gIBIS: A Hypertext Tool for Exploratory Policy Discussion

JEFF CONKLIN and MICHAEL L. BEGEMAN
MCC, Software Technology Program

This paper describes an application-specific hypertext system designed to facilitate the capture of early design deliberations. It implements a specific method, called Issue Based Information Systems (IBIS), which has been developed for use on large, complex design problems. The hypertext system described here, gIBIS (for graphical IBIS), makes use of color and a high-speed relational database server to facilitate building and browsing typed IBIS networks. Further, gIBIS is designed to support the collaborative construction of these networks by any number of cooperating team members spread across a local area network. Early experiments suggest that the IBIS method is still incomplete, but there is a good match between the tool and method even in this experimental version.

Categories and Subject Descriptors: D.2.1 [**Software Engineering**]: Requirements/Specifications—*methodologies, tools*; D.2.2 [**Software Engineering**]: Tools and Techniques; H.1.0 [**Models and Principles**]: General; H.1.2 [**Models and Principles**]: User/Machine Systems; H.4.2 [**Information Systems Applications**]: Types of Systems—*decision support (e.g., MIS)*; H.4.3 [**Information Systems Applications**]: Communications Applications; J.1 [**Administrative Data Processing**]

General Terms: Design, Documentation, Management

Additional Key Words and Phrases: Issue-based information systems, planning

1. INTRODUCTION

There is a growing recognition that hypertext is an ideal model on which to base a support environment for the system design process. In the MCC Software Technology Program we have been working on a hypertext project called the *Design Journal* which is aimed at providing a team of system designers a medium in which all aspects of their work can be computer mediated and supported. This includes the traditional documents such as requirements, specifications, high-level design, and the design document itself, but it also includes such things as interviews with users, scenarios, design reviews, designers' early notes and sketches, design decisions and rationale, internal design constraints, meeting minutes, and so forth. The Design Journal places particular emphasis on the capture of the design rationale as the central aspect of the process which may serve to integrate all of the other documentation. In addition, our research is

Authors' address: MCC, Software Technology Program, 3500 West Balcones Center Drive, Austin, TX 78759-6509. ARPA: conklin@MCC.COM begeman@MCC.COM.

directed at the *upstream* of the design process, where most of the information is informal, and for which there is little technical support.

By design rationale we mean the design problems, alternative resolutions (including those which are later rejected), trade-off analysis among these alternatives, and a record of the tentative and firm commitments that were made as the problem was discussed and resolved. Our research has two thrusts: (1) to understand the internal structure of design decisions and the higher level dependencies which grow up among decisions, and (2) to address the interface problems inherent in capturing large amounts of informal design information and in providing effective methods for indexing and retrieval within that information. As part of the former thrust, we have been developing our own theory about the structure of design decisions, called *ISAAC*. As part of the latter thrust we have built a running prototype of the Design Journal, called *gIBIS*. At the time of the design of gIBIS, however, the ISAAC theory was not yet ready to be encoded as a running tool. Instead, gIBIS is based on a similar though somewhat simpler model of design deliberation called *Issue Based Information Systems* or *IBIS*.

2. THE IBIS METHOD

The IBIS method was developed by Horst Rittel [10] and is based on the principle that the design process for complex problems, which Rittel terms "wicked" problems, is fundamentally a conversation among the stakeholders (e.g., designers, customers, implementers) in which they bring their respective expertise and viewpoints to the resolution of design *issues*. Any problem, concern, or question can be an issue and may require discussion (if not agreement) in order for the design to proceed. Indeed, in the IBIS model it is this "argumentation" which constitutes the design process. (This does not preclude "arguing" with oneself, and gIBIS works as well for monologues as for dialogues.) Rittel developed this model over 15 years ago and has used it successfully in diverse design situations such as architectural design, city planning, and planning at the World Health Organization.

The IBIS model focuses on the articulation of the key *Issues* in the design problem. Each Issue can have many *Positions*. A Position is a statement or assertion which resolves the Issue. Often Positions will be mutually exclusive of each other, but the method does not require this. Each of an Issue's Positions, in turn, may have one or more *Arguments* which either support that Position or object to it. Thus each separate Issue is the root of a (possibly empty) tree, with the children of the Issue being Positions and the children of the Positions being Arguments.

There are nine kinds of links in IBIS. For example, a Position *Responds-to* an Issue, and this is the only place the Responds-to link can be used. Arguments must be linked to their Positions with either *Supports* or *Objects-to* links. Issues may *Generalize* or *Specialize* other Issues and may also *Question* or *Be-suggested-by* other Issues, Positions, and Arguments. As an escape mechanism, *Other* nodes may connect to any other node type with *Other* links.

A typical IBIS discussion begins with someone posting an Issue node containing a question such as "How should we do X?" That person may also post a Position node proposing one way to do X and may also post some Argument nodes which

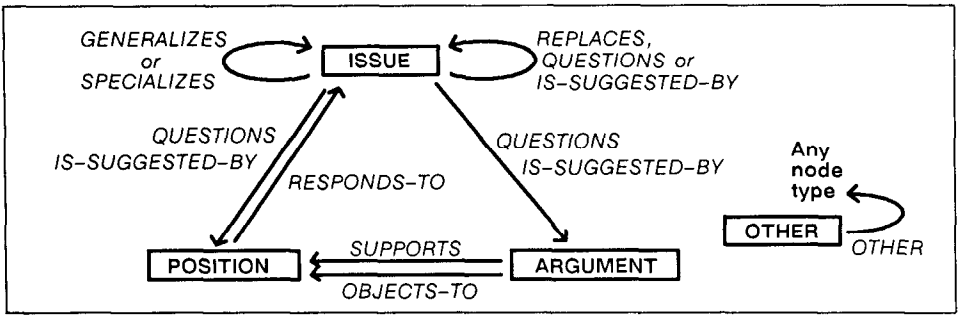

Fig. 1. The set of legal rhetorical moves in IBIS.

support that Position. Another user may post a competing Position responding to the Issue and may support that with a set of Arguments. Others may post other Positions or Arguments which support or object to any of the Positions. In addition, new Issues, which are raised by the discussion, may be posted and linked into the nodes which most directly suggested them. Figure 1 shows a state transition diagram specifying all of the legal moves within the IBIS method.

IBIS may be considered to be a Decision Support System (DSS), but only of a very unusual sort. There is no stopping rule, nor is there in the IBIS method a particular way of registering that an Issue has been resolved by agreement upon some Position. Rather, the goal of the discussion is for each of the stakeholders to try to understand the specific elements of each others' proposals, and perhaps to persuade others of one's viewpoint. The method makes it harder for discussants to make unconstructive rhetorical moves, such as "argument by repetition" and name calling, and it supports other more constructive moves, such as seeking the central issue, asking questions as much as giving answers, and being specific about the supporting evidence of one's viewpoint.

In implementing gIBIS we have made certain changes and extensions to the IBIS method to allow needed flexibility. However, we tried to change the method as little as possible. The IBIS method has been in use by Rittel in various design and planning activities for many years, and we felt it was important to push his method as far as it would go, understanding its strengths and weaknesses, before we started making any radical extensions.

The extensions to Rittel's IBIS in the current gIBIS tool are (1) an additional "Other" type for nodes and links as an "escape" mechanism for users who could not find a way to express a thought within the IBIS framework; (2) an additional "External" type for nodes that contain nonIBIS material such as requirements documents, design sketches, or code; and (3) the ability to let Positions "specialize" or "generalize" other Positions, and likewise with Arguments.

3. THE gIBIS TOOL[1]

There were three technological themes guiding our design of gIBIS. The first was an interest in exploring the capture of design history: the decisions, rejected options, and trade-off analysis—in short, the rationale behind the design itself.

[1] gIBIS is implemented in C on a Sun Microsystems workstation.

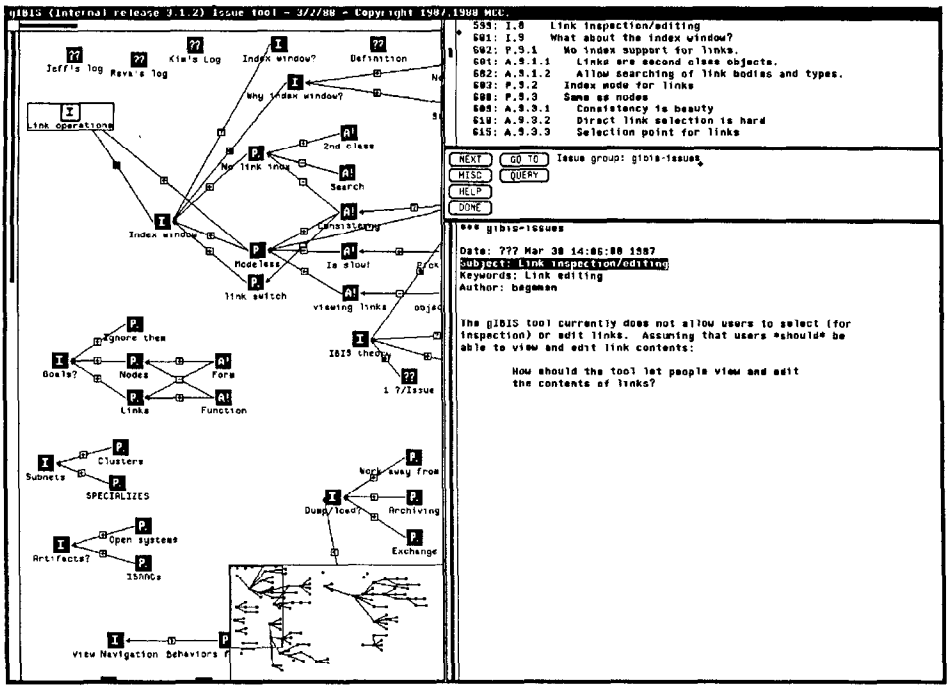

Fig. 2. The gIBIS interface.

For this purpose, the IBIS framework seemed a good starting point. The second theme was an interest in supporting computer-mediated teamwork and particularly the various kinds of design conversations that might be carried on via networked computers, email or the news [5, 9]. And third, we needed an application in which we would have a sufficiently large information base to investigate navigation (i.e., search and browsing) of very large information spaces. All of these factors lead us to this application and to the more specific requirements discussed below.

As can be seen in Figure 2, the basic gIBIS interface is divided into four tiled windows: a graphical browser on the left, a structured index into the nodes on the top right, a control panel below the index window, and an inspection window in which the attributes and contents of nodes and links can be viewed. This interface is somewhat unusual in that the only way to view the contents of a node or link is to select it, causing it to be immediately displayed in the single inspection window.[2]

3.1 The Browser

The browser provides a visual presentation of the IBIS graph structure. Nodes and their interconnecting links are displayed on a canvas of virtually unlimited

[2] The examples used throughout these sections are taken from the issue group "gibis-issues" which we have used to capture many of the design issues for the gIBIS tool itself. The reader is warned not to confuse statements which appear in the network (e.g., "The gIBIS tool does not allow users to select links") with the current state of the tool [this particular IBIS conversation resulted in an implementation for selectable links].

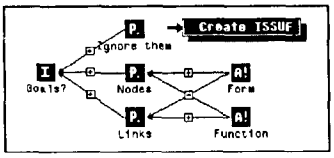

Figure 3

size. Most of the browser is dedicated to a local view of the network: a "zoomed in" view of the current area of interest which shows the full detail of the nodes and links. The lower right portion of the browser is reserved for a global overview of the data: a "zoomed out" abstraction of the entire network in which node labels, link-type icons, and the secondary links which make up the network's fine structure are filtered out. In addition to giving an overview of the entire network, the global view also indicates the scope and position of the current local view by a rectangular overlay (in this example, the local view extends down from the top left corner of the global view).

The canvas can be scrolled within the window area of the browser by the use of traditional scrollbars (seen at the top and left side of the browser) or more directly by "snap scrolling"—a method in which the user clicks the mouse anywhere within the local view to center that location in the window. This method allows the user to easily fine-tune the positioning of the display and to scroll diagonally without having to reposition two independent scrollbars. Scrolling to an area outside of the local view is also possible by directly repositioning the local view indicator in the global view window. Simply dragging the rectangle to a new area within the global view causes the local view to be updated appropriately.

The browser supports a direct manipulation style interface [13] to the nodes and links (i.e., display objects). Display objects can be selected simply by "clicking on them" with the left button of our three-button mouse. Selecting a display object causes it to become highlighted and boxed in the browser, its contents to appear in the inspection window (see Figure 2), and its index line to be scrolled to the top of the index window. A right-click on the mouse causes context-sensitive menus to be displayed. It is by these menus that objects are created, edited, deleted, moved, and so forth. As an example, let's being with the case in which the user presses the menu button when no object is selected. The menu of Figure 3 appears indicating that the only legal operation is Issue creation (i.e., beginning a new IBIS structure). By contrast, if a node of type Issue is selected, the menu changes as shown in Figure 4 to reflect the legal operations on Issues.

In this example, the user is choosing to create a follow-up node of type Position which, when populated and submitted, will be placed to the right of the selected Issue and will automatically be linked to it by a link of type Responds-to. Upon making this menu selection, the inspection window (lower right of Figure 2) divides itself in half and a Node Creation window, appears under it. This window is preloaded with a structured template to be filled in, as shown in Figure 5.

The user fills in the template's structured fields (e.g., Subject, Keywords, . . .) and then provides an optional elaboration of the node's subject (i.e., an unstructured node body). When the node has been completed, the user pushes the SUBMIT button in the control panel (which appears only during Node Creation/

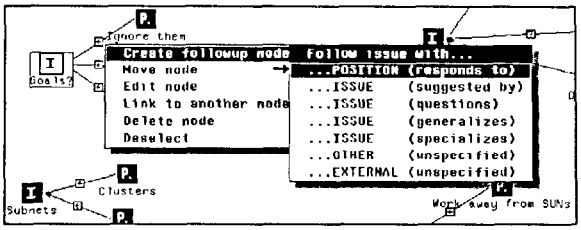

Figure 4

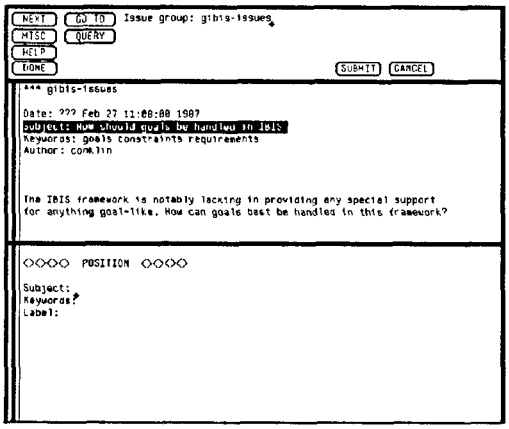

Figure 5

Editing), the node gets parsed and stored, and the browser and index windows are updated to include it.

When users follow the "Link to another node" menu item, a pullright menu appears (Figure 6) which constrains them to select from the set of legal outgoing link types for the currently selected node.

After the link type is chosen, the new link appears stretching from the source node to the current mouse position. The user moves the mouse to the destination node (the link follows the mouse by "rubber banding" across the canvas), and the user then drops the end of the link on its destination again by menu selection (Figure 7).

In addition to being able to select nodes and links, users can select canonical IBIS subnets (i.e., a single Issue followed by its set of Positions, followed in turn by their set of Argument nodes) as an entity as well. The tool provides support for the movement and automatic layout of these subnets as a whole. Further, gIBIS allows aggregation of these subnets into a single composite *IPA* node which provides additional structure to represent an analysis of the competing Positions and commitment to one of them (i.e., Issue resolution). This operation is depicted in Figure 8.

Although it has a structure and body all its own, the IPA node by default inherits its label, subject, and keywords from the root Issue of the underlying subnet. Selecting the composite causes traversal of the underlying IBIS subnet,

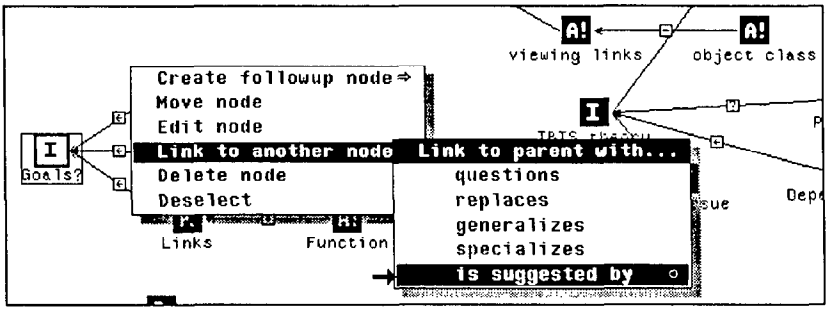

Figure 6

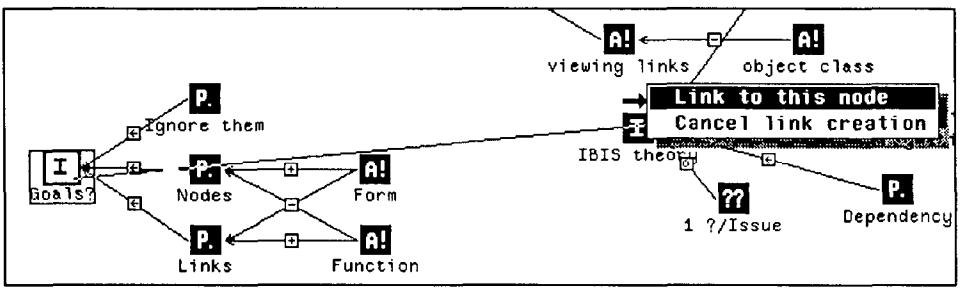

Figure 7

composing an "inherited" body which is shown in the inspection window along with the text which is specific to the composite. Since the inherited body of the composite can grow to be quite long for an aggregation of a large IBIS subnet, users can suppress (or reveal) its inclusion in the inspection window by using a function key.

It should be noted that the links from nodes in the subnet to nodes that are outside the subnet (i.e., the *extraIPA* links) are not displayed when the subnet is aggregated. This is because the semantics of linking Issues, Positions, and Arguments does not extend to IPA nodes, for example, it would be strange for an Issue to question the entire subnet subsumed by an IPA node. The fact, however, that *nothing* is shown reflects the state of the tool, not our thinking. We have considered several positions for the issue "What should be done about extraIPA links?" The first (which reflects the current implementation) is "Ignore them." The second, which is somewhat better, is "gIBIS should automatically build a *cable* between the IPA and any extraIPA nodes." Such a cable would have no semantics besides indicating that links exist between the underlying subnet and other nodes in the network. A user who is interested could then "click on" the cable to view its insides and hence discover what links it subsumes. The third position is "gIBIS should automatically create links which are semantic transformations of the extraIPA links in the underlying subnet." For example, the *questions* link mentioned before could be upgraded to a link of type "questions the Issue of." This is the most appealing alternative since it graphically conveys

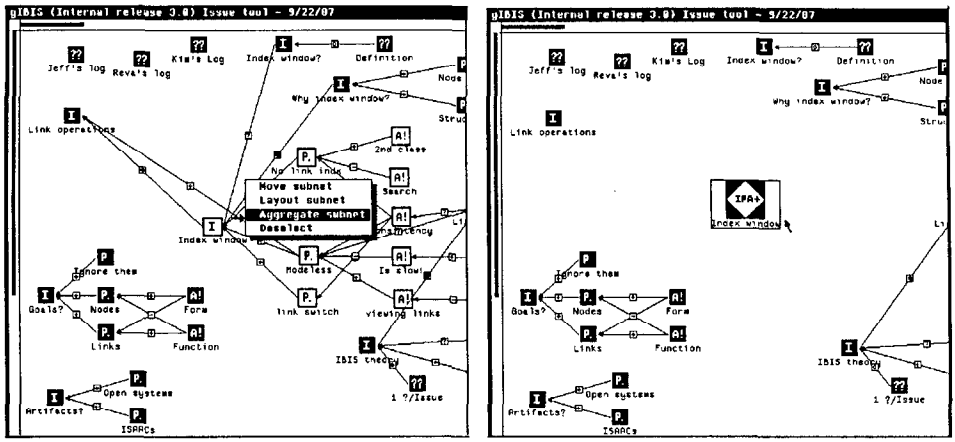

Fig. 8. A canonical IBIS subnet before and after aggregation.

```
599: I.8    Link inspection/editing
601: I.9    What about the index window?
602: P.9.1    No index support for links.
681: A.9.1.1    Links are second class objects.
682: A.9.1.2    Allow searching of link bodies and types.
603: P.9.2    Index mode for links
600: P.9.3    Same as nodes
609: A.9.3.1    Consistency is beauty
610: A.9.3.2    Direct link selection is hard
615: A.9.3.3    Selection point for links
```

Figure 9

more information than the other two, and it could be combined with the cabling approach of the second position.

3.2 The Node Index Window

The node index window provides an ordered hierarchical view of the nodes in the current IBIS network. The network is traversed following Primary links (discussed later) in depth-first order starting from each Issue (i.e., the root of each canonical IBIS subnet). The Issues, Positions, and Arguments are given sequence numbers like those one would expect to find in an outline editor [8]; for example, Figure 9 shows the Subject line for Issue 8 (I.8) which has no children, I9 whose first Position node (P.9.1) has two Argument nodes as children (A.9.1.1 and A.9.1.2), and so forth. Issues are simply ordered by creation date. The integers in the leftmost column are unique object identifiers and can be ignored. The view configuration panel allows the user to tailor the index information to reflect not only by Subject, but also by Author, Keyword, or node Label.

Nodes can be selected through the index as well as the browser. Clicking on a node's index line causes that node to become current: Its icon is highlighted in

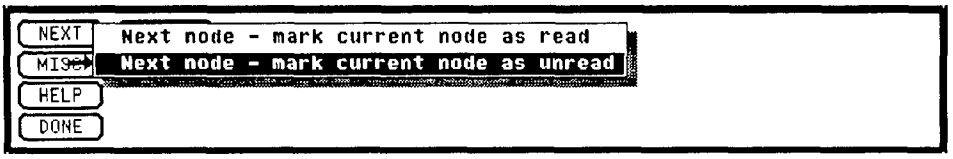

Figure 10

the browser, the canvas is scrolled (if necessary) to make the node visible in the local view, and the node's contents appear in the inspection window. Through this index-window-based access, we have provided a second browsing method which provides a linear, compressed view of the data in the network.

3.3 The Control Panel

The control panel, shown in Figure 10, is composed of a set of buttons which extend the tool's functionality beyond simple node and link creation. Each button has a menu hidden behind it which extends or tailors its basic function. The NEXT button, for example, will normally cause gIBIS to record that you have read the current node before displaying the next node in the network, but pressing the right mouse button while over the NEXT button causes this menu to appear—a slight extension of the basic functionality which leaves the current node marked unread.

For those functions that have no extended function, the menu is simply a longer explanation of the functionality provided. For example, the GOTO button that causes gIBIS to load a particular issue group's data into the browser has a simple help menu behind it which instructs the user to "Enter an issue group name and push this button."

3.4 Interface Configurability

The MISC button hides a grab bag of functionality. Of the functions available, we will describe one in depth: the TOOL CONFIG item, which allows the user to tailor particular aspects of the interface.[3] Upon selecting this item, a new window appears (see Figure 11). It contains the gIBIS configuration parameters, their current settings, and constraints on their legal settings. Configuration parameters are divided according to which window they affect: the index, browser, or inspection window. Figure 11 shows a user modifying the node attribute upon which the index window will be built. We wish to emphasize two major points about the browser now: the concept of a "primary" link and the use of color.

3.5 Primary and Secondary Links

Recall that when a node is created, it is usually linked automatically into the existing network of nodes. This automatic link, the *first* link which connects a node into the network, is considered to be that node's primary link. The user

[3] Other functions behind the MISC button allow users to send tool gripes/suggestions to the developers, mark all nodes as having been read/unread, linearizing and printing the IBIS net, and [un]subscribing to Issue groups.

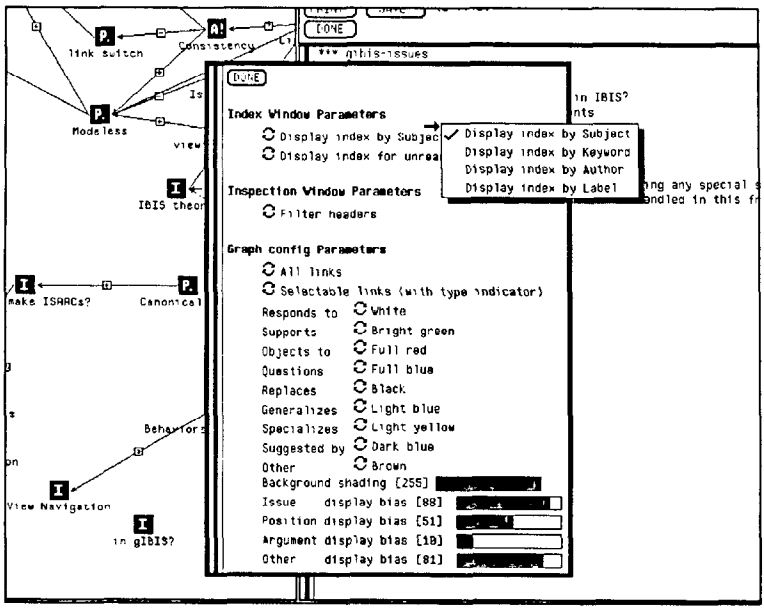

Fig. 11. The tool configuration window.

may later connect that node to others in the network by using the linking facility described above, but all subsequent links are considered to be *secondary* links and are distinguished from the primary link both visually and navigationally.

Filtering the secondary links from a canonical IBIS subnet results in a hierarchy, and this hierarchy is the basis for the index window's structured linearization. Take the case in which three Positions were created in response to an Issue (Figure 12). Two of the Positions have supporting arguments. In this example, the Positions were mutually exclusive; therefore, each Argument also objected to the other Position, and hence the authors created secondary links to make this explicit.

We have found that it is easier for the casual browser of an IBIS network to understand the network if, on first pass, the secondary links are turned off so that the browser only displays the primary links. The NEXT button leads the user through the network in the canonical IBIS order (the same sequence as the index window), and the primary-link browser view reinforces their understanding of how the current node relates to the surrounding conversational structure. For subsequent passes, the user may wish to enable the visibility of secondary links in order to understand the cross-relationships which the authors of the network have encoded. (In keeping with the design philosophy of tightly coupled windows, selecting a node with the NEXT button causes the same scrolling/highlighting as selection via the browser or index window.)

Although the current version of the tool binds the concept of primary link to the temporally first link which connects a node to the rest of the network, we believe that users should be able to later override this. As a network evolves, the semantic relatedness of its elements change (and hence do primary relationships).

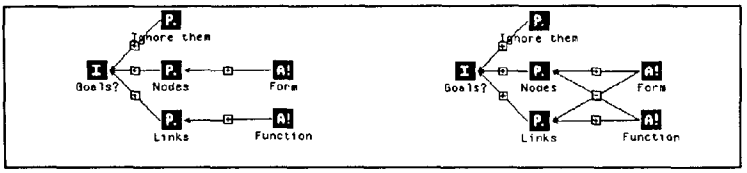

Figure 12

Once the structure of a network is established, an author might wish to go back and create a *guided path* through the information by redefining which are primary and secondary links.

3.6 The Use of Color

gIBIS was designed for use on SUN workstations with color monitors. On the basis of this, we chose to make use of color to indicate node and link-type information, as well as some special node states such as "currently selected" and "matches the current query." We also gave users the ability to configure the tool to customize the type-color mapping.

This flexibility caused some trouble at first and we quickly proposed (and then encoded in the tool) a set of standardized color mappings. Having colored nodes and links turns out to be one of the most compelling aspects of the tool. Users can quickly learn the type mappings for the most commonly used nodes and links, and type identification then becomes a rapid, unconscious activity. Although users occasionally change their mappings using the TOOL CONFIG panel for special purposes (like making some links invisible for presentations), they most commonly set their mappings and leave them alone.

Later, we needed to support a community of users with monochrome monitors, so we provided iconic information which duplicates the information encoded by color. Although the tool by default presents both color and iconic information to the user, both can be suppressed. Often, the color user will suppress link icons in order to make the browser appear less cluttered.

The use of color presents its own set of problems, however. The technique of type-to-color mapping is obviously limited to those users who have color display devices and are not themselves color blind. It is also limited to situations in which the number of mappings remains rather small. In our application there are nine link types, and the feeling is that we are near the limit of people's ability to reliably perform the mapping. By adding the link-type icons, the mapping complexity drops and more link types could be "safely" added.

More surprising, however, is the large machine-to-machine variation among the color monitors. The variation in overall brightness, convergence, and red, green, blue (RGB) gun saturation has eliminated the possibility of using a single, standardized set of color mappings for all machines. Color settings that produce bright, highly defined images on one screen can look very dark, muddy, and indistinct on another. To address this, we have provided the four sliders at the bottom of the TOOL CONFIG window which allow users to "fine tune" the colormap on their machines. Although this approach lets users construct

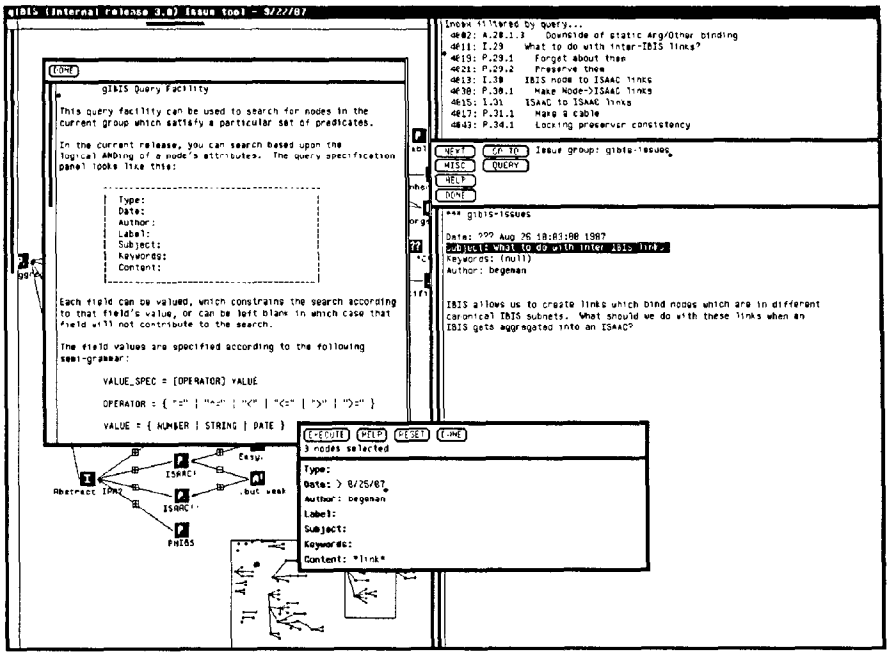

Fig. 13. The query control and help windows.

colormaps which adequately distinguish the types, we doubt whether we could ever reliably use more subtle shading schemes to communicate type information.

3.7 Search and Query

The last control panel feature we wish to discuss is the QUERY button. Pressing it results in the appearance of a small query-construction window as seen centered in Figure 13. The query window itself contains a small control panel and a query specification section which is query by example in that the user creates a proto-node against which those in the current IBIS net will be matched. The following example searches the network for those nodes authored by "begeman" after 8/25/87 which contained the string "link" in their body. Upon pressing the EXECUTE button, the query is parsed and evaluated, and the query's results are displayed both in the browser (selected nodes are turned a bright yellow in both the local and global views) and in the index window (the window only shows the index lines for those nodes which satisfied the query). The user can then examine those nodes using the standard navigation techniques described above. Pressing the HELP button reveals yet another window (obscuring the browser window below) which contains instructions on how to formulate queries, the query grammar, and a number of examples.

This query specification technique allows users to easily formulate node content searches based on the logical ANDing of predicates over node attributes. Although we have thought about extending the grammar to allow full Boolean expressions over the predicates (and have designed a visual interface for the specification of

structural queries), there has been so little demand for this that we have chosen to focus our implementation resources elsewhere. We feel that these more sophisticated queries may be required when the networks become very large, but experience shows that the simple query engine, which has been provided, is sufficient for searching over networks of moderate size.

3.8 Some Key gIBIS Requirements

gIBIS is primarily a vehicle for the exploration of Issue-based methodologies for the capture of design rationale. It was intended from the start to be used by small teams collaborating on "real" projects within the Software Technology Program at MCC. Because of this, we had the following constraints to design for:

(1) *The tool had to be reliable.*
We recognized that since people would be using gIBIS to capture information which was important to them (i.e., not for "toy" problems/experiments), our data storage had to be very reliable. Losing or corrupting an IBIS Network was considered to be an intolerable fault.

(2) *The tool had to support multiple concurrent users.*
As a tool to facilitate team collaboration, gIBIS had to provide true multiuser support. This meant shared, coordinated access to centralized IBIS networks, automatic notification of significant changes (e.g., new nodes) to the nets, access control and locking to prevent multiple updaters from corrupting the data, and "lightweight" IBIS groups that teams could create and share at their own volition. The user community had already been using the USENET news network to hold machine-mediated group conversations and was expecting richer and more powerful functionality from gIBIS.

(3) *The tool had to perform reasonably well.*
With the goal of producing a real tool for use on real problems, we had to provide reasonable performance from the beginning. Accessing an existing node or link's contents needed to be almost instantaneous, whereas larger tasks such as loading a new IBIS network into the browser or performing a query could take longer (10–15 seconds for a large network seemed reasonable). Our basic guideline was that the performance of the tool should not break a user's rhythm of creating and browsing a network.

(4) *We had to implement gIBIS with very limited resources.*
Because the entire project team consisted of $2\frac{1}{2}$ people (one author working full time on the methodology, the other author and a student working on the tool), we had to import as much functionality as possible. Wanting to concentrate our efforts on the interface and the method, we chose to build gIBIS on top of an existing relational DBMS.

3.9 The Choice of an RDBMS

Choosing a relational DBMS as our storage manager provides us with concurrency control, record-level locking, reliable data storage, fast access methods, and a reasonable search engine for free. In addition, the DBMS we have (SunUNIFY/SunSimplify) provides us with an uninterpreted data type: basically a field into

which we can store arbitrarily long passages of text, digitized voice, graphics bitmaps, or whatever. We are therefore able to store the body of a node as an integral part of a record in the database—something which many of today's DBMSs do not support. In retrospect, we feel that the decision to implement gIBIS on top of a DBMS has allowed us to focus on our research topic and has saved many months of development effort.

Unfortunately, the DBMS does not provide an adequate notification mechanism (*triggers* in DBMS parlance) to alert an application when a table or set of records gets modified (e.g., when a new node gets added to an Issue group). To overcome this, we had to build our own notification layer on top of the database. This layer keeps track of the state of the DBMS *with respect to each individual user*. When the database gets modified in such a way as to cause a change in any user's view of the data, those affected users are sent a notification, and their copy of gIBIS updates their view appropriately. In this way, gIBIS provides an effective, tight coupling between its users and their views of the evolving Issue networks.

3.10 References to External Data

Using a DBMS as our storage manager presented one major drawback, however: It closed the system. In essence, all of the objects which the Issue networks reference need to reside within the DBMS. Unfortunately, many objects that give rise to Issue-based discussions (like requirements or architecture documents) as well as those that result from these networks (such as code and documentation, the *artifacts* of design) are external to the database and hence out of reach. For this reason, we felt compelled to create a special *surrogate* type of node which allows gIBIS to reference external objects in a "blind faith" sort of way. A surrogate has two parts: a pointer to the external object (usually a fully qualified pathname to a file) and an optional display program which gIBIS should invoke to display the object. If the default display program is invoked, the external object is assumed to be a text file and is loaded into gIBIS's standard inspection window. If, on the other hand, the user specifies a display program, that program is invoked and passed the external pathname as an argument. Using this facility, external data and programs can be integrated into gIBIS. Some examples of external data, which we have seen, include simple textual documents, static graphic figures, dynamic simulations, a spreadsheet, and even a full-scale hypertext network managed by MCC's PlaneText hypertext system.

4. EXAMPLE

In this section we walk through a short example of issue deliberation using gIBIS. The example we use comes from the "gibis issues" issue group (as before) and starts with an issue about the treatment of *goals* in gIBIS. This issue and its discussion, that is, its canonical IBIS subnetwork, is visible in Figure 8, and close-up views of it are repeated in the figures below.

The example Issue is "How should goals be handled in IBIS?"[4] (see Figure 14), and its body expands on this question: "The IBIS framework is notably lacking

[4] In our gIBIS discussions we use the term "IBIS" to mean the methodological component of gIBIS, and not the more general sense of Rittel's IBIS approach. We are not critical of Rittel's IBIS.

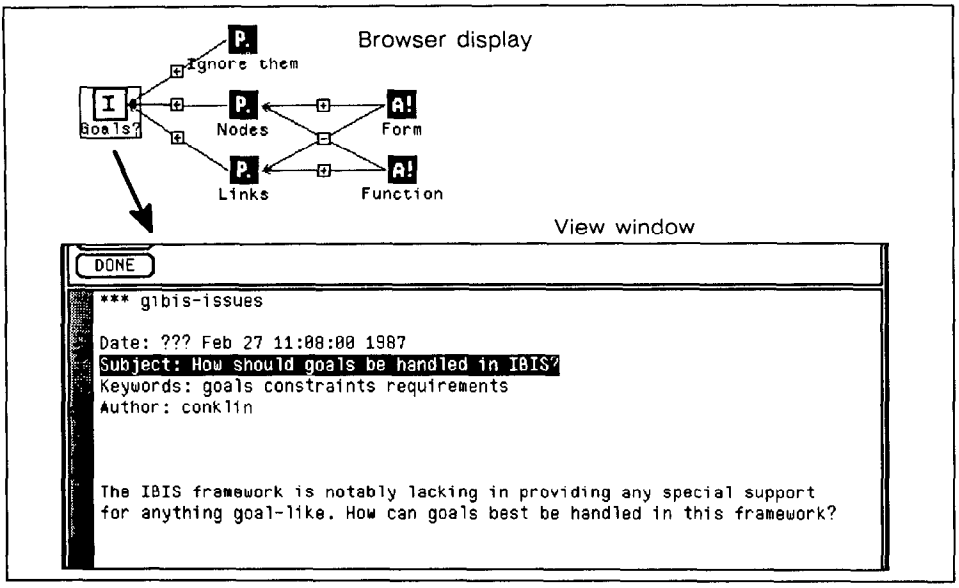

Fig. 14. The "Browser display" above shows the example issue network from the gIBIS browser. The Issue node has been selected, causing its contents to be shown in the "View window".

in providing any special support for anything goal-like. How can goals best be handled in this framework?" We chose this IBIS discussion for this example because it is self-contained within the network and because a minimum of background knowledge is needed to follow it. This issue is like several others in this issue group in that it explores the need to extend the basic Issue-Position-Argument object type trilogy with other first class types. (Note that it is not particularly important that you agree with the reasoning about the issue, or even fully understand it, to benefit from following the example.)

The first Position responding to this Issue (see Figure 15) proposes that goals should not be addressed at all by IBIS. This kind of Position illustrates a more general phenomenon that there is often a "null" Position for any issue which says "Do nothing" or "None of the above." This is distinct from another kind of default Position which says "This issue doesn't make sense." Although this Position would be a legal move in IBIS, it is much better to challenge the offending Issue by "Questioning" it with another Issue.

Notice also that the body of this Position, by claiming that "Goalness is just a feature of certain ideas—IBIS should not support it," goes beyond simply stating the response to also providing some justification for that response. Normally this should be avoided in IBIS; discussions are clearer and the whole method works better if Issues are *just* a single question, Positions are *just* a single response, and Argument nodes each contain a single objection or supporting point. In this case, the strictly proper thing to do would have been to create a supporting Argument node which said "Goalness is just a feature of certain ideas." However, there are practical limitations to the degree of fine granularity people are willing to make explicit—if nothing else, there is a time and complexity cost to creating each new node. Thus, just as it is acceptable for the Issue node to contain a

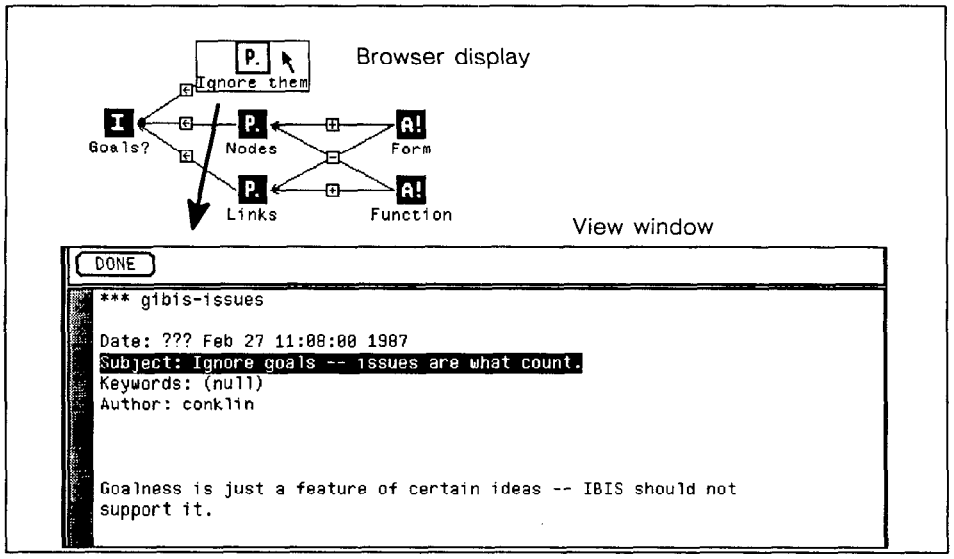

Fig. 15. Clicking with the mouse on the top Position node reveals its contents in the view window. This is the "null" Position for this Issue.

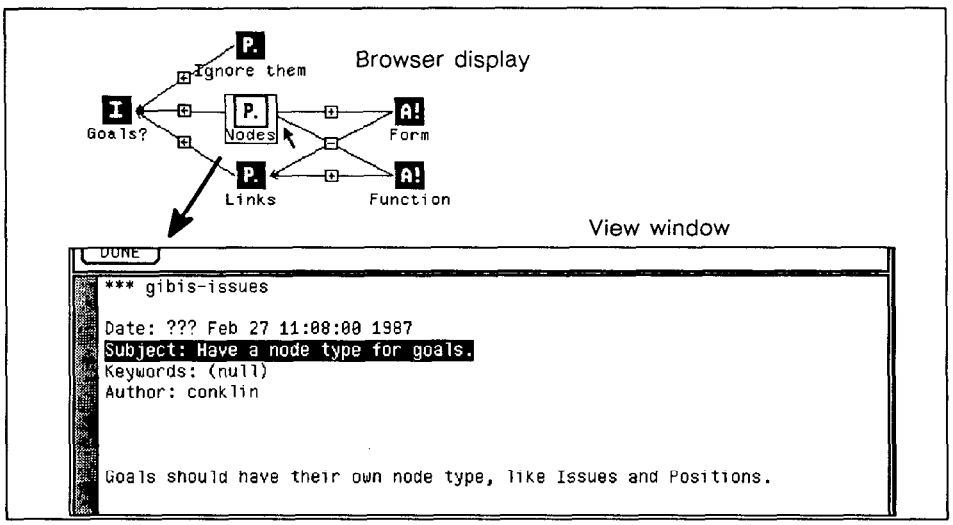

Fig. 16. The second Position proposes to make a node type for goals.

noninterrogative line of background material, Positions must be allowed to sometimes contain their own support (if it is very brief).

The second Position, "Have a node type for goals," proposes to treat goals as a new node type in gIBIS (see Figure 16). This position is short and to the point and leaves its support entirely to the Argument node which "Supports" it (see

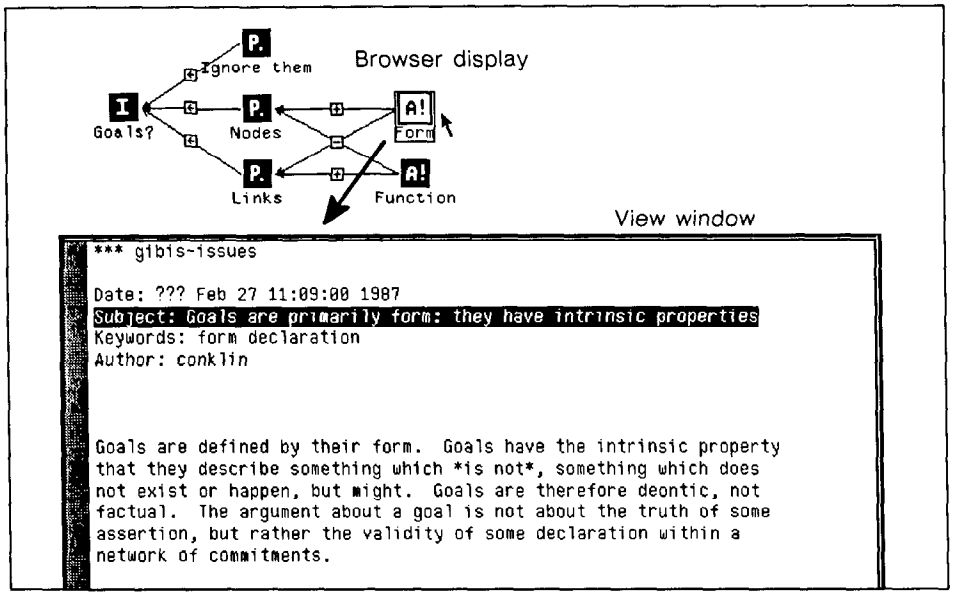

Fig. 17. This Argument node has a primary Supports link to its Position and also has a secondary Objects-to link to the third Position.

Figure 17). This Argument node argues that goals should be treated as first class objects since they have properties. Note that there is no clear logical necessity between the Argument "Form" and the Position "Nodes." Rather, the text of the Argument simply tries to lend support, somewhat informally, to the "Nodes" Position. If this Issue were discussed further, one way that it might grow would be for further Arguments to be added, supporting and objecting to the various Positions, and bringing out other criteria and evidence about them. The force of Arguments in IBIS is not to establish truth or falsity, but rather to persuade.

In Figure 18 the third Position "Links" proposes that goals be brought out through the relationship among IBIS objects, in particular between Positions (in other words, that a given Position can function as a goal and another as a solution which somehow satisfies that goal.)

This is an example of two Positions ("Nodes" and "Links") which are mutually exclusive—both cannot be selected as the resolution of the issue. This is the most common relationship among the Positions of an Issue, although there is nothing inherent in IBIS that requires there to be no overlap among the Positions. One kind of Issue that invites overlapping Positions, for example, is the kind that asks "What are the goals of this project?" Although it is not completely clear from the browser view of the subnetwork, each of these Positions is supported by "its" Argument (the one connected by a primary link), and is objected to by the other Position's Argument. This is a simple example of the graphical view revealing logical and rhetorical relationships among IBIS elements.

The second Argument (Figure 19) rounds out the discussion by supporting the "Links" Position. While writing this paper one author noticed that this Argument

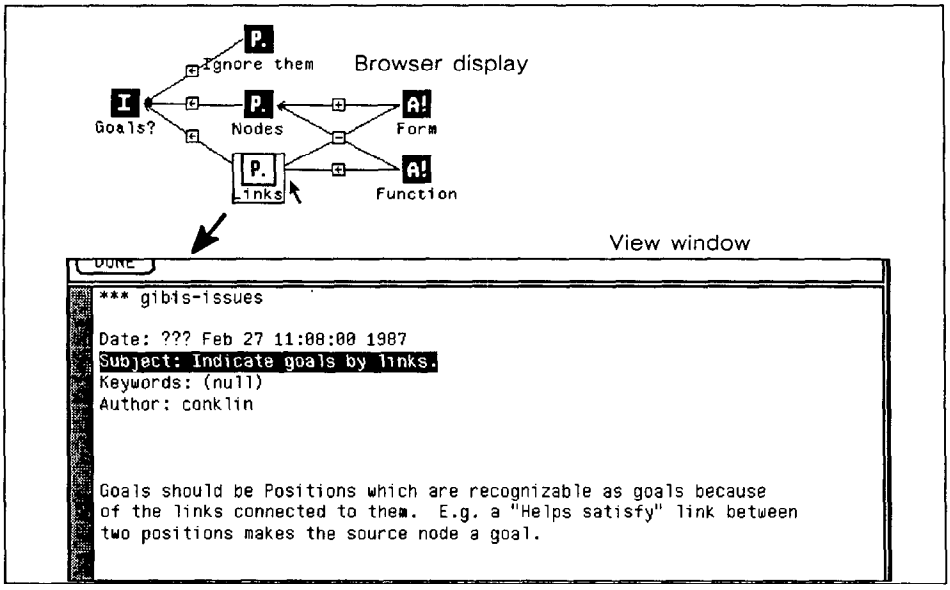

Fig. 18. The third Position. The crisscross pattern of objects and supports links suggests that the second and third positions are mutually exclusive.

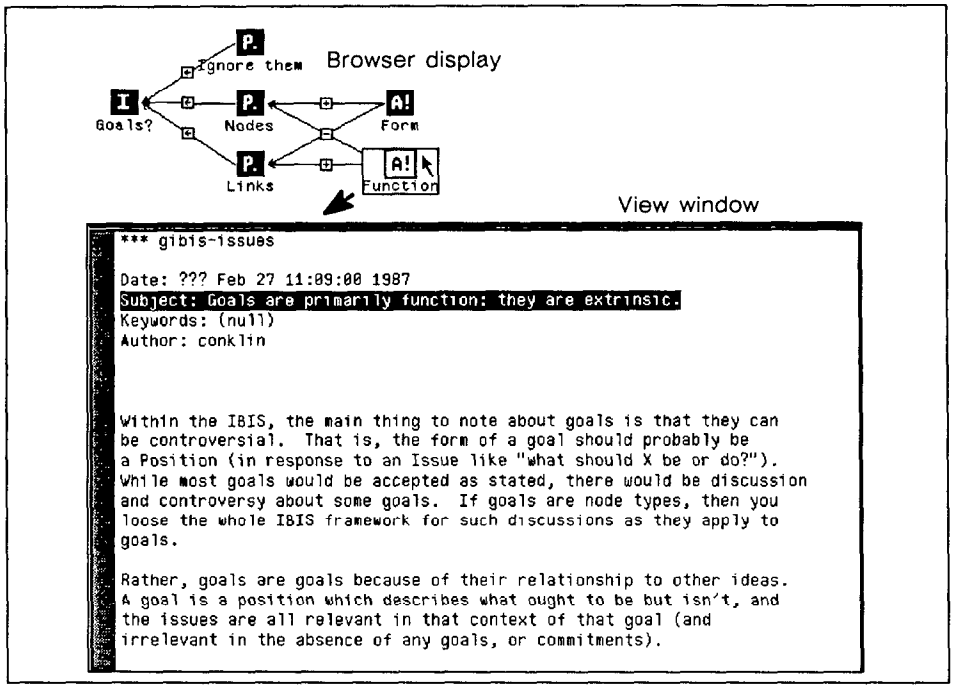

Fig. 19. The supporting argument for the third Position.

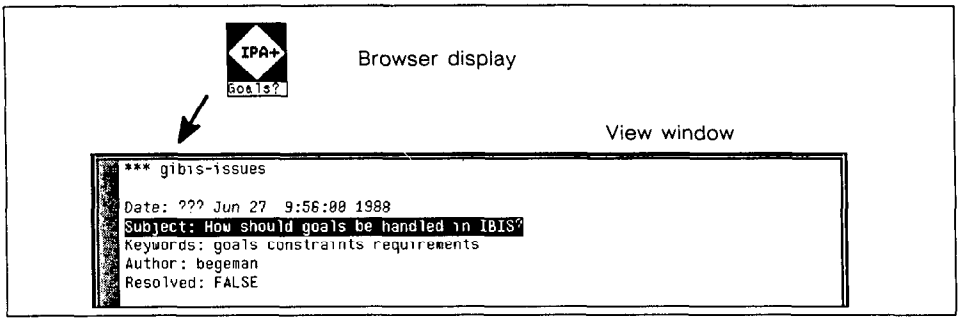

Fig. 20. The "IPA" aggregate node which summarizes the discussion of the Issue. The "Resolved:" field still has the default value "FALSE".

node actually contains two different arguments: the first is primarily directed against the "Nodes" Position and says that if you treat goals as IBIS nodes, then you are treating them as noncontroversial elements, which undermines the whole IBIS approach of treating nothing as a priori true; the second argument (in the second paragraph) argues that nothing is inherently a goal, and that certain ideas (in IBIS, Positions) are goals by virtue of their relationship with other ideas. In a moderated issue group the author of this node would probably have been asked to place these arguments in separate Arguments.

The IBIS method does not deal directly with issue resolution. Figure 20 shows the gIBIS IPA node for this issue where, if the issue were resolved, the description of that resolution would be placed. The description would tell which of the three Positions had been selected and perhaps also provide additional justification. It is interesting to note, however, that since no specific support for goals (such as is described in this example) has ever been implemented in gIBIS, the issue was *de facto* resolved, at least temporarily, to the first Position, "Ignore them."

5. OBSERVATIONS

In this section we wish to present some trends and observations of the uses, strengths, and weaknesses of this hypertext tool. These are preliminary findings. In this section we have tried to be as candid as possible about the weaknesses and research problems of both the IBIS method and the gIBIS tool. We hope that this candor does not create an overly negative impression about what we feel is a very positive research effort.

It is important to keep in mind that none of our users was, at least initially, more than passingly familiar with the IBIS method itself, so there was quite a bit of learning and experimenting going on while users constructed their networks. Indeed, it could be said that most of our users regarded *themselves* as experimenters, exploring different ways of working and using the tool, conventions, and so forth.

5.1 Network Structure

In this section we describe the usage of the gIBIS tool during a one-year period, from mid-February 1987 to mid-February 1988, in terms of statistics on the

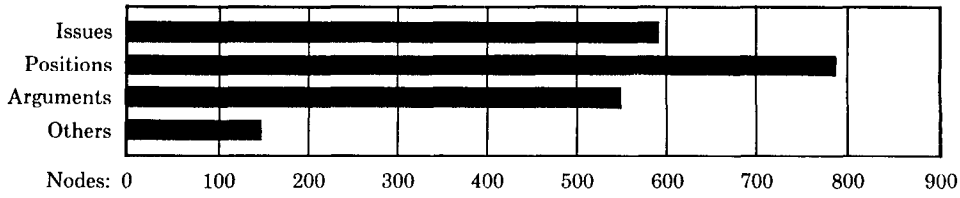

Figure 21

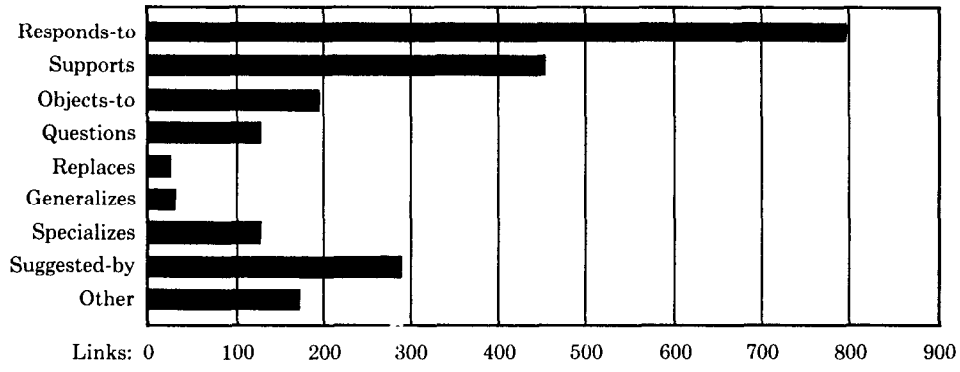

Figure 22

networks. Thirty-two people participated in the creation of 33 issue groups in all.[5] (One issue group, not presented here, took part in an experiment in which gIBIS was used as a box-and-arrow tool, but the IBIS node and link types were ignored.) As of February 1988, 2091 nodes had been created in roughly equal numbers of Issues, Positions, and Arguments across all of the issue groups, as depicted in Figure 21. Thirty-one percent of the Issue nodes had no Positions, and the remainder of the Issue nodes had, on average, 1.9 Positions. On the other hand, 59 percent of the Position nodes had no Argument nodes, and the remaining Positions had an average of 1.7 Arguments.

Connecting these nodes were 2214 links, in the proportions shown in Figure 22.

While it is still too soon to draw any conclusions from these numbers, they do at least indicate that users had a greater tendency to post supporting Arguments than objecting ones, and that it was somewhat more natural to specialize nodes than to generalize them.

Finally, we wish to indicate the levels of individual participation within the fifteen largest issue groups during this trial period. Each issue group is shown with a code letter (e.g., "A"), and the bars above that designator show the number of nodes contributed by each participant, with one bar per participant. Thus, issue group "B" had two participants, one of whom posted 190 nodes, the other of whom posted 30. As Figure 23 suggests, many issue groups were constructed largely by a single participant, while a few issue groups had reasonably balanced

[5] As of late August 1988 there were 48 issue groups in all.

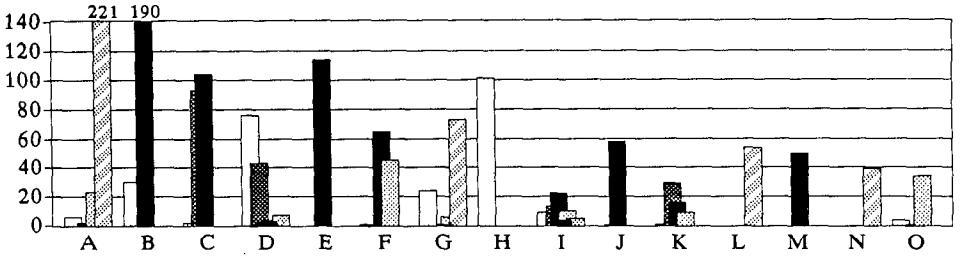

Figure 23

participation. This reflects a pattern of gIBIS usage that falls into two categories: Some people used the tool primarily as an isolated hypertext tool for structured thinking and design; others used the tool primarily as a vehicle for structured communication.

Issue groups were created on a wide variety of topics, including

Research planning: Exploration of problems in doing requirements by analogy
Conceptual analysis: Analysis of a journal article on hierarchical design
Project review: On-line "meeting" between reviewers and the project being reviewed
Requirements analysis: For an object-oriented database design project
Software design: Reconstruction of the early design of an elevator controller using a specific design method
Customer/designer interaction: Ongoing negotiation about the design of gIBIS
Large group discussion of organizational problem: Whether or not our research program should move to another location

5.2 The Usefulness of Explicit Rhetorical Structure

One early surprise about the IBIS method was that, although most people felt that IBIS was awkward and overspecialized before learning to use it, many users of the gIBIS tool have come to regard IBIS as a powerful method for research thinking and design deliberation. Users who worked alone in an issue group reported that the Issue–Position–Argument framework helped to focus their thinking on the hard, critical parts of the problem and to detect incompleteness and inconsistency in their thinking more readily. Users who collaborated in issue groups reported that the structure that it imposed on discussions was very useful and served to expose "axe grinding, hand waving, and clever rhetoric." They also valued the tendency for assumptions and definitions to be made explicit.

Some of these advantages can be traced to the *semistructured* nature of IBIS networks [12]. The writer is aided in structuring a complete message without any constraint on expressibility, while the reader is provided with recurrent structure in the textual material that aids both search and comprehension. Both reader and writer are aided by the explicit rhetorical structure of IBIS, which makes apparent at least the general structure of an unfolding discussion. Indeed, we feel that a distinct advantage stems from the particular structure that IBIS provides. That is, there is a good match (though this will be difficult to prove) between

some of the cognitive structures and processes of design and the three node types and nine link types that compose IBIS.

However, as we press gIBIS into service in an ever wider variety of design applications, we find that there are some major shortcomings. There is no specific node (or link) type for goals and requirements, and several users have requested support for these. There is no particular support for making a decision (or reaching consensus) among the various positions of an issue. Design decisions usually result in the addition of solution elements to the design itself (e.g., code, module structure), but these elements are not supported by gIBIS and must be stored externally to the tool, that is, artifacts cannot now be integrated with, or linked to, the decisions that lead to them. All of these extensions were anticipated during our theoretical work on the Design Journal, though our experiences with gIBIS are suggesting some changes to our theory of design rationale.

5.3 The Synergy of Tool and Method

We have observed an interesting synergy—a mutual facilitation between tool and method. The noncomputerized IBIS method is cumbersome and would not have reached the popularity that it has here in our lab without the high speed gIBIS tool to support it. On the other hand, gIBIS is not the only hypertext system available in our environment, and yet it has achieved a wider and more prolonged usage in a much shorter time than has PlaneText [4], the other system. We speculate that there is a particularly good match between the requirements of the IBIS method and the hypertext facilities of the gIBIS tool.

For example, one of the clear successes of this project has been the use of color to indicate the type of IBIS nodes and links. Perhaps this is in part because there are just a few distinct node and link types in IBIS, and each has a reasonably well-defined semantics, so that the browser display can use bright primary colors which, after some familiarization, come to have a strong association with the semantics. Evidently, despite its narrow design and rigid functionality, gIBIS provides facilities which can be quickly learned and appreciated by researchers working on ill-defined design problems. This experience has led us to begin the design of a *toolkit for building gIBIS-like systems.* The basic functionality for such a system would include typed nodes and links (user defined); customizable interface views of the network that map object features into display features in a general way; high-speed real-time interaction (with notification, soft locks, etc.) among coworkers in the hyperdocument; rapid global search of the entire network, both for node contents and network structure; and strong support for classificational hierarchies and composite nodes. This effort is an integral part of our hypertext research [6]. One local step in the direction of a more flexible tool is "Germ" [2], an extension of gIBIS in which the node and link types and their legal connections are defined in an external schema file, thus allowing any node-and-links model to be supported using a gIBIS-like interface.

5.4 The Dangers of Premature Segmentation

One common but subtle difficulty in hypertext systems is that it is sometimes unnatural to break one's thoughts into discrete units, particularly when the problem is not well understood and those thoughts are vague, confused, and

shifting. With gIBIS this effect is pronounced, because the IBIS method imposes a rather austere selection of node and link types on the user. In particular, design conversations often feature commitments of the form, "Let's try X. It has advantage Y." Notice that this is a Position and its supporting Argument, with no Issue articulated for the Position. Some users have complained that they do not always see the Issue or Position immediately, and that they would like to have a "protonode" to simply record ideas before structuring them.

To some extent this complaint is to be expected: The tool supports a method which demands that one think within a particular framework (e.g., focusing on issues without necessarily resolving them), and this can be disruptive. However, even some users who are quite familiar with the IBIS method still insist that they occasionally require support for recording unstructured material.

It has been widely noted ([1, 15]) that the early phase of consideration of a writing or design problem is critical and fragile and must be allowed to proceed in a vague, contradictory, and incomplete form for as long as necessary. However, any insights and breakthroughs should be immediately (and reversibly) capturable, and the tool should support the emergence of a coherent structure as that develops in the designer's mind.

Ultimately, of course, it will be valuable to have teased apart these elements into separate issues, positions, and arguments. But in the moment of struggling to solve the problem, the cognitive overhead required to segment the "muck" into discrete thoughts, identify their types, label them, and link them is prohibitive. We are considering the addition of a "brainstorming" mode in which it is easy to jot down snippets of text (and perhaps graphical sketches), providing only minimal organization to these elements. This will lead to the development of tools to aid in the structuring of this "raw" material into the IBIS framework.

5.5 Capturing Issue Resolution

A somewhat complementary problem exists at the other end of the deliberation process: How should the resolution of an issue be represented and displayed? The IBIS method suggests that an issue is resolved by selecting (it does not matter how) one of the positions that respond to it as being "the right answer," or at least "the position we are committed to for now." This could be represented as marking the Position node as "SELECTED" and could be displayed simply by marking such Position nodes distinctively in the browser, e.g., by giving them a somewhat different color from unselected Positions.

We have recently added a resolution display feature but do not yet have enough experience with it to report on user acceptance. We combined indicating resolution with the aggregation in IPA nodes so that once an issue's discussion is aggregated into an IPA node one can indicate that the Issue is resolved. At the moment, this is done by changing the value of the "Resolved:" field to TRUE and adding a short piece of text indicating which of the Issue's Positions was the one selected as the resolution.

However, we suspect that it will not always be sufficient to simply flag the selected Position. One reason is that the rationale for adopting a particular conclusion may require more explanation; it may be that not all of the argumentation occurred within the gIBIS tool, or that there is a broader perspective for

the resolution than that in which the pros and cons of the established Positions were argued. Similarly, the resolution of an issue sometimes transcends the fixed options which were originally perceived to be available. Such emergent resolutions often combine elements of the original options, and often they abandon assumptions or presuppositions that were hidden in those options. Sometimes when such "breakthroughs" occur there is no need for further discussion: It is clearly the right solution. gIBIS needs to allow for such leaps in the argument without unduly constraining the Issue to a well-structured resolution. This may be as simple as providing the kind of free-text annotation of an Issue-Position-Argument tree described above, or it may require a facility for marking such discussions as "irrelevant in light of Position X."

5.6 Integrating Artifacts with Reasoning

Often the reasoning process that is supported and captured by gIBIS is based upon some document and results in the creation or modification of other documents. As an example from the software design domain, if the "input" problem is a problem statement and the target "output" is a requirements document which formalizes the requirements, the intervening design process can be viewed as issue-based deliberation. However, these traditional documents or "artifacts" should be linked into the issue-based network. Artifact elements that are the basis of some IBIS issue should be linked to the Issue node. Argumentation about the issue which draws on existing artifacts for evidence should be represented as a link from the Argument node to the artifact(s). And the resolution of the issue, an assertion in some Position node, should be linked to the artifact element which implements that assertion. (There is an application built on top of NoteCards, called Instructional Design Environment, or IDE, which is a design support tool for training courses, and which captures design decisions as first-class objects. IDE has a very explicit theory about how the decisions integrate with the other design elements [14].)

Currently, artifacts are integrated into the IBIS network by putting the documents into Other nodes and using Other links. This has been satisfactory when (a) the document was stable enough to permit isolation into a gIBIS Other node, and (b) the document was short enough to determine what specific piece of it was being referenced by the IBIS nodes. However, no one has tried to use gIBIS to capture or support their reasoning during programming or debugging. The External node might be more useful for this—at least the document can be a conventional file. But the problem of referential precision remains: Links in gIBIS are strictly node-to-node. What is really needed is the ability to link to a specific section or set of sections of the document text, as is commonplace in more conventional hypertext systems like NoteCards [7] and Intermedia [18].

The problem of "opening" gIBIS up so that it can link to and be linked to by other documents and tools is an important one, but one which raises some of the hardest problems facing hypertext technology. Among other things, it suggests that gIBIS, rather than being a separate tool for "deliberative occasions," should be a mode or utility within a hypertext-based environment.

5.7 A Problem with Context in Nonlinear Documents

One of the chief elements of our experimentation with gIBIS is to investigate the use of hypermedia as a medium for cooperative work. In some cases where several users worked cooperatively in a shared issue group an unexpected problem emerged. Unless each author was careful to write clearly and completely, the readers found that, while they had a sense of understanding the individual nodes, they could not follow the thread of the writer's thoughts as it wound through several dozen nodes. That is, there was the sense that the hypertext tool forced ideas to be expressed in a fine-grained, separated manner, and that this obscured the larger idea being developed by the author.

In one respect this is the familiar problem of cognitive overload common to many hypertext systems: The freedom of choice inherent in branching documents simply requires greater care from the writer and attention from the reader. Another factor could be the unfamiliar separation of Position and Argument (i.e., idea and justification) in IBIS.

But we suspect that there is a related but more subtle issue here: that traditional linear text provides a continuous, unwinding thread of context as ideas are proposed and discussed, a context in which the writer is directly, if unconsciously, working to guide the reader to the salient points and away from the irrelevant and distracting ones. Indeed, a good writer anticipates the questions and confusions that the reader may encounter and carefully crafts the text to prevent these problems.

The hypertext (or at least gIBIS) author, however, is being encouraged to make observations discrete and to separate them from their context. Indeed, we have observed the problem that the gIBIS writer, being in a hurry to capture a design issue and its analysis, sometimes writes only the bare minimum necessary to record the essence of the issue, positions, and arguments (presumably with the intention of returning later to "clean up" the network and make their postings more readable). Even the careful author, however, is in danger of not anticipating all the various routes by which a reader may reach a given node, and so may fail to sufficiently develop the context necessary to make the node's contents clear, if not compelling.

Several techniques may be useful in ameliorating this problem. The notion of a "path," described by Bush [3] and Trigg [16], may provide a sufficient linearization that readers of the network can glean a useful context from segments of a network (see the article by R. H. Trigg, pp. 398). Also, we are experimenting with higher level constructs that aggregate a set of nodes. The IPA node type described above combines the display of all of the nodes of an IBIS subtree (the Issue, its Positions, and their Arguments) into a single node and allows additional IPA-specific text to be appended as well. This will linearize the discussions of individual issues and reduce the sense of fragmentation one sometimes has when reading a gIBIS network, but it is probably not sufficient to create or restore the context in which those nodes were created. Finally, part of the context that the writer has in mind is the relative importance of the various points being presented, and we are investigating ways of incorporating a simple importance metric directly into gIBIS nodes. In one methodological experiment, gIBIS users experimented with providing one of the three keywords "HI IMPORTANCE," "MED

IMPORTANCE," or "LO IMPORTANCE" in each node they create. This measure could be used to guide the reader to the most salient points first (see also [11]), as well as to control the level of clutter in the browser display.

5.8 Annotative or "Meta" Discussions

It is a commonplace of human conversations to "go meta" and make a comment on the *process* (as opposed to the content) of the discussion, for example, "But that isn't the issue here." Similarly, in IBIS discussions there is sometimes a need for a metadiscussion when a participant in an issue group feels that someone has poorly or inaccurately used the IBIS structure to present ideas. For example, if B feels that the content of A's Issue node is, in fact, two Issues and a Position about one of the Issues, B needs some way to express this, and, in fact, to initiate a discussion about this "metaissue" with A within the context of the issue group.

In fact, it has been noted that there are three levels of description for collaborative work: *substantive* (the content of the work), *annotative* (comments about substance), and *procedural* (comments about procedures and conventions for use of the medium) [13]. In an IBIS framework, all three levels can theoretically be treated as Issues and their argumentation. For example, in the case of B's disagreement with A, B could post an Issue, connected by a "questions" link to A's Issue, asking "Isn't this really two Issues and a Position?" Although this is a perfectly valid move in the IBIS rhetorical framework, it has drawbacks. This Issue is by its nature metasubstantive, although it is unclear whether it is annotative or procedural. But by placing it in the network, B creates an Issue that adds complexity to the browser display without illuminating the substance of the problem being discussed and initiates a discussion that may well lead to a change in the network, after which the metadiscussion will have only historical interest.

There are several ways of resolving this problem. One is to have special metalevel Issue, Position, and Argument nodes to distinguish these metadiscussions from the substantive ones. Or we could use regular IBIS nodes but provide a mechanism by which any node could be labelled as "only of historical interest"; such nodes could be archived, or at least have their display suppressed so that they would not normally be visible. Third, we could provide each node with its own "metalayer" so that discussions about the match of the node's contents to its IBIS type would be tracked in this specialized part of the node, which again would only be displayed upon request. This approach does not require the metadiscussion to follow the IBIS format since metacomments are simply a kind of generic annotation. And this seems to be the best course to follow, given that the meta-discussions we have observed in gIBIS are usually more of an action-based conversation (e.g., "Please do *X*", "OK, done") than an issue-based conversation. At the moment we are experimenting with a simple version of this third option: Any user may append a "metaline" at the end of the body of any node and may then begin an annotative or procedural discussion in that part of the node by entering his comments and signing them. The author of the node might append a response at the end of the node or might simply revise the network to correct the structural error.

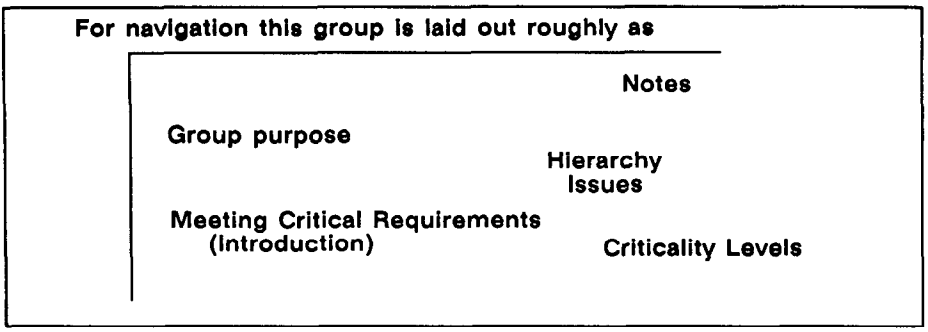

Fig. 24. One user's node containing the browser layout map.

5.9 Macrolevel Organization of the Browser Space

One of the "hot issues" in hypertext research is the problem of the effective use of a graphical browser to navigate in networks that have more than a few dozen nodes. This is linked to the more general problem of disorientation [4] but bears particularly on the visual and spatial aspects of disorientation in a large data space. The gIBIS browser ran into these difficulties as well, of course, since the problem is largely independent of implementation. (Note that the global view mechanism described above, which shows a highly reduced view of the entire network, was added after the data for this paper were gathered. Thus although this feature has enjoyed very high user satisfaction and acceptance, our observations here are based on use of the gIBIS tool with only the small-scale browser.)

In its current form the gIBIS browser must share the screen space with the node viewing and control panel windows, and, therefore, cannot occupy more than about half of the screen. This provides the browser enough room to show no more than 40 to 50 nodes at one time. Although this may sound like a lot of nodes, recall that the browser only displays a very brief one or two-word label for each node in the browser. To get any detail about a node requires mousing it and reading its contents in the node viewing window.

For some users this feature made 40–50 nodes the largest network that they wanted to try to work in. Two users, however, developed a way to partially overcome the spatial disorientation problem. These users divided their networks into regions that were meaningful in the terms of the problem they were working on. Nodes were classified according to broad semantic features, and these features were also identified with regions of the browser canvas. For example, one user created an Other node, labelled "LAYOUT", in which she placed a map of her network (Figure 24).

This technique has several advantages. Surprisingly, users reported that the effort of coming up with a layout revealed aspects of their problem that were not obvious beforehand. But the map also organized their work within gIBIS by easing the problem of deciding where to place new issues and by providing a natural basis for finding nodes whose location *and* keyword information had been forgotten. This is an aid to navigation of large networks that the authors had

never considered and which we will pursue supporting directly in the tool. This experience has reminded us of the value of having "real users" testing out new tools.

5.10 Coping with Change in an Evolving Network

Any database has to have mechanisms for managing changes to the data it contains. Often this is at best a versioning scheme which allows older versions of the data to be marked and archived. In an application like gIBIS, however, the issue of change is of unusual importance because the very nature of an "issue base" is that it is a vehicle for an evolving discussion in which older material may be accurate and highly important, inaccurate and only of historical interest, or anything in between. For example, the original form in which an Issue was framed may have been biased toward a particular Position, or may have contained a presupposition that was later made explicit and rejected. How should this "outdated" form of the Issue be handled?

In some cases the Issue and its discussion subnet may be isolated and simply wrong, in which case it will be easy to decide to archive that subnet and delete it. But more often there will be parts of the subnet that are wrong, misleading, or irrelevant, and others which are still quite relevant or important to the network, and which are directly linked to network regions where discussion is quite active. How can these partially invalid discussion segments be prevented from "poisoning" the network?

The answer seems to have two parts. One is that we need mechanisms for systematically indicating the age and relevance of network material, such as displaying older nodes as yellowed and more frequently visited nodes as frayed. Like the mechanisms suggested above for importance, salience, and confidence, age and relevance would be somewhat subjective measures that could only be partially automated. The other mechanism for managing change is completely human: As issue networks grow in size and importance, it will become increasingly important for organizations to have people whose job is to maintain the currency and hygiene of the issue base.

CONCLUSIONS

We have described the IBIS method, the gIBIS tool, and some preliminary observations about the use of the tool. Our experiments with gIBIS are informing our theory about the structure of design decisions and design rationale and are providing us with important insights about the design of the Design Journal, a hypertext-based environment for system engineering which we will continue to design, prototype, and test in the next few years. More important, our experiences suggest that the computer is indeed a powerful medium for collaboration and debate among members of a team, but that the integration of computers into the fine detail of real work is attended by some severe breakdowns. Some of the breakdowns are due to inadequate interfaces, others to inappropriate underlying representations, and still others to insufficiently rich models of work practices and methods. Our experience with gIBIS suggests that we are just at the beginning of a long but exciting path, which will culminate when we have succeeded in

making such tools as effective and transparent in structuring communication as the telephone has grown to be in simply transmitting it.

REFERENCES

1. BROWN, J. S. Notes concerning desired functionality, issues and philosophy for an AuthoringLand. Xerox PARC CIS Working Paper, Palo Alto Research Center, Palo Alto, Calif., 1982.
2. BRUNS, G. Germ: A metasystem for browsing and editing. MCC Tech. Rep. STP-122-88, MCC, Austin, Tex., Apr. 1988.
3. BUSH, V. As we may think. *Atlantic Monthly 176*, (July 1945), 101–108.
4. CONKLIN, J. Hypertext: A survey and introduction. *IEEE Computer 20*, 9 (Sept. 1987).
5. EVELAND, J., AND BIKSON, T. Evolving electronic communication networks: An empirical assessment. In *Proceedings of CSCW'86: MCC/ACM Conference on Computer-Supported Cooperative Work* (Austin, Tex., Dec. 3–5). MCC Software Technology Program, Austin, Tex., 1986, pp. 91–101.
6. HALASZ, F. Reflections on NoteCards: Seven issues for the next generation of hypermedia systems. *Commun. ACM 31*, 7 (July 1988), 836–852.
7. HALASZ, F., MORAN, T., AND TRIGG, R. NoteCards in a nutshell. In *Proceedings of ACM CHI'87: Human Factors in Computing Systems and Graphics Interface* (Toronto, Canada, Apr. 5–9). ACM, New York, pp. 45–52.
8. HERSHEY, W. Idea processors. *BYTE* (June 1985).
9. HORTON, M., AND ADAMS, R. How to read the network news. Distributed by Mr. Adams quarterly over the USENET news network, Center for Seismic Studies, Arlington, Va.
10. KUNZ, W., AND RITTEL, H. Issues as elements of information systems. Working Paper No. 131, Institute of Urban and Regional Development, Univ. of California, Berkeley, Calif., 1970. (See also Rittel, H., APIS: A Concept for an argumentative planning information system. Working Paper No. 324, Institute of Urban and Regional Development, Univ. of California, Berkeley, Calif., 1980.)
11. LOWE, D. G. Cooperative structuring of information: The representation of reasoning and debate. *Int. J. Man-Mach. Stud. 23* (1985).
12. MALONE, T., GRANT, K., LAI, K.-Y., RAO, R., AND ROSENBLITT, D. Semi-structured messages are surprisingly useful for computer-supported coordination. In *Proceedings of CSCW'86: MCC/ACM Conference on Computer-Supported Cooperative Work* (Austin, Tex., Dec. 3–5). MCC Software Technology Program, 1986, pp. 102–114.
13. NORMAN, D. A., AND DRAPER, S. W., Eds. *User Centered System Design*. Lawrence Erlbaum Associates, Hillsdale, N.J., 1986.
14. RUSSELL, D. M., MORAN, T., AND JORDAN, D. The instructional design environment. In *Intelligent Tutoring Systems: Lessons Learned*, J. Psotka, D. Massey Jr., S. Mutter, Eds., Lawrence Erlbaum Associates, Hillsdale, N.J., 1987.
15. SMITH, J. B., WEISS, S. F., FERGUSON, G. J., BOLTER, J. D., LANSMAN, M. L., AND BEARD, D. V. WE: A writing environment for professionals. Tech. Rep. 86-025, Dept. of Computer Science, Univ. of North Carolina at Chapel Hill, 1986.
16. TRIGG, R. H. A network-based approach to text handling for the online scientific community. Ph.D. dissertation, Univ. of Maryland, (University Microfilms #8429934), College Park, Md., 1983.
17. TRIGG, R., SUCHMAN, L., AND HALASZ, F. Supporting collaboration in NoteCards. In *Proceedings of CSCW '86: the Conference on Computer-Supported Cooperative Work* (Austin, Tex., Dec. 1986). MCC, Software Technology Program, Austin, Tx.
18. YANKELOVICH, N., HAAN, B., MEYROWITZ, N., AND DRUCKER, S. Intermedia: The concept and the construction of a seamless information environment. *IEEE Computer 21*, 1 (Jan. 1988).

Received June 1988; revised August 1988; accepted August 1988