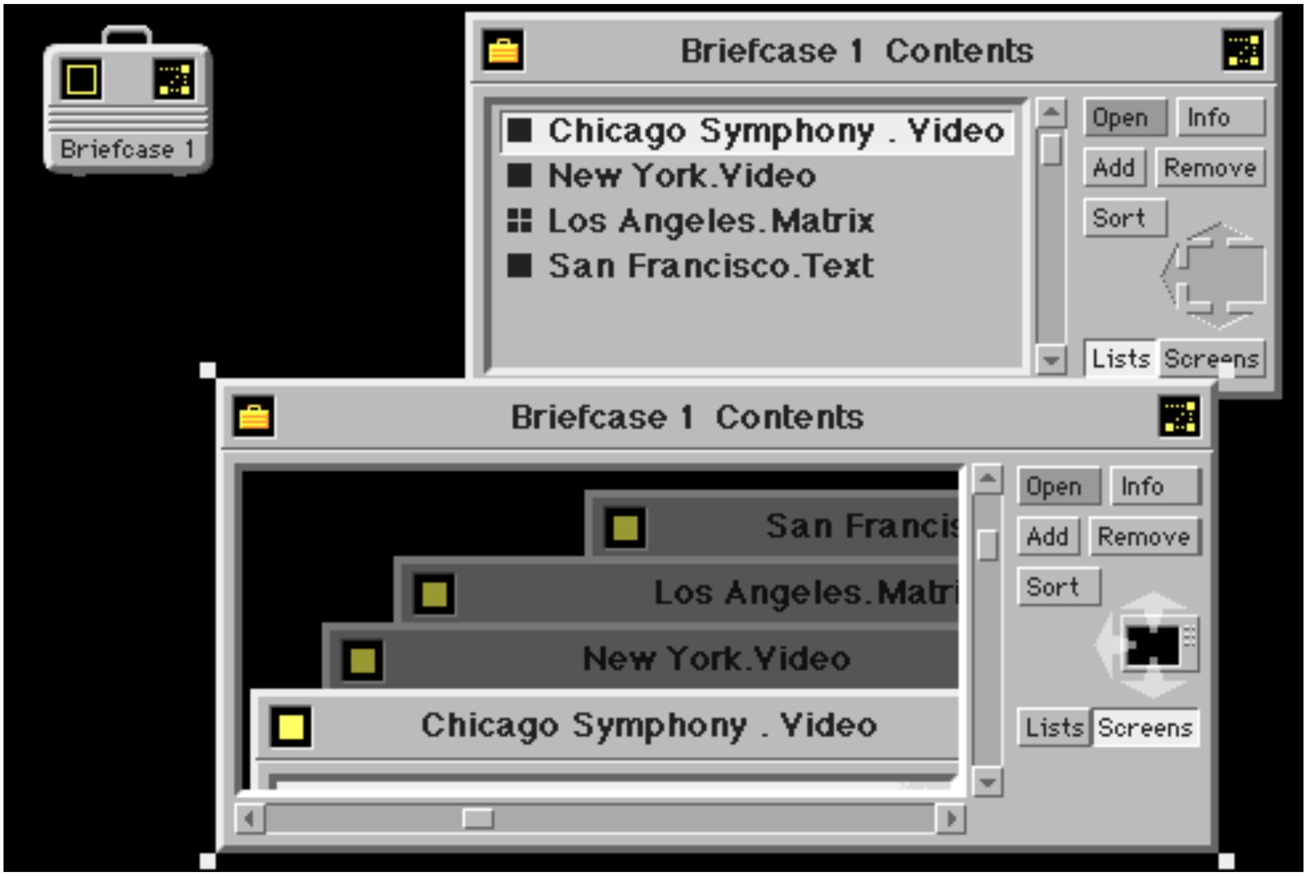


Figure 17: **Briefcases** Above is the *briefcase icon* and the two ways in which its *directory* can be viewed, in *lists mode* (top, behind) and in *screens mode* (bottom, front, *current focus* w/ *resize squares*). Note the similarity of *control panels* between the *briefcase directories* and other *directories* (see Figs. 6 and 7).

Figure 18: **Briefcase Source Glyph** When a *briefcase* is opened, the *source glyph* in the *header panel* contains a *briefcase symbol*. Like all *source glyphs*, it can be used in one of two ways. The user can use it to flash the *source briefcase icon* in the *current space* by selecting and holding on it or close and return to it to its *source briefcase icon* by activating it.

3.6 Briefcases

Briefcases are the means by which a user can collect copies of elements as he or she travels through *InfoSpace* and bring them along. This may allow the user to have constant access to needed elements that don't exist in his or her *current space* and save time traveling back and forth. It may also serve as a clipboard or scrapbook, analogous to those found in two-dimensional graphical user interfaces, yet have extended functionality.

Briefcases can be symbolized in several forms. The simplest is an icon that resembles a briefcase. The second is opened up to a *briefcase directory* and resembles and functions similarly to any other *directory*. The third is opened as the *current space*, in which case the contents (being viewed in the *screens mode*) are exchanged with the *current space* using the *swap-space glyph*. In this configuration the *briefcase* appears again, but this time looking as though it has been opened.

The *briefcase icon* contains a *source glyph* that indicates the place where the *briefcase* was first created, or a point or level specified by the user. The *briefcase* also contains a *hyper-connections glyph* enabling it to be *hyper-connected* to any other element.

The name of the *briefcase* will be determined and specified at creation, (and can be subsequently renamed), by the user. Name

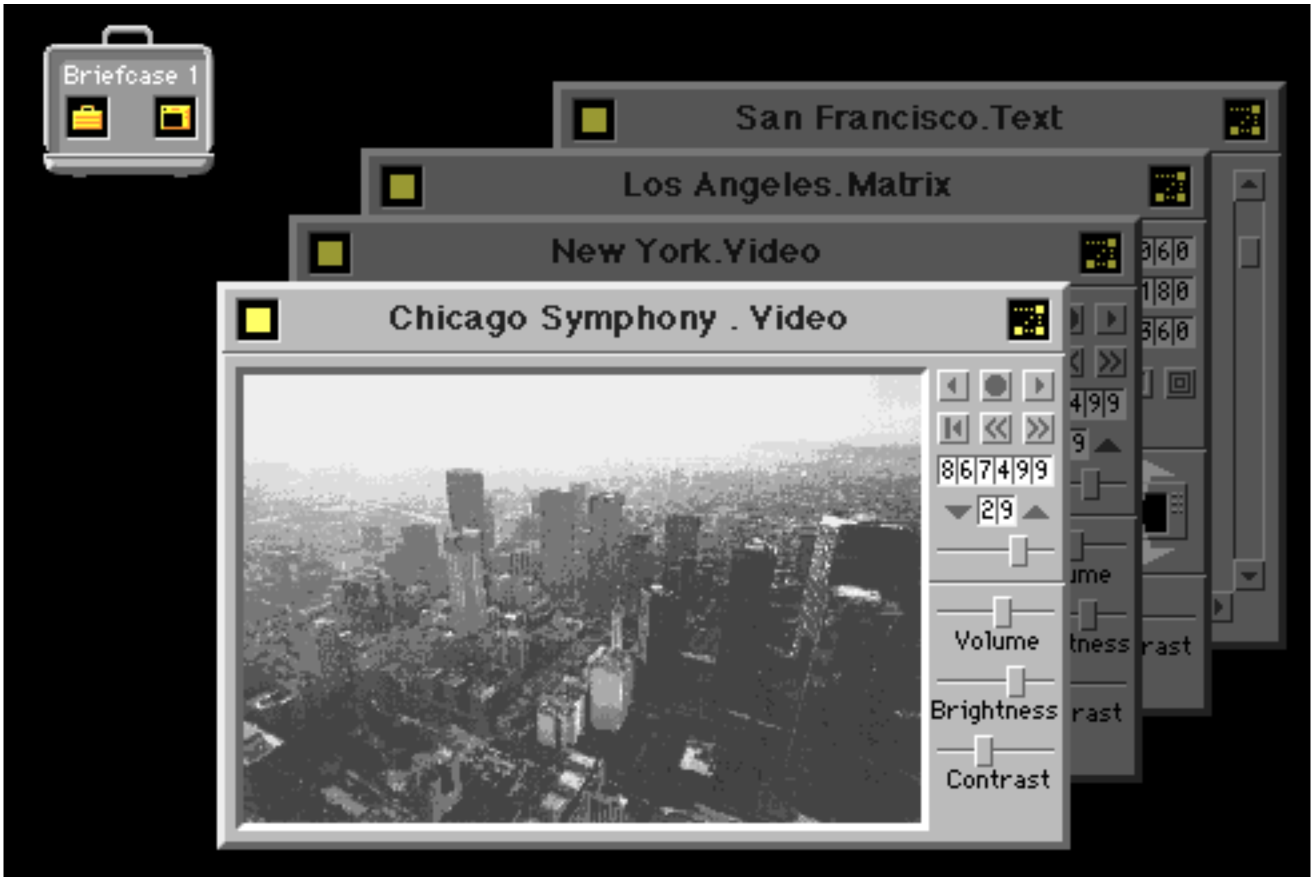


Figure 19: **Opened Briefcase** By *activating-to-3D* a *briefcase directory* via the *swap-space glyph*, the *space* of the *briefcase directory* will be exchanged with that of the *current space* (which will appear on a *screen* of its own). The *source briefcase icon* will come with the *briefcase directory* contents into the new *current space* and will be appear as an *opened-briefcase icon*. In addition to its name, which will appear highlighted in the *opened-briefcase icon*, there will be two *source glyphs*. The first one, containing a *briefcase symbol*, will allow the user to flash the *source briefcase icon* in the previous *current space* (now visible on a *screen* within the *current space* of the *briefcase* contents) by selecting and holding on it or close and return to it to its *source briefcase icon* by activating it. In this case the *current space* will automatically revert back to the previous *current space* and the *briefcase* will appear as the minimal or *shut-briefcase icon*. The second *source glyph*, containing a *screen/directory symbol*, will also allow the user flash the *source briefcase icon* in the previous *current space* (now visible on a screen within the current space of the briefcase contents) or close and return to its source briefcase directory by activating it. In this case the *current space* will also automatically revert back to the previous *current space* and the *briefcase* will appear as the *briefcase directory*, the viewing mode of which will be determined by the mode it was in when *activated-to-3D*. Not shown above is the *open-briefcase space control panel*, which is analogous to a *matrix control panel* (see Fig. 5).

altering will be accessible by selecting the name and entering a change of text. This is similar to the way names are created and edited in *blocks* or *matrices*.

When the *briefcase* is opened to a *briefcase directory*, the *header panel* will display a *source glyph* showing a *briefcase symbol*. By selecting and holding it selected, the corresponding *briefcase icon* will flash. This will aid the user in determining which *briefcase icon* he or she is examining the contents of when more than one briefcase exists. When *activated*, the *briefcase directory* will close down and return to its representative *briefcase icon*.

When the *briefcase* has been *activated to 3D* and appears as an *opened briefcase* along with its contents in the *current space*, another *glyph* is present in the *opened briefcase-icon*. In addition to the *briefcase-symbol source glyph*, there is also a *screen-symbol source glyph*. By *activating* the *briefcase-symbol source glyph* the user will return to the previous *current space* and the *briefcase* will revert to its minimal *briefcase icon*. If the user *activates* the *screen/directory-symbol source glyph*, he or she will return to the previous *current space* and the *briefcase* will revert to its *briefcase directory*. Features like these give the user more flexibility in interacting with and controlling the interface.

Briefcases function similarly to *matrices* with a few differences. *Briefcases* are very portable. That is, they automatically travel along with the user, wherever he or she goes within *InfoSpace*. *Matrices* can be transformed into *briefcases*, either as copies or altogether. *Briefcases* can likewise be transformed into *matrices* either as a copy or altogether. *Briefcases* can be hidden from view or restrained from traveling with the user, remaining in a particular level and place in *InfoSpace*. In this case, the *path*

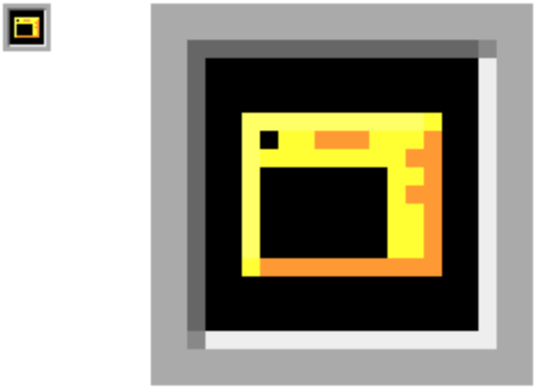


Figure 20: **Screen-Symbol Source Glyph** When a *briefcase directory* is *activated-to-3D*, the *opened-briefcase icon* (see Fig. 19) contains two *source glyphs*. The first is the *briefcase-symbol source glyph* that contains a *briefcase symbol* (see Fig. 18 as well as the *screen-symbol source glyph* shown above). Like all *source glyphs*, it can be used in one of two ways. The user can use it to flash the *source briefcase icon* in the previous *current space* (now visible on a *screen* within the *current space* of the *briefcase* contents) by selecting and holding on it or close and return to it to its *source briefcase directory* by *activating* it. By *activating* either of these *source glyphs*, the *current space* of the *briefcase* will close down and the previous *current space* will again become the *current space*. Having these two *source glyphs* allows the user flexibility in choosing how he or she would prefer the *briefcase* to be shown upon returning to the previous *current space*, either as a minimal *shut -briefcase icon* or a *briefcase directory*.

history and *system uptrace panel* will record these occurrences and the user will be able to quickly see where all *briefcases* exist and/or retrieve them.

Adding elements to the *briefcase* is very direct. When the *briefcase* is in the *briefcase-icon mode*, the user can add a *block* or other element to it simply by selecting and moving the desired element onto the *briefcase icon*. Similarly, when the *briefcase icon* is opened to a *briefcase directory*, the user may move elements onto a list or *screen* and that element will be copied into the *briefcase* automatically.

Once in the *briefcase*, elements may be copied and placed into a new level, removed and placed into a new level, or erased altogether. In order to access these functions, the user must be viewing the *briefcase* in the *briefcase directory mode*.

4. CONCLUSIONS AND SUBSEQUENT DESIGN DIRECTIONS

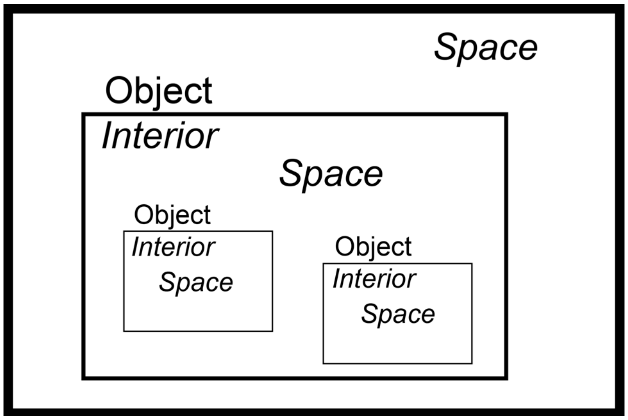
(Note - June 2020: Section 4 and associated figures 21-35 were written and developed between February 1993 and April 1993 and added to the original February 1991 InfoSpace white paper. The resulting extended InfoSpace white paper was presented at 3CyberConf, the Third International Conference On Cyberspace – held at the University of Texas at Austin in May 1993)

In 1988, when I began work on the behavioral descriptions and illustrations that led to *InfoSpace*, I approached it purely as a design exercise in which I could examine, visualize and document various methods of visual, physical and cognitive interaction. Working alone, I had no illusions that I could singularly design and model an entire interactional system, but felt its essential, underlying elements, relationships and patterns could be conceived and described. From this simple, underlying model I could then branch off into descriptions and illustrations of various elements and interactional aspects until I saw reason to stop, go back, and reiterate the underlying elements and interrelationships. One of my original intentions was that this work would be useful in the development of a physically and visually dynamic virtual environment in which users could communicate and interact with information in ways not possible via passive, static, sequential, or two-dimensional methods. Another intention was that this work could serve as a way for me to discuss such interactional ideas with others, hopefully leading to collaborative projects.

It was entirely from this assumption and using the experience I'd gained designing user interfaces for personal, business and medical equipment and software that I began to design this virtual, three-dimensional and interactional information space. The project was inspired to a great extent by converging patterns I perceived from my reading over the previous ten years in many diverse areas, ranging from psychology to systems theory. Two sources that contributed much to my early views on modeling information and its psycho/social results were Julian Jayne's *The Origins of Consciousness in the Breakdown of the Bicameral Mind* [1] which theorized on contributing factors in the evolution of mind and consciousness, and several of Buckminster Fuller's books [2][3][4]. Partly from the excited inspiration I felt, as well as being somewhat curious to find out how close my original intuitions were, most of *InfoSpace* was modeled before I came in contact with other virtual reality research. Between 1988 and 1991, when first-iteration work on *InfoSpace* was completed, my knowledge of and influence by work being done elsewhere on virtual reality had grown immensely. In late 1990, I first became aware of others in the VR field and their work, through the VR conference on The WELL [5], though, by that time most of the underlying concepts in *InfoSpace* had already been completed.

Subsequently, the model begun in *InfoSpace* is continuing and evolving in joint work with virtual architect, Clayton Graham, of PowerSpace. During the development of the *InfoSpace* model, it became apparent that there were many directions in which further ideation and developmental modeling could proceed. In the *InfoSpace* project, the interior spaces in which all interaction with object-represented information takes place were left completely undescribed. There were many branching aspects of spatial interrelationships, navigational motion, system boundaries, etc., that were left to be explored and modeled along with questions regarding how the *metainformation* necessary for each informational object to spatially/interactionally orient itself in relation to other objects, was to be created, gathered, or captured. Also, in *InfoSpace*, the only dynamic behavior of information illustrated was in a form of axis-oriented swarming, while dynamic shape and size transformations, and further interrelational behaviors were left for future exploration and modeling.

The current project, *InterSpace™*, is essentially the second iteration of *InfoSpace* and combines, evolves and expands on its interactional systems and behaviors, together with the dynamic



Figures 21-28: **Diagrams of Evolving Concepts from InterSpace™, the Second Design Iteration of InfoSpace™**

Figure 21: **Objects and Spaces** The relational organization within *InterSpace* is one of spaces and objects. The interior of any object is a space. Inversely, every space is represented externally as an object in a higher level space. In this way, it is very similar to *InfoSpace*.

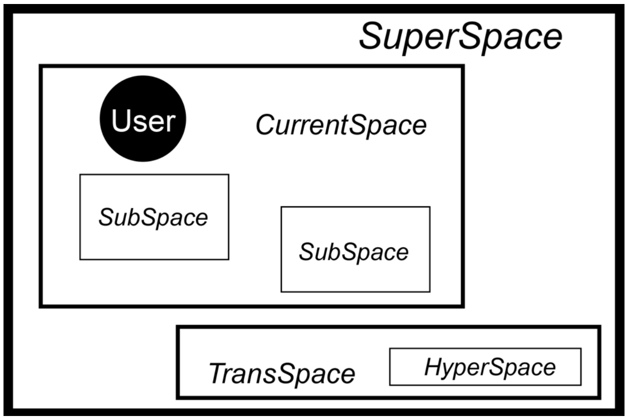


Figure 22: **User-relative Spatiality** Spatial relationships are relative to the user's location. The *space* a user is currently in is the *current space*. Consequently, all *spaces* above this level are *superspaces*, all objects represented in the *current space* and all objects nested below are *subspaces*. *Spaces* directly adjacent to the current space are *transspaces*. All other *spaces* are *hyperspaces*.

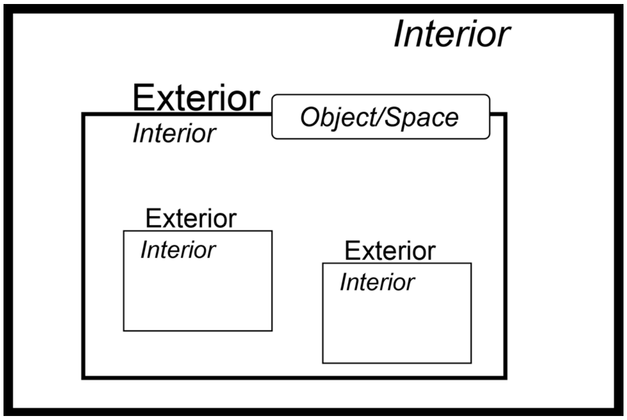


Figure 23: **The Exterior and Interior of Object/Spaces** The basic informational unit in *InterSpace* is the *object/space* and consists of two aspects, the first of which is the *primary information* represented by the *object/space* (a specific text, a video, a 3D model, an environment, etc.) The second aspect is *metainformation*, or information about the information (Objective/Universal: type, creator, origin, size, connections; and Subjective/Personal: preferred, assigned, portrayed. etc.) which provides hooks that are recognized and utilized by the user's interface in visually and behaviorally representing the *object/space's* objective exterior and spatial interior and the respective relationships to parent spaces and child objects.

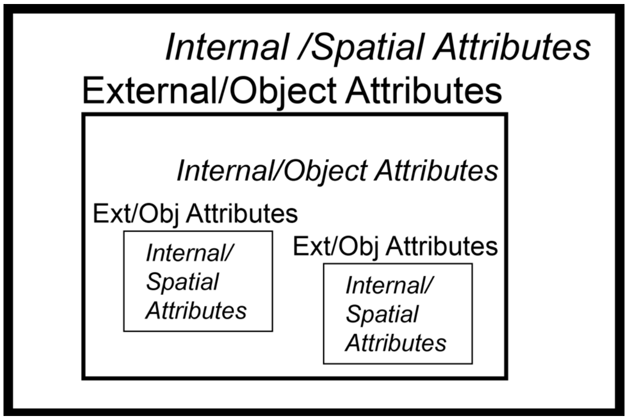


Figure 24: **Metainformational Hooks and Attribute Mapping** The *metainformation* contained by *object/spaces* provides *hooks* onto which various *attributes* can be selectively mapped. *Attributes* are visual representations and/or interactional behaviors of *object/spaces*, and are obtained as the result of the user's interface recognizing these *hooks* as encountered in a *current space* and then mapping *attributes* to the object/spaces accordingly. This process is called *attribute mapping*.

architectural/symbolic representations of spaces, objects, paths and user-perspectives developed by Graham [6].

As a result of going back and re-examining the underlying elements and interrelationships, we are developing an expanded classification of *object/spaces*, user-centered spatial and navigational concepts. Also being explored in depth are the ways in which *metainformation*, or information about information, can be created, captured and/or specified along with how it can be interacted-with/augmented-by a user-centered *interface database.* The *visual, physical*, and *auditory channels of interaction* and the division of the user-interface into aspects that are deployed through the concepts of *radial-projection* and *mapping* are two areas of interactional behavior that are being explored and

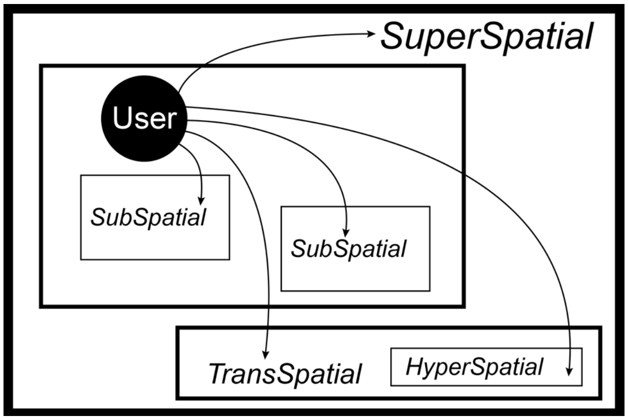


Figure 25: **InterSpatial Moves** Spatial moves, like spatial relationships, are relative to the user's location, or *current space*. Moves upward to higher levels of *space* are *superspatial moves*. Inversely, moves into the *spaces* of *current space child objects* are *subspatial moves*. Lateral moves to adjacent *spaces*, or *transspaces*, within the same parent space are *transpatial moves*. All other moves are considered *hyperspatial moves*.

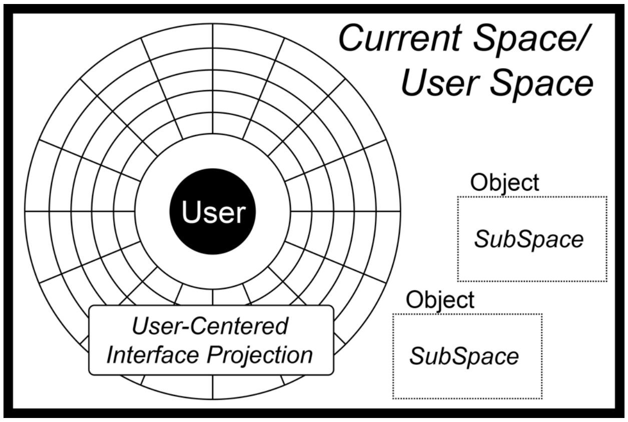


Figure 26: **User's Space and Interface Overlay** The user's *space* is centered on the user and is projected simultaneously into the *current space*. In this way, the interface always travels with the user from *space* to *space*, with individual elements of the interface deployed or hidden as desired. More about the user interface and how it's projected and mapped into the *current space* is explained in Fig. 28.

modeled in greater resolution. A simplified set of diagrams illustrating some of the evolving concepts from *InterSpace* are included here (see Figs. 21-28). New terms being developed for *InterSpace* are shown in italics in Figures 21-28, but are not included in the glossary at the end of this paper as they are still under development. These will be finalized and included in a full descriptional document along with dynamic, computer-generated virtual models (Screenshots shown in Figs. 29-35).

Since *InfoSpace* exists only as a descriptional model on paper, full experiential data on its functionality and/or appropriateness as an interface cannot yet be evaluated. My initial intuitions about the need for such a system, however, have been reinforced through the act of conceiving and modeling these concepts. However, parts of the interactional system being modeled here do have related concepts functioning in current software and devices, even if employed differently or in two-dimensional space only. Psychologically, it will be crucial to create for users a familiar interactional path or bridge, spanning from their familiarity with current two-dimensional systems to the newer dynamic, three-dimensional systems to come. Bringing commonly understood and utilized interactional displays, controls, and methods [7], such as windows, interactional dialog panels, informational displays [8][9], symbols [10], behaviors, etc. forward into new three-dimensional informational systems will likely result in faster user-familiarization. My experience working with and designing a wide variety of interfaces has shown me that there are many underlying similarities in interactional patterns, regardless of the specific functionality of the software or device.

One can only speculate at the individual and sociological changes that such a major informational and communicational shift will bring. The technology of written language was very likely developed out of necessity and now with the increasing complexities of information in our present society overwhelming our present methods, it seems almost imperative to attempt the deliberate design of better and/or radically different methods of interacting with more information. The spoken word was not abandoned after written language was developed, and in similar fashion, newer, higher forms of communication and information organization are likely to swallow whole and intact preceding forms. Emerging sciences, such as *chaos* [11] and *complexity* [12][13] employ models almost inaccessible cognitively without computer visualization. As these fields of research and thinking lead to new ways of visualizing patterns of information in many forms, they will almost certainly require robust interactive systems in which this information experience can take place. The visual channel is particularly interesting for its link via the visual cortex to the pre-conscious brain [14]. This raises the possibility that an interactional information system might be able to utilize the visual cortex in processing informational patterns and relationships visually at speeds not possible with other informational mediums relying primarily on cognitive input and processing.

It's difficult to imagine the world before the written word. Yet the limitations of cave walls, tablets, paper, or even computer screens as they're still primarily used today, have forced us to reduce the dynamic, multi-dimensional, interrelational systems that are our world, down into static, linear, sequential, non-interactive representations. Our view of the unnerving complexity inherent in many of today's informational challenges such as economics, geopolitics, the environment, etc., must surely be very similar to the way things such as simple record-keeping looked when all we had were spoken words and our memory. Like our ancestors before us, if we are to meet today's challenges successfully, we must evolve the ways in which information is perceived and communicated. For the first time in history, we have the technological means to create and interact with information in a dynamic, systemic and interrelational manner. If the historical changes that followed the written word are any indication of what happens when information technology is advanced, we are poised for a big leap forward.

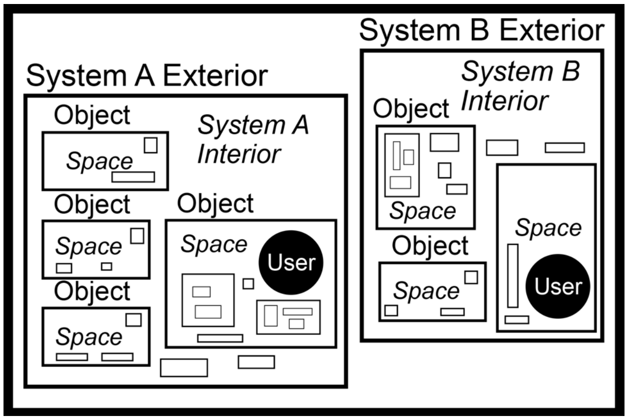


Figure 27: Systems and Real-time Visualization All of the nested information in any separate database is a *system*. For example, each user will have a personal *system* which may consist of all the information stored on his media, or, the user may enter *InterSpace* from within a *multi-user system* (public kiosk or multi-user terminal). These *multi-user systems* (virtual communities or large databases for example) are likely to be much larger than *personal systems*.

Information in *systems* will not be stored in full visual form, but will be visualized on the fly in current spaces according to how the user's *interface database* is set up to portray it upon encountering it. For example, an object in a *current space* may represent a video. The user's interface, upon recognizing an object of this type in the *current space*, will utilize its *interface database* to portray the video object in the way specified to portray all videos when they're encountered (i.e.: it gets a view screen and video controls, etc. *mapped* to it). This process will occur through the use of *hooks*, a type of *metainformation* that each object knows or is told about itself that the interface can then recognize and *map* appropriate visual or interactional elements and behaviors to. This will allow users to personalize the ways in which they visualize information.

A user may perceptually "travel" to other systems by connecting to them and bringing back in real time only the *primary information* and its *metainformation* from a *current space* that is then visualized by utilizing the user's *interface database*. As the user moves about in this other system from space to space, new *primary information* and *metainformation* will be brought back and visualized. In this way, the user will have the perception of extending his personal visualizational settings, while other users may visually experience the same spaces and objects differently according to the attribute setup of their interface database.

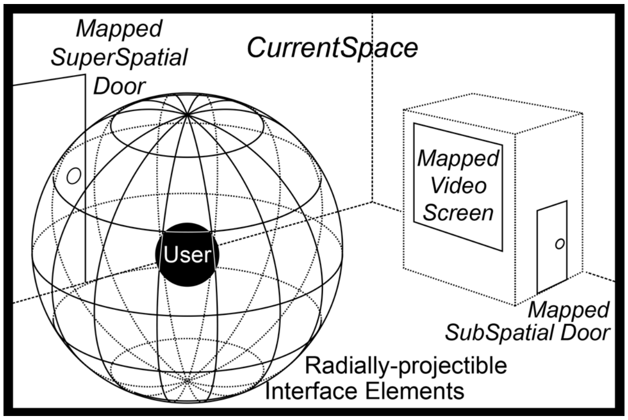


Figure 28: Interactional Channels, Projection, and Mapping Utilizing the *interface database* and its settings, the user will interact with object/spaces, both from interior and exterior viewpoints via three types of *interactional channels*: the visual, the auditory, and the physical. These *interactional channels* will overlap perceptually through parallel signaling (i.e.: an object that is simultaneously visually "long" and auditorily high-pitched, etc.)

The *visual interactional channel* will consist of two forms of interface visualization, the *projected*, and the *mapped*.

The first form of interface visualization is *projection*. The *projected interface* consists of a virtual, user-centered, radially-projectible array of interactional screens or objects. This will allow for cognitive interaction through elements such as navigational maps, lists, and informational, feedback, or view screens.

Some of these *projected interface elements* may be paired-to/controlled-by real parallel *physical interactional controls* through *parallel signaling*.

The second form of interface visualization is *mapping*. As opposed to radially-projected screens and objects, *mapping* will allow the user to experience direct, object/space-centered interactivity by *mapping* interactional objects or *attributes* directly into the *current space* and onto the *current objects* by recognizing and utilizing *metainfomational hooks*. *Smart doors*, either *superspatial* or *subspatial*, that automatically attach themselves in appropriate/specified, recognizable places within *spaces* or on objects would fall into this category. Video view/control *screens* that automatically map themselves onto objects of class-type video would also fall into this category.

The *physical interactional channel* will consist of two main categories, the *indirect physical interface* (i.e.: keyboards, switches, sliders, etc.), and *direct physical interface*, comprised of virtual physical controls and feedback devices (i.e.: gloves, bodysuits, etc.). As stated inversely above, these physical interactional elements may be paired with parallel projected interface elements through *parallel signaling*.

The *auditory interactional channel*, except in cases of voice-recognition, will function primarily to transmit augmentational information regarding status and/or location of objects, or, quite possibly, the user. This sound will be three-dimensionally placed when possible, to reinforce the perception of spatial relationships and meaning.

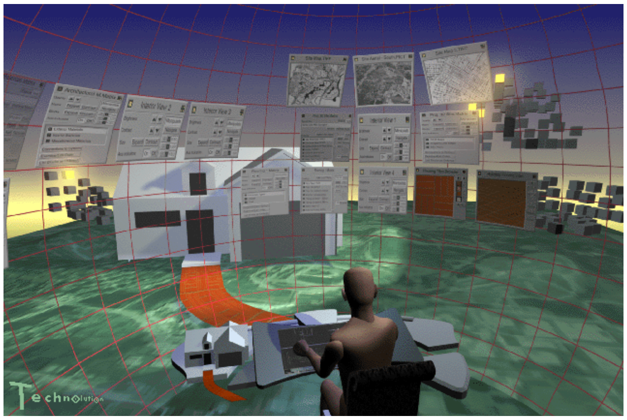


Figure 29: Architect's InfoSpace Environment
An architect's environment utilizing *InfoSpace* to browse, visualize and organize project information and other creative and information-intensive tasks. *Informational matrices* are visible along with *interactive directories*, *panels* and *screens*.

This illustration was created by Max Sims, Technolution

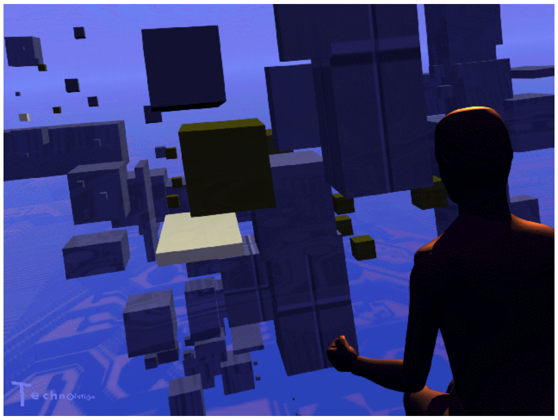


Figure 30: Interaction with an Informational Matrix
Matrices of objects representing information and *subsystems* of information float before a user in the *InfoSpace* environment. *Indexed object metainformation* allows for dynamic changes in size, configuration and appearance in response to user-issued queries about the information in this *matrix*.

This illustration was created by Max Sims, Technolution

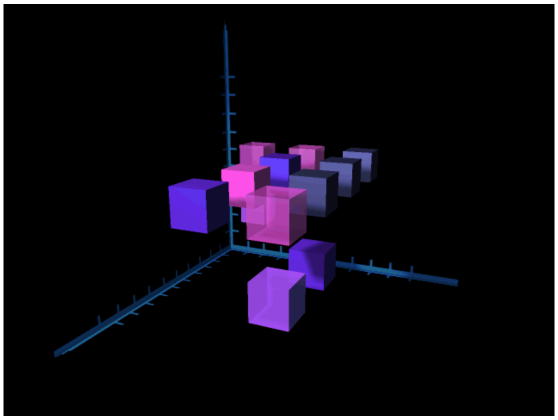


Figure 31: Matrix/Construct Morphing Sequence 1
This is the first in a series of five illustrations showing the *visual/spatial transformation* of an *InfoSpace matrix* resulting from a user-issued query about the information in the *matrix*. Length, size, position, etc., are all ways in which comparative or interrelational information about this set can be shown. Each object may represent a file or set of files, or nested system of files.

This illustration was created by Clayton Graham, PowerSpace

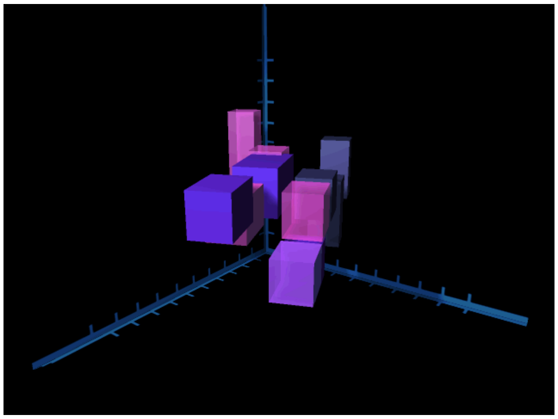


Figure 32: Matrix/Construct Morphing Sequence 2
This is the second in a series of five illustrations showing the *visual/spatial transformation* of an *InfoSpace matrix* resulting from a user-issued query about the information in the *matrix*. Here the *matrix* is, in an animated manner, *transforming* from a cluster of objects into a *construct* of intersecting objects, each with unique spatial positions and visual configurations to reveal specified interrelationships.

This illustration was created by Clayton Graham, PowerSpace

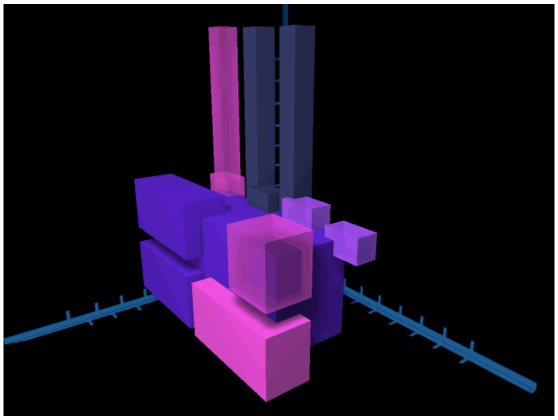


Figure 33: Matrix/Construct Morphing Sequence 3
This is the third in a series of five illustrations showing the *visual/spatial transformation* of an *InfoSpace matrix* resulting from a user-issued query about the information in the *matrix*. Here the *matrix* construct is complete. The objects have all come to rest, with spatial positions and visual configurations representing the "answer" to the *matrix query*.

This illustration was created by Clayton Graham, PowerSpace

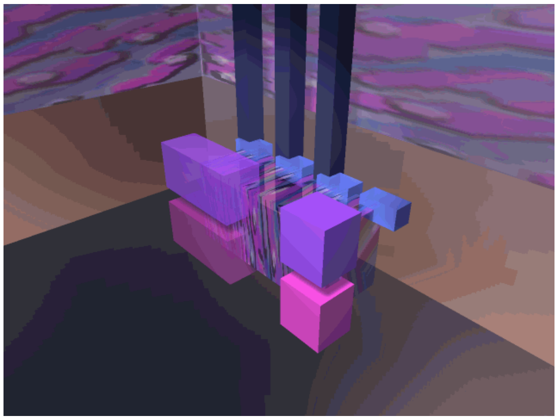


Figure 34: Matrix/Construct Query Result Visualization 1
This is the fourth in a series of five illustrations showing the *visual/spatial transformation* of an *InfoSpace matrix* resulting from a user-issued query about the information in the *matrix*. In this view of the *matrix construct*, the *primary object* to which all other objects are relative is shown with a patterned surface. This pattern is generated algorithmically from the object's internal data, and allows for abstract recognition and differentiation.

This illustration was created by Clayton Graham, PowerSpace

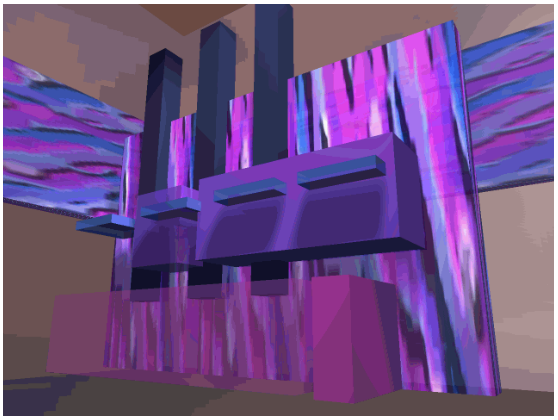


Figure 35: Matrix/Construct Query Result Visualization 2
This is the fifth in a series of five illustrations showing the *visual/spatial transformation* of an *InfoSpace matrix* resulting from a user-issued query about the information in the *matrix*. Here, the user has changed position and the *matrix construct* has gone through yet another *spatial transformation* in response to another query.

This illustration was created by Clayton Graham, PowerSpace

5. GLOSSARY OF INFOSPACE TERMS

(Note – June 2020: Section 5 – Glossary of InfoSpace Terms was compiled between November 1988 and February 1991 and pertains only to the first version of the InfoSpace white paper and subsequently printed and distributed to various individuals and research groups in the Silicon Valley beginning in November 1989.

These groups and individuals included Joy Mountford (November 1989), Director of Apple Computer's Advanced Technology Group (ATG); Bill Moggeridge and Bill Verplank (November 1989) of IDTwo (later IDEO); Stuart Card, (1990), Xerox PARC; Bill Joy (October 1990), Co-founder and Chief Scientist of Sun Microsystems/SunPro which I was consulting to at the time under Vice President, Eric Schmidt.

Distribution of the InfoSpace white paper was my attempt to find and convince some Silicon Valley team to let me join with them to build an internet browser.

InfoSpace, even in its first iteration version, combined the concepts of an Internet Browser (combined with the OS UI), metadata-powered information visualization built into the base OS UI experience (my own term, working in a complete vacuum, was "metainformation"), and 3D VR, prior to my exposure to research and work being done in Virtual Reality.

Due primarily to my somewhat appearing out of nowhere (I had just moved at the end of 1989 to the Silicon Valley from Dallas, Texas where I was working as a young pioneering Interaction Architect and Designer on the InfoSpace concept model alone concurrent with my experience working with Norm Cox on the development of Sun Microsystem's OS GUI, "Open Look,") and lacking a Ph.D. or other credentials contextualizing my conception of this paper model, my work and ideas were not recognized as significant or worthy of collaboration on what could have been an internet browser pre-dating Mosaic, using a reduced and collaboratively refined set of these features.)

2D(3D) - The two-dimensional representation of three-dimensional information that appears on *screens*.

3D Control Panel - A *control panel* used to manipulate or navigate a three-dimensional object, model, or environment. This can apply to *InfoSpace* elements such as *matrices* as well as application-level 3D objects.

Activate - The action of selecting and opening all in one physical or voice signal from the *physical interface* or by using the *cursor operable controls*.

Activate to 3D - To convert the 2D(3D) flat representation on a screen into the current three-dimensional *space*.

Activation - To cause, through an interactive signal, an event or action to take place. Usually refers to the *activation* of a *glyph* or *icon*, which has already been targeted and selected by the user.

Address - The location of a *block* or *superblock* within a *matrix* resulting from the values or orders assigned to the three *parameters* which correspond to the three spatial *axes* X, Y, and Z.

Axis - In *space*, one of the three possible directions at 90 degree angles to one another

Axis Indicators - The three X, Y, and Z orientation lines that allow the user to visually keep track of the spatial orientation of a *matrix*.

Block - The principal organizational unit of information or data within the *InfoSpace* concept, the *block* represents a singular file. It is represented by a suspended cube-shaped object. A *block* is rendered as a simple cube with smooth sides.

Block Directory or **SuperBlock Directory** - A floating menu that contains a name for the *block* or *superblock* it represents and lists the information or *submatrices* that might be contained within. This is the type of directory that appears when the user is utilizing the *InfoSpace* model via the *explore mode*.

Briefcase - A container that can be brought along with the user as he or she travels through *InfoSpace* and into which can be placed any kind of element or copy of an element (i.e. *blocks*, *superblocks*, etc.) The *briefcase* can be closed down to a *briefcase-symbol icon* or opened up in the form of a *briefcase directory*. The user can create multiple *briefcases*, make them invisible, or restrain them to a particular place or level.

Briefcase Symbol - A graphic symbol resembling a *briefcase* with a handle and used in *source glyphs* and *directories*.

Clean-up and **Arrangement Controls** - Routines that will aid the user in arranging *matrices*, *screens*, *directories* and other elements within a *space*. These may include stacking, moving, dimming, and other such commands that will help the user avoid visual and spatial clutter. In addition to standard clean-up and arrangement controls, the user will be able to specify and create custom routines.

Components - The different parts of the *InfoSpace* interface, i.e. *space*, *blocks* and *superblocks*, *systems*, *screens*, *directories*, *manipulators*, *navigators*, etc..

Control Panel - The area on a *screen* that allows the user to *manipulate* or operate the display of information. For example, if the *screen* was a video application, the *control panel* might include the means to start and stop the video, control the speed at which it is played, or any of a host of other possible controls.

Current Focus or **Focus** - The element that the user is currently working with and is active within the *current space*. *Screens* and *directories* generally can be resized while the *current focus*.

Current Space - The infinite, three-dimensional *space* that surrounds the user and all opened elements. Also known as the *Primary Space*. (See also *Screen's Space*).

Cursor - The floating cross-hairs that the user moves about within the *space* to point out, select, and subsequently open objects.

Cursor Operable Controls - *Interface controls* that the user *activates* with the cursor from within the *space*. These may correspond with equal *physical interface controls* that the user has access to. *Cursor operable controls* may appear on various elements within the *space* such as *directories*, *screens*, etc..

Directory Item - A listing containing names, information about type, and any notational *symbols* or *icons* that may be contained within a *directory* of any type.
Directory Listing Glyph - The small *symbol/glyph* displayed to the left of a *directory item* that denotes whether that listing is for a *block*, *superblock*, *matrix*, *system*, or other *InfoSpace* element.

Element - Any *glyph*, *icon*, *block*, *superblock*, *matrix*, *system*, *directory*, *screen* or other component or level of *InfoSpace* that can be identified for purposes of manipulating, traveling to, *hyper-connecting* to, performing an action upon, referencing, or listing, etc..

Explore Mode - The term used to describe a user accessing *block directories* or *superblock directories* by pointing the cursor at a particular *block* or *superblock* and selecting it. This is called *exploring* since it involves the user essentially poking around or *exploring* a *matrix* or set of *blocks*.

Find Mode - The term used to describe a user accessing *block*, *matrix*, or *system directories* by *summoning* a particular *directory* via the *system* tools. This method is employed when the user is certain of what he or she wants to access and is trying to locate and access it within a *system*, or *matrix*.

Find Mode Directory - An enlarged *directory* that contains a field where the user can enter the name of an element to find and/or retrieve.

Glyphs (see also ***Symbols***, ***Icons***) - Small targetable graphics containing a *symbol* that can be selected and/or *activated*. For definition purposes, the order is 1 - *Symbols* (purely graphic, non-interactive representations), 2 - *Glyphs* (*Symbols* that when hit or activated perform some action or elicit some event), and 3 - *Icons* (independent, miniaturized, floating representations of *InfoSpace* elements that when selected and activated, open up into their full form and size).

Header Bar or **Header Panel** - The area at the top of a *screen*, *information panel*, *control panel*, or *hyper-connection panel* that contains the *source glyph*, the name of the element, and (if appropriate) a *hyper-connection glyph*.

Highlight - The visual change in an element of the interface, usually brightening, changing in color, or both, that indicated that it has been selected.

Hit or **Hitting** - To give the *physical interface* a signal after targeting a particular element, *glyph*, or *control symbol* (i.e. button, slider, elevator, etc.).

Hyper-Connection - A two-way, trans-spatial relational connection between any two or more elements within *InfoSpace*. Every *hyper-connection* is divided into two corresponding parts (*in-from* and *out-to*) and is recorded in the *hyper-connection panels* of connected elements as such. Every time an *out-to hyper-connection* is established from element "A" to element "B", it is registered as an *out-to hyperconnection* in element "A"'s *hyper-connection panel* and as an *in-from hyper-connection* in element "B"'s *hyper-connection panel*.

Hyper-Connection Panel - The panel displaying and allowing for control of *hyper-connections* between elements. Divided into three parts for *in-from hyper-connections*, *out-to hyper-connections*, and a *control panel* section, a *hyper-connection panel* will be accessible for any *hyper-connectable* element.

Icons (see also ***Symbols***, ***Glyphs***) - An element that has been reduced to a floating selectable/activatable *symbol* such as a *briefcase*. A *block* or *superblock* is also a form of icon. For definition purposes, the order is 1 - *Symbols* (purely graphic, non-interactive representations), 2 - *Glyphs* (*Symbols* that when *hit* or *activated* perform some action or elicit some event), and 3 - *Icons* (independent, miniaturized, floating representations of *InfoSpace* elements that when selected and *activated*, open up into their full form and size).

Identifier Glyphs - Small, two-dimensional graphic *symbols* that indicate the type identity of a listed item. These *symbols* also appear as *source glyphs* and can represent *blocks*, *superblocks*, *matrices*, *systems*, *briefcases*, *screens/directories*, etc..

In-From Hyper-Connections - One of the two corresponding parts of a *hyper-connection*, listed in the *in-from* section of an element's *hyper-connection panel* and representing a *hyper-connection* coming in from another element.

InfoSpace - The name given to this model of a three-dimensional user-interface and interrelational database system.

Light source - The unseen point in the *space* that represents an imaginary source of light determining how the illumination and shading of the *blocks* and other elements in the space will be rendered.

Lists Mode - One of the two visualizing modes of *directories* (see also *Screens Mode*), where the contents of the directory window are displayed as *sortable* listing of contained elements.

Manipulation - The term used to describe the interaction mode in which the user is stationary and tumbles and otherwise moves the *matrix* or 3D object or model around in front of him or her.

Manipulators - The *physical controls* and corresponding visual representations of these controls that the user operates in order to *manipulate* the *matrix* or 3D object, or model.

Matrix Control Panel - A *control panel* that is associated with a particular *matrix* and with which the user can access information, alter the visual appearance, *manipulate*, *navigate*, or reconfigure the *matrix*.

Matrix Level - An assigned number that indicates the nesting position of a particular *matrix* within a *system*.

Matrix or **Matrices** - The organizing three-dimensional grid-like structures that *blocks* fit into within a *space*. *Matrices* are formed by arranging *blocks* and *superblocks* within them according to the three *matrix parameters*.

Matrix Parameters - Sets of three orders or values assigned to a *block* or *superblock* that determine the address of a *block* or *superblock* within a *matrix*. Each *matrix parameter* corresponds to one of the three *matrix* axis' X, Y, or Z and together define a point in three-dimensional space. The *matrix parameters* can be shuffled, changed, and reordered by the user to alter the configuration of the *matrix* and therefore create a different visual relationship for the user. This is one of the primary features that makes *InfoSpace* a powerful means of exploring and understanding large and complex data structures.

Navigation - The term used to describe the interaction mode in which the *matrix* is held stationary in the *space* and the user, by means of a joystick and/or other *physical interface controls*, essentially flies or *navigates* around or through the *matrix*. *Screens* and all peripheral tools and *navigators* remain fixed in relationship with the user while in *navigation mode*.

Navigators - The *physical controls* and corresponding visual representations of these controls that the user operates in order to *navigate* in and around the *matrix* and/or 3D objects, models, or environments.

Open - The action the user takes to turn a selected *system*, *matrix*, *superblock* or *block* into a floating *screen* that contains its contents (either information if it is a *block* or a 2D(3D) representation of a *submatrix* if it is a *superblock*). This is accomplished through a physical or voice signal the user gives to the *physical interface controls*. The open function can take place with a cursor-selected *block* or *superblock* in a *matrix* or from a selected name of a *system*, *matrix*, *block* or *superblock* from within a *directory*.

Open-Briefcase Icon - A graphic *symbol* resembling a *briefcase* that is lying opened and displayed when a *briefcase* has been *activated-to-3D* and its contents and *space* are *current*.

Order - A means of establishing a sequence for a *matrix parameter*. An example would be January, February, March, etc..

Out-to Hyper-Connections - One of the two corresponding parts of a *hyper-connection*, listed in the *out-to* section of an element's *hyper-connection panel* and representing a *hyper-connection* going out to another element.

Physical Interface - The *physical controls*, buttons, knobs, joysticks or other interaction devices the user operates to communicate with and control the *InfoSpace system*.

Previous Current Space - The *space* that was the *current space* directly before the *space* that is currently the *current space*. For example, if the user is in *matrix* "A", examining 3D *Screen* "B" and then *activates* the *swap-space glyph* on "B"'s *control panel*, the *current space* of "A" will be transferred to a *screen* and become the *previous current space*, while "B's *space* will become the *current space*.

Primary Space (see *Current Space*)

Resize Squares - Small, targetable white squares that appear at all four corners of any *screen*, *control panel*, or *directory* that is the *current focus*. They allow the user to stretch or compress the size of the element in two-dimensions.

Review Backward History - The process of playing back in reverse the recorded travel of the user through *InfoSpace*. This is accessed via a mode available on the *system uptrace panel*.

Review Forward History - The process of playing forward from a point in the recorded travel of the user through *InfoSpace*. This is accessed via a mode available on the *system uptrace panel*.

Screen - Three-dimensional but flat-screen monitors that float in the *space* in front of the user. The purpose of the *screen* is to display two-dimensional information (text, two-dimensional graphics, video, etc.) or the two-dimensional representation of three-dimensional information [2D(3D)]. A *screen* is resizable and repositionable in the *space* and this is controlled by the user.

Screen/Directory Symbol - A graphic *symbol* resembling a miniature *screen* or *directory* and used in one of the *source glyphs* in the *opened-briefcase* to allow the user to close the *current briefcase* space down to a *briefcase directory* rather than down to a *shut-briefcase icon*.

Screens Mode - One of the two visualizing modes of *directories* (see also *Lists Mode*), where the contents of the *directory* window are displayed as an arrangeable collection of *screens* or a manipulatable *matrix* representing contained elements.

Screen's Space - An infinite, three-dimensional *space* that is partially viewable to the user via a window in a *screen* or *directory*, but not virtually accessible until it is *activated-to-3D*, thus becoming the *current space*. Also known as *Secondary Space*. (See also *Current Space*).

Secondary Space (see ***Screen's Space***)

Select - To *hit* after targeting with the cursor, pointing device, or visual indicator. Can sometimes mean select and hold. *Selecting* involves placing the cursor over a *block*, *superblock* (in the *explore mode*) or over a *directory* item (in the *find mode*) and giving a physical or voice signal to the *physical interface controls*. If the user is employing the *explore mode*, a *block* or *superblock directory* appears in front of the user giving the name of the selected *block* or *superblock* and listing the information or *submatrices* contained within. The selected *block* or *superblock* is also highlighted to indicate that it is the *block* that the *directory* is referring to. If the user is in the *find mode*, selection involves placing the cursor over the *directory* Item he or she may wish to locate or subsequently open. The user gives a physical or voice signal to the *physical interface controls* and the *directory* item is highlighted or otherwise denoted and the corresponding *system*, *matrix*, *superblock* or *block* visible within the *space* is highlighted.

Shut-Briefcase Icon - A floating *icon* resembling a *briefcase* that represents a user-specified container of elements. It can be *activated* to open to a *briefcase directory*, listing or displaying its contents.

Sorting - Any method by which the user may rearrange a listing within a *directory*, or group of elements. This may be by type, date, size, etc..

Source Glyph - The *glyph* located in the upper right corner of a *screen*, *control panel,* or *directory* in its *header panel* that indicates the type of element from which this *current focus* originated or opened from. The *source glyph* can function in two ways, either by selecting and holding it, in which it will cause its source element to flash, or by *activation*, whereupon it will close down the element and return it to the source element.

Space - The three-dimensional *space* that the user is within and oriented to at any given time.

Stereoscopic - The effect of perceived three-dimensionality created through sending two views (from slightly different viewpoints) to the user's right and left eyes. This can be achieved through the use of a device known as eyephones, which incorporate two small video monitors, one for each eye.

Storyboard Configuration - A two-dimensional layout of the elements within a *matrix*. Can be relationally ordered and addressed with three-axis *parameters*. Can also serve as a non-ordered spatial collection of *blocks* and *superblocks* and other *matrix* elements.

SubBlock - A relative term used to denote a *block* that is a part of a *superblock* being examined in the *current matrix level*.

SubMatrix - A relative term used to denote a *matrix* that is a part of a *superblock* being examined in the *current matrix level*. In other words, the *matrix* of one lower level.

Summon - The act of invoking or calling up a *directory*, *system* tools or any other *InfoSpace* element from its previous hidden state.

SuperBlock - A relative term used to denote a *block* that represents the entire *matrix* being examined in the *current matrix* level.

SuperMatrix - A relative term used to denote a *matrix* that contains the *block* that represents the entire *matrix* being examined in the *current matrix level*. In other words, the *matrix* of one higher level.

Swap-Space - The act of exchanging the *current space* with the *screen* or *secondary space* contained within a *screen* or *directory*. Accessed via the *swap-space glyph*, the *space* of the *current focus* will come out and become the users primary or *current space* and the *previous current space* will appear on a *screen*. This can function recursively and will allow the user easy access to multiple spaces.

Symbols (see also Glyphs, Icons) - Small graphics representing an object, action, or idea. For definition purposes, the order is 1 - *Symbols* (purely graphic, non-interactive representations), 2 - *Glyphs* (*Symbols* that when *hit* or *activated* perform some action or elicit some event), and 3 - *Icons* (independent, miniaturized, floating representations of *InfoSpace* elements that when selected and *activated*, open up into their full form and size).

System - The term given to represent the entity of nested *supermatrices*, *matrices*, and *submatrices* that a user designates as a single related group.

System Directory - A *directory* of all the *systems* that are contained within a computer or network running *InfoSpace*. This type of *directory* can be *summoned* through the *find mode*.

System Level - An assigned number that indicates the nesting position of a particular *system* within a nested group of *systems*.

System Matrix Directory - A directory of all the *matrices* that are contained within a *system*. This type of directory can be *summoned* through the *find mode*.

System Tools - Utilities, some with visual representations, that are either displayed or hidden (to be *summoned*), the user utilizes in order to accomplish some goal or action.

System UpTrace Panel - A status and *informational panel* the allows the user to keep track of where he or she is, where he or she has been with a listing of open *systems*, *levels* and the *current focus*. The user also has the option of bringing up a *path history* that will allow him or her to see and automatically travel back and forth, if desired, along past paths through *InfoSpace*.

User - A person who is operating within *InfoSpace*.

Value - A numerical means of establishing a sequence for a *matrix parameter*. An example would be sales values of $1,000.00, $2,000, $3,000, etc..

READING

M. Benedikt, ed., *Cyberspace: first steps* (Cambridge, MA, MIT Press, 1991)

S. Brand, *The Media Lab* (New York, NY, Viking Penguin, 1987)

F. Capra, *The Turning Point* (New York, NY, Simon and Schuster 1982)

J. M. Carroll, ed., *Designing Interaction* (Cambridge, UK, Cambridge University Press, 1991)

P. Falzon, ed., *Cognitive Ergonomics* (San Diego, CA, Academic Press, 1990)

H. Gardner, *The Mind's New Science* (New York, NY, Basic Books, 1985)

D. Gelernter, *Mirror Worlds* (Oxford, UK, Oxford University Press, 1991)

H. Haken, *The Science of Structure: Synergetics* (New York, NY, Van Nostrand Reinhold, 1981)

S. S. Hall, *Mapping the Next Millennium: the discovery of new geographies* (New York, NY, Random House, 1992)

M. Laver, *Information Technology: agent of change* (Cambridge, UK, Cambridge University Press, 1989)

D. Marr, *Vision* (New York, NY, W. H. Freeman, 1982)

K. Pimentel and K. Teixeira, *Virtual Reality* (New York, NY, Intel/Windcrest/McGraw Hill, 1993)

R. Ornstein, *The Evolution of Consciousness* (New York, NY, Prentice Hall Press, 1991)

H. Rheingold, *Virtual Reality* (New York, NY, Summit Books, 1991)

P. Russell, *The Global Brain: speculations on the leap to planetary consciousness* (Los Angeles, CA, J.P. Tarcher, 1983)

H. Samet, *The Design and Analysis of Spatial Data Structures* (Reading, MA, Addison-Wesley, 1989)

N. Stephenson, *Snow Crash* (New York, NY, Bantam Spectra, 1992)

B. Wooley, *Virtual Worlds* (Oxford, UK, Blackwell Publishers, 1992)

REFERENCES

1. J. Jaynes, *The Origins of Consciousness in the Breakdown of the Bicameral Mind* (Boston, MA, Houghton Mifflin 1976)

2. B. Fuller and R. Marks, *The Dymaxion World of Buckminster Fuller* (New York, NY, Doubleday 1960)

3. B. Fuller, *Ideas and Integrities* (New York, NY, Collier 1963)

4. B. Fuller, *Operating Manual for Spaceship Earth* (New York, NY, Simon and Schuster 1969)

5. The WELL (The Whole Earth 'Lectronic Link,) a major internet bulletin board located in Sausalito, CA (via modem (415) 332-7398).

6. C. Graham, *Rhythm of Time; The Visitor; Industrial Church; Oasis; CyberSchema* (videos of Dynamic Computer-generated Architecture and Environments, 1992)

7. B. Laurel, ed. and S. J. Mountford, *The Art of Human-Computer Interface Design* (Reading, MA, Addison-Wesley, 19)

8. E. R. Tufte, *The Visual Display of Quantitative Information* (Cheshire, CN, Graphics Press 1983)

9. E. R. Tufte, *Envisioning Information* (Cheshire, CN, Graphics Press 1990)

10. H. Dreyfuss, *Symbol Sourcebook* (New York, NY, Van Nostrand Reinhold 1984)

11. J. Gleick, *Chaos: making a new science* (New York, NY Viking 1987)

12. M. M. Waldrop, *Complexity: the emerging science at the edge of order and chaos* (New York, NY, Simon and Schuster 1992)

13. R. Lewin, *Complexity: life at the edge of chaos* (New York, NY, MacMillan 1992)

14. R. M. Friedhoff and W. Benzon, eds., *Visualization: the second computer revolution* (New York, NY, Harry N. Abrams 1989)

Email: jleft@orbitnet.com

This June 2020 Microsoft Word version of the *InfoSpace* white paper is a reformatted reproduction of the original 1991 and 1993 QuarkXpress version. Some typographical errors were corrected.