Supporting Reflection-in-Action in the Janus Design Environment

Raymond McCall
College of Environmental Design
University of Colorado, Boulder

Gerhard Fischer and Anders Morch
Department of Computer Science
University of Colorado, Boulder

Introduction

We have developed a computer-based design aid called Janus, which is based on a model of computer-supported design which we think has significance for the future of architectural education. Janus utilizes a knowledge-based approach to link a graphic construction system to hypertext. This allows the computer to make useful comments on the solutions which students construct in a CAD-like environment. These comments contain information intended to make students think more carefully about what they are doing while they are doing it. In other words, Janus promotes what Donald Schon has called "reflection-in-action" [Schon, 1983].

The Janus design environment is named for the Roman god with a pair of faces looking in opposite directions. In our case the faces correspond to complementary design activities we call construction and argumentation. Construction is the activity of graphically creating the form of the solution—e.g., a building. Traditionally this has been done with tracing paper, pencils and pens. Argumentation is the activity of reasoning about the problem and its solution. This includes such things as considering what to do next, what alternative courses of action are available, and which course of action to choose. Argumentation is mostly verbal but partly graphical.

As an initial approximation, we can say that construction corresponds to what Schon calls action, while argumentation corresponds to what he calls reflection. Janus promotes reflection-in-action by providing computer support for argumentation about construction during construction. Janus integrates computer-support for both construction and argumentation, the former in the form of a graphic construction kit, the latter in the form of IBIS hypertext. It accomplishes this integration using a knowledge-based approach.

In this article we first describe precursors of Janus and then Janus itself. Finally, we explain the relevance of Janus to Schon's theory of architectural education.

The efforts to develop computer support for construction and for argumentation have proceeded in parallel with little or no interaction between them. The former is associated with CAD (computer-aided design), the latter with hypertext--in particular what is known as IBIS hypertext.

Work in CAD dates back more than twenty five years and is well-known in architectural circles. Almost invariably, the term CAD is construed to mean computer graphics, though not all graphics systems support the design activity we are calling construction. Many graphics systems support only drafting or rendering of already designed--i.e., constructed--forms. Nevertheless, there are systems, such as solids modeling systems, whose central purpose is to aid construction of complex objects, such as buildings.

Work in hypertext is nearly as old as that in CAD but until a few years ago was considered exotic and was pursued by only a small number of researchers [Conklin, 1987]. Within the last year there has been an explosion of interest in hypertext. This is largely due to Apple's HyperCard and to periodicals, such as Byte [October, 1988] and Communications of the ACM [September, 1988], which have devoted issues to hypertext. Even so, its past and potential future impact on computing are still not widely understood. Thus, for example, it seems that relatively few people realize that both the word processor and the mouse are spinoffs of hypertext research.

Work on software that developed into IBIS hypertext began about 1976 and until 1984 was pursued mostly in Europe. In this article we assume that the reader is familiar with CAD but not with the concepts of IBIS hypertext. We therefore start by explaining the IBIS approach underlying Janus and the hyperext technology which implements it.

Ibis Hypertext

In the mid-1970's advocates of the so-called "argumentative approach" to design methodology began development of computer support for design argumentation. By the early 1980's this resulted in hypertext based on the IBIS method.

The IBIS Design Method

Rittel's Original IBIS: In the late 1960's Rittel developed the notion that design problems were wicked problems [Rittel, 1972]. This meant they were intrinsically open-ended, situation specific and controversial in ways that defeated attempts to treat them like problems of science or mathematics. To deal with this wickedness Rittel called for an argumentative approach to design, an approach which acknowledged and promoted the judgmental, political and creative nature of design. The aim of the argumentative approach was to support the designer's reasoning without trying to automate it. To implement this approach Rittel developed the IBIS (Issue-Based Information Systems) method [Kunz, Rittel, 1970].

IBIS centers on the deliberation of issues arising in design, these issues being framed as questions. By deliberation we mean

1) identifying alternative answers to issues;
2) stating arguments for and/or against the proposed answers;
3) resolving the issues by selecting answers on the basis of the arguments.

In Rittel's IBIS issues are linked together by various relationships. These include an issue's being similar to, more general than, temporal successor to, and logical successor to (giving rise to) other issues.

The PHI Approach to IBIS: PHI (Procedural Hierarchy of Issues) [McCall, 1979, 1987] extends IBIS by broadening the scope of the concept issue and by altering the structure relating issues, answers and arguments. In Rittel's IBIS an issue is a question which is deliberated. In PHI every design question is counted as an issue, regardless of whether or not it is deliberated. PHI dispenses with the various inter-issue relationships of the original IBIS and uses instead only so-called serve relationships. These indicate that the resolution of one issue influences the resolution of another issue. Subissue of is the main serve relationship.

The overall structure of a PHI issue base, i.e., hypertext databases of issue discussion, is a quasi-hierarchy of issues with subissues, i.e., a tree-like structure such as that shown in Figure 1. PHI also allows and encourages the development of quasi-hierarchies of answers with subanswers and of arguments with subarguments.

The changes in scope and structure which PHI introduces to IBIS allow creation and effective use of far larger issue bases. They also increase the range of situations to which IBIS is applicable. In fact, there is a descriptive theory of design, called Issue-serve Systems [McCall, 1986], which predicts that PHI can model all the describable processes and information of design.

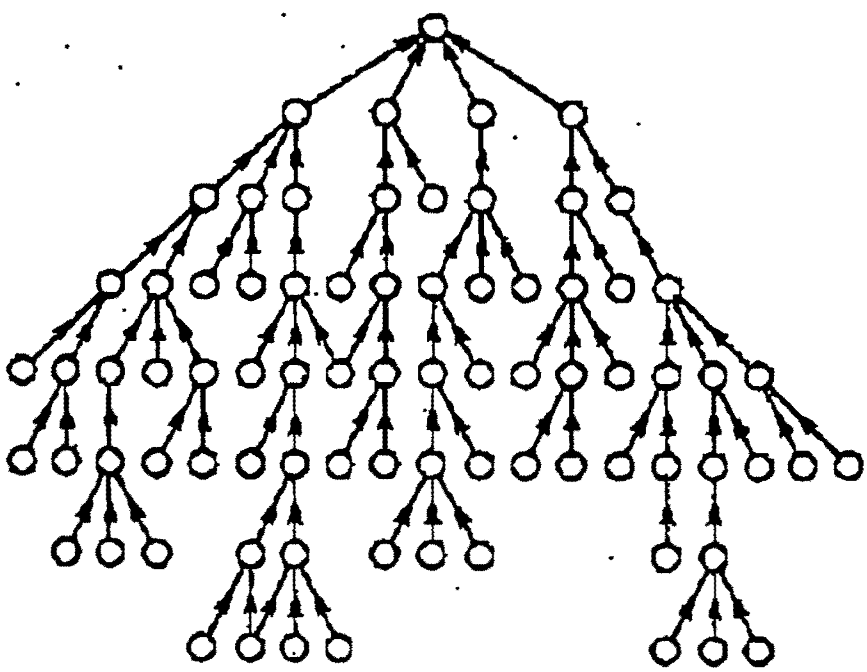

Figure 1: A quasi-hierarchy of issues with subissues.

Hypertext

Hypertext is software for managing non-linear structures of information [Conklin, 1987]. A linear structure is purely sequential, such as in a novel. A non-linear structure is a graph with labelled links. In this graph the nodes contain information—e.g., text—and the links correspond to relationships between the nodes. A common example of a non-linear structure of information would be a reference manual with many cross-references—e.g., "see also."

In many hypertext systems the nodes can also contain non-textual information, including graphics, animation, video, sounds and even executable code. Sometimes the term hypermedia is used to designate such systems; but the term hypertext is usually taken to include these as well.

Besides non-linear structure, the other defining characteristic of hypertext is navigation. Navigation means moving around in the hypertext graph by traversing its links. Usually this works as follows. The contents of a node–e.g. text–are displayed on the computer screen along with labels denoting its links to other nodes. Clicking on a link label with the mouse causes display of the contents of the node which the link points to. Repeated application of this link traveral allows the user to travel around in the information "hyperspace."

Previous Implementations of IBIS Hypertext

The MIKROPLIS System: The MIKROPLIS hypertext project began in 1980 as a means for implementing the PHI approach to IBIS. A prototype was finished in 1985 and has been in testing at the University of Colorado, Boulder since then.

MIKROPLIS [McCall, 1987] is a text-only system which superficially resembles an outline processor. Actually, it has full-blown database management capabilities and can handle graphs with labelled links and tens of thousands of nodes. At the nodes are texts of essentially arbitrary length. Clusters of nodes are displayed in outline format. Retrieval is by navigation and/or by use of an English-like applicative–i.e., functional–query language. This allows the retrieval of complex substructures of the hypertext issue graph using key terms which are numbers, indexed terms or substrings.

MIKROPLIS supports both exploration and creation of large and complex PHI issue bases. To date it has been used to construct issue bases on a range of subjects, including the design of housing projects, individual houses, neighborhood shopping areas, kitchens, and even health care policy. The largest of these is the equivalent of about 500 single-spaced pages in length but uses only a small fraction of the capacity of the MIKROPLIS system.

Rittel's System: Rittel himself developed an IBIS hypertext system on his Apple II in the early 1980's [Conklin, 1987]. This was a small-scale system which displayed only one text–e.g., one issue–at a time and had limited retrieval capabilities. It was nevertheless of great significance as the inspiration for the gIBIS system developed at MCC.

The gIBIS System: GIBIS, recently featured in an article in Byte Magazine [Begeman and Conklin, 1988], was developed by Jeff Conklin and Michael Begeman of the Software Technology Program at MCC (Microelectronic and Computer Technology Corporation). Based on Rittel's original IBIS, gIBIS has a sophisticated graphics interface and allows concurrent use of a single issue base through a local area network.

The ViewPoints System: ViewPoints is a hypertext application based on PHI and implemented in HyperCard on the MacIntosh computer. It was developed as part of the initial exploration of principles for Janus. It has fewer retrieval capabilities than MIKROPLIS but has graphic capabilities not available in MIKROPLIS. ViewPoints can be regarded as the direct conceptual precursor of the hypertext component of Janus. In particular, ViewPoints uses an issue base for kitchen design which is essentially the same as that used for Janus.

Evolution of the Janus Design Environment

Deficiencies of Separate Computer Support for Construction and Argumentation

Computer-supported construction facilitates design. The problem is that it facilitates both good and bad design. To create good design designers need knowledge about how to evaluate what is constructed, i.e., knowledge about criteria such as utility, safety, aesthetics, social value and cost. To increase the likelihood of good design a computer-aided design system should therefore supplement support for construction with a store of evaluative knowledge. But adding such knowledge to a construction support system would not in itself be enough. For, as Schon has argued, good design requires that designers have more than knowledge. It also requires that they reflect-in-action, i.e., think about what they doing while this thinking can still make a difference to what they do. Ideally, a computer-aided design system should promote and support such reflection-in-action.

Computer-supported argumentation, in the form of PHI hypertext, is a natural complement to a construction system. An issue base for the problem domain can contain much of the knowledge which students need. The PHI-IBIS method also provides a natural stimulus to and vehicle for reflection. It might therefore seem sufficient to provide the student with a stand-alone PHI system, such as MIKROPLIS or ViewPoints, to use during construction. Our experience, however, suggests that this strategy is ineffective.

With a stand-alone PHI system one has to interrupt construction and search for relevant argumentation in the issue base, just as one would if using a book. Our experiences suggest that, for a number of reasons, students will tend not to do this. One reason is that the interruptions are disruptive. A second is that searching for useful argumentation will not always pay off and students have no way of knowing when it will. A third is that students are often unaware that they need information and thus do not search for it.

Stand-alone construction and PHI hypertext systems leave a gap between action and reflection. Bridging this gap requires an integration of computer support for construction and argumentation. But how is this integration to be accomplished? An answer to this question was derived from the CRACK project.

CRACK: From Construction System to Design Environment

Student designers need systems which both support construction and contain knowledge for distinguishing good from bad in construction. We call such systems design environments [Fischer, Lemke 1987-1988]. CRACK (CRitiquing Approach to Cooperative Kitchen design) [Fischer, Morch 1988] is a design environment which is a direct precursor of Janus. It consists of two components: 1) a construction kit for kitchen floorplan layout and 2) a knowledge-based critic for evaluating these layouts.

A construction kit has a palette of parts and a work area for assembling these parts into complex designs. The parts are appropriate to a particular problem domain. In CRACK the domain is kitchen design and the palette contains kitchen equipment--such as sinks and stoves--and architectural fixtures--such as walls and windows. A set of operations on these parts are defined, such as move, rotate and scale. Users design by using the mouse to select parts from the palette and arranging them in the work area. This allows them to work directly in the problem domain, without having to type or to build objects from lines and simple shapes.

The knowledge-based critics in a design environment detect and criticize partially constructed solutions on the basis of knowledge of design principles. The critics in CRACK are state-driven condition-action rules which take action when non-satisfying partial designs are detected. The action they take is to display criticism based on principles of kitchen design derived from reference books and protocols of professional designers.

CRACK is knowledge-based but not an expert system. It aims to inform and support the judgment of designers not to "de-skill" them by judging or designing for them. Thus the CRACK user is free to ignore, turn off or even--within limits--to alter the criticism displayed.

From CRACK to JANUS

CRACK's criticism is really what we have been calling argumentation. In CRACK, however, this is not in PHI form nor is there any hypertext component. CRACK's argumentation also has a superficial "cookbook" character and does not show the complex argumentative background that an issue base can show. Nevertheless, CRACK demonstrates how to connect construction and argumentation using a knowledge-based approach.

The Janus design project began with the observation that the "critiquing approach" of CRACK could be used to connect a construction kit to a full-blown PHI hypertext system. In particular, CRACK's context-sensitive mechanism for triggering criticism could provide entry into precisely that point in the hypertext issue base where the relevant argumentation lies. The system could then display the argumentation relevant to the current construction situation without the user's having to search or even ask for it. This solves two problems: 1) how to improve the argumentation capabilities of CRACK and 2) how to make PHI inform construction.

Basically, Janus works as follows. The user constructs a kitchen layout with the construction kit. After each part from the palette is placed in the work area the system analyzes that placement. Brief criticism is displayed by the system if a design principle has been violated. Should the user want to see the argumentation underlying this criticism--e.g., to understand or challenge it--he or she can activate the hypertext system. Janus will then display the relevant section of the issue base. This display can be used as the starting point for further exploration of the issue base by navigation. When finished exploring, the user can return and complete the construction task.

Conceptually, Janus is an integration of two separate software systems: CRACK and ViewPoints. There are, however, differences in implementation details. In particular, the "ViewPoints" part is implemented in Concordia/Document Examiner, rather than HyperCard. The Document Examiner is a hypertext system developed by Symbolics Corporation for users to browse through its system documentation on line. Concordia is the authoring system which allows the user to create documents which can be browsed using the Document Examiner.

CRACK: Janus' Construction Face: The screen layout for CRACK, the construction face of Janus, is shown in Figure 2. The "Palette" in the upper left window contains the parts of the kitchen, called design units (DUs). These include both kitchen equipment--sinks, stoves, refrigerators, etc.--and architectural features--walls, windows and doors. The upper right window shows the "Work Area," where the actual construction takes place. A partially constructed kitchen is shown in this area. Construction is accomplished in the same manner as in CRACK: by using the mouse to select DUs from the "Palette" and position them in the "Work Area." Students can also reuse and redesign complete floorplans by selecting one of several examples from the "Catalog," which contains a varied collection of finished floorplans. After each placement--or repositioning--of a DU, critics analyze the layout and display any relevant criticism in the "Messages" window.

The Knowledge-based Connection between Construction and Argumentation: So far Janus functions like the original CRACK system. But the transition to the full-blown hypertext argumentation is made merely by clicking on one of the texts (criticisms) displayed in the "Messages" window. This provides entry into the hypertext at the relevant portion of the issue base.

ViewPoints: Janus' Argumentation Face: The screen layout for ViewPoints, the argumentation face of Janus, is shown in Figure 3. Different types of views of the PHI argumentation are shown in three windows. The "Viewer" window shows the actual argumentation in the relevant section of the issue base, including both text and graphics. Text is displayed in outline format and can include issues, answers and arguments. The "Outline" pane shows the identity of the issue being dealt with and the structure of its deliberation. "Visited Nodes" shows names of the sections of the issue base which have already been retrieved by the user. Each type of view provides areas which, when "clicked on" will cause the display of some other section of the argumentation, which in turn contains mouse-sensitive areas. This is "navigation," the hallmark of hypertext.

Computer Supported Reflection-in-Action

Schon defines reflection-in-action as thinking about how to act in a situation while the situation is at hand and action can still make a difference to it. We interpret each situation at hand as being associated with an issue which the designer is trying to resolve--for example, the issue "Where should the sink be located in the kitchen?" Janus "knows"--or rather assumes--that this issue is being addressed when the user selects a sink from the "Palette" and places it in the "Work Area." It presents relevant argumentation while this can still make a difference to the decision taken on the issue.

Janus promotes two kinds of learning by doing: learning design principles and learning to reflect-in-action. It also promotes two kinds of reflection-in-action: reflection triggered by violation of principles of design and reflection on the principles of design themselves.

Reflection Triggered by Violations of Principles of Design

Schon sees good design as "a reflective conversation with the situation," in which the designer acts and the situation "talks back." A goal of studio education is to get students to engage in such reflective conversations. Students have difficulty doing so because they cannot "hear" what the situation is "saying." In other words, they do not see the unintended consequences of their construction actions. Schon points out that such unintended consequences--pleasant or unpleasant--are the crucial stimulus to reflection. Unfortunately, for neophyte designers the problem situation does not speak for itself; it needs a spokesman, i.e., a critic. When available, studio teachers play that role and thus lead students to reflect. Janus supplements the work of these teachers with knowledge-based critics and does so while decisions about construction are actually being made--something which teachers often cannot do. By informing students when principles of kitchen design have been violated it prompts them to re-think their designs.

Reflection on principles of Design

Schon stresses that design is far more than the application of standard "principles." It is also professional artistry: the ability to reflect on situations which are conflictual, uncertain and/or unique. In these "intermediate zones" principles do not suffice. The assumptions underlying them may be violated, the "experts" may disagree, or there may be no principles at all. In such cases which are not "in the book" or for which their is no consensus the designer must reflect carefully on how to act.

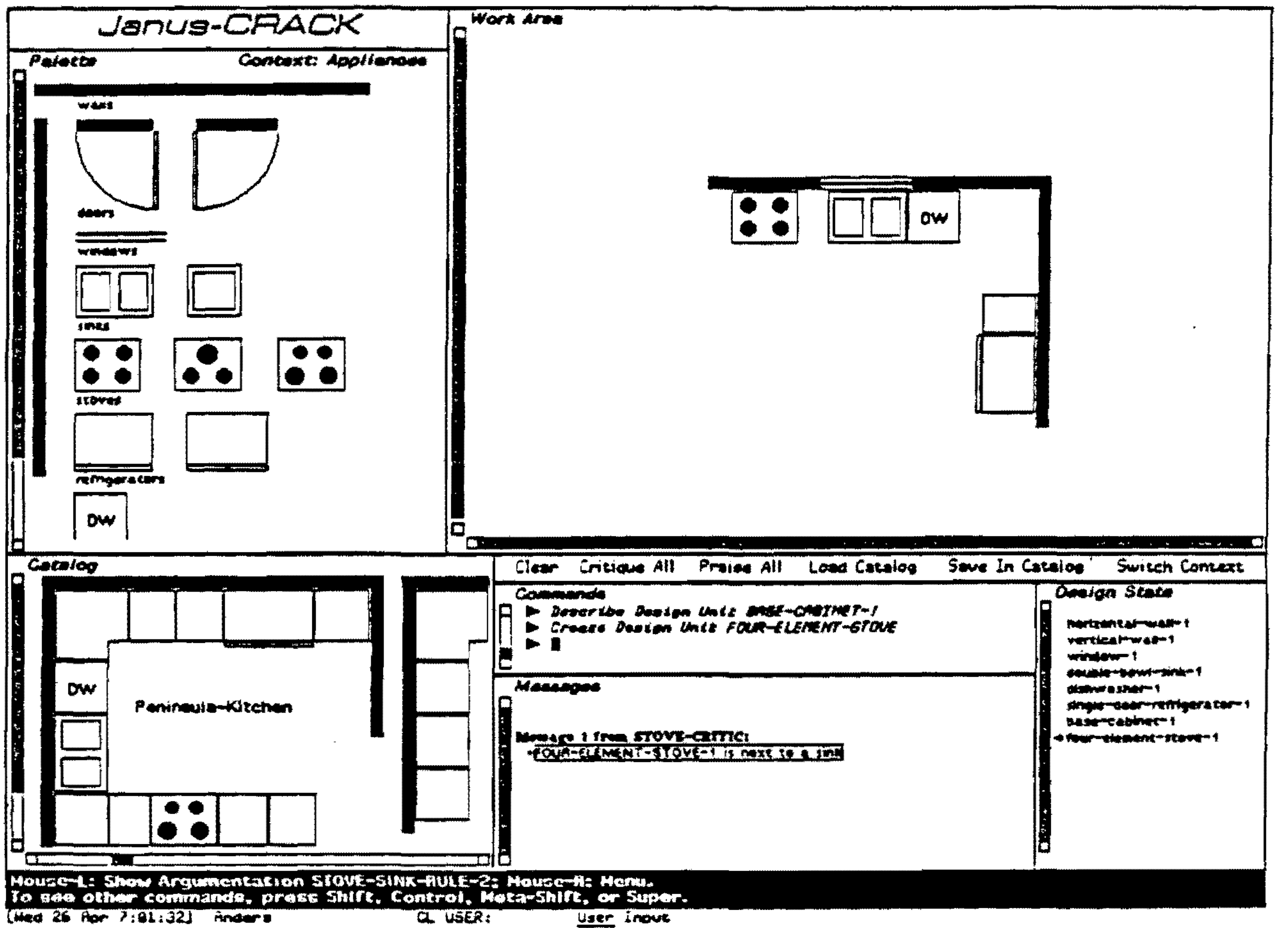

Figure 2: CRACK: Janus' Construction Face.

Construction in Janus-CRACK is based on a direct manipulation interaction style, i.e., using the mouse, although commands can also be entered by keyboard in the "Commands" pane. Building blocks (Design Units) are selected from the "Palette" and can be moved around with the mouse to desired locations inside the "Work Area." Students can also reuse and redesign complete floorplans by selecting one of several examples from the "Catalog." The "Messages" pane displays criticism by the knowledge-based system. Clicking on a criticism with the mouse will bring the student into the ViewPoints hypertext (See Figure 3.).

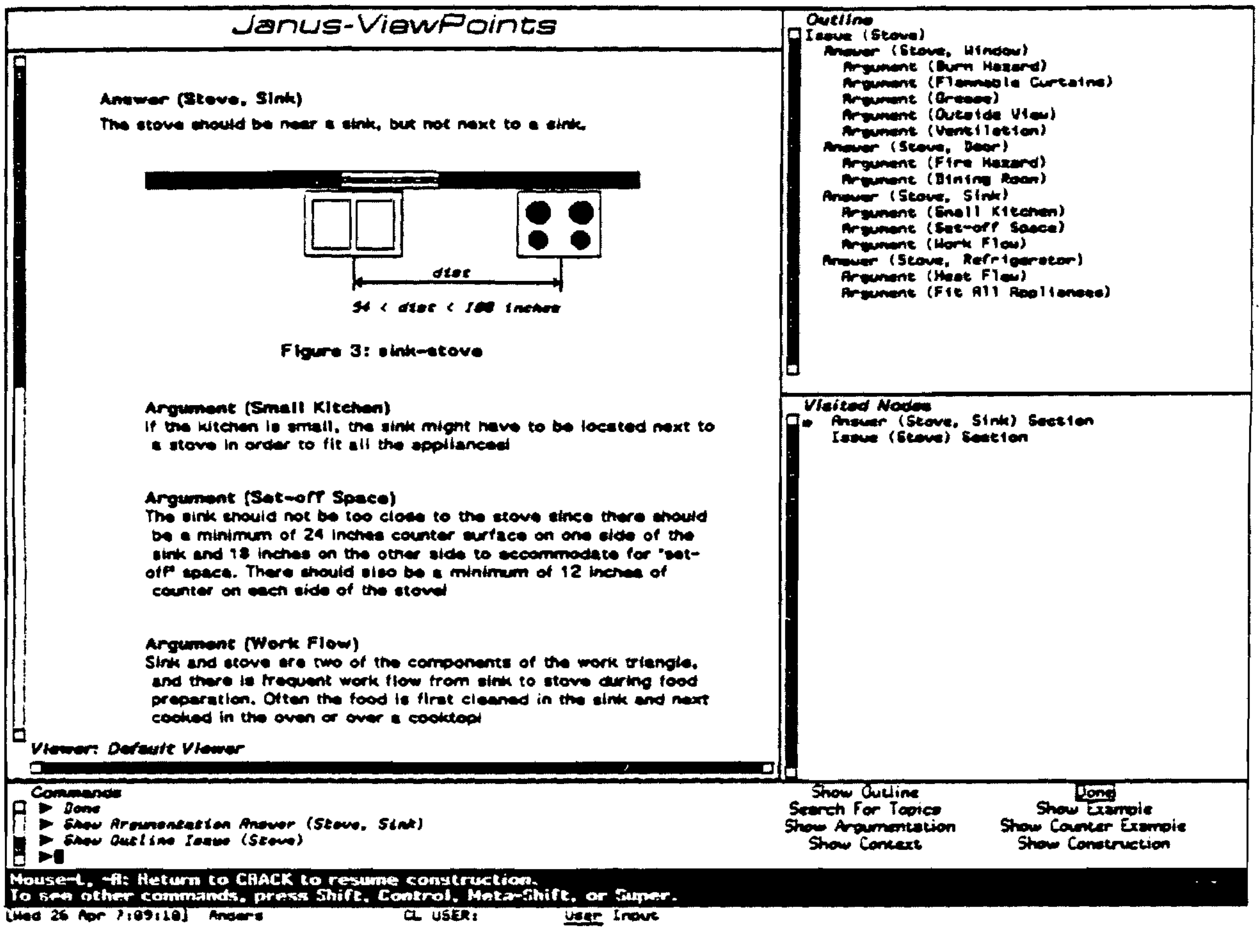

Figure 3: ViewPoints: Janus' Argumentation Face.

Janus' argumentation component is a PHI-IBIS hypertext system implemented using the Symbolics Document Examiner. When students click with the mouse on criticism in the "Messages" pane in CRACK, the construction face of Janus (See. Figure 2.), they are brought into ViewPoints at the point where the relevant argumentation is to be found. T.e "Viewer" pane shows argumentation on an answer suggested by the criticism. (Note that this pane can be scrolled.) The "Outline" pane shows the identity of the issue being dealt with and the structure of its deliberation. "Visited Nodes" shows names of the sections of the issue base which have already been retrieved by the user. In this example, the issue is "Where should the stove be located?" The student has evoked criticism from the knowledge-based system by placing the stove near the sink. The ViewPoints system shows the user various pros and cons of placing the stove near the sink.

Few of the standard "principles of design" are inviolable. They are merely rules of thumb whose appropriateness must be judged by the designer in each new situation. In other words, the designer must not merely reflect on how to apply principles but also on whether the principles should be applied as-is, modified for the particular situation, or simply abandoned.

IBIS is a natural stimulus to and vehicle for such reflection. It was designed to deal with the controversiality, open-endedness and essential uniqueness of Wicked Problems. This fits well with Schon's characterization of the "intermediate zones" of design as "conflictual, uncertain and unique."

An issue base can supply a variety of information to stimulate and improve reflection. This includes not only design principles but also issues, alternative answers and arguments which help the student to decide whether standard principles are applicable to a situation at hand. Where exceptions must be made to such principles, the argumentation underlying them can provide a basis for reasoning about how to deviate from them.

Janus can also display useful information for situations where no principles exist. This information includes issues to consider, possible answers as well as relevant criteria and other arguments. All of these require students to judge for themselves. They inform student judgment but do not judge for the students.

Conclusion

There is much that can and should be done to extend the work described above. We need to expand the current issue base on kitchen design to include more social, psychological, aesthetic and monetary criteria. We need to develop issue bases in other problem domains so that Janus can be applied to these domains. While kitchen design has been a useful starting point, the significance of the Janus approach will lie in its applicability to a wide range of problems.

In working with Janus we have discovered several additional ways of connecting the construction and argumentation. Implementing these will further promote reflection-in-action and extend the range of problem domains to which Janus is applicable.

Much can be done to make Janus a better aid to the designer's ability to understand the consequences of construction decisions. Inclusion of color, 3-D graphics and multiple views are perhaps the most important. Computer-based analysis–e.g., of lighting–as well as simulation and gaming of use situations would also be useful.

These and other considerations indicate to us that exploration of the Janus approach has only begun. The Roman diety, Janus, was the god of gateways. It is our hope that the Janus design environment is a gateway into a new area of research on computer-aided reflection-in-action.

REFERENCES

Begeman, M.; Conklin, J., "The Right Tool for the Job," Byte, September, 1988.

Conklin, J. "Hypertext: An Introduction and Survey," IEEE Computer, Vol. 20, No. 9, September, 19

Fischer, G.; Lemke, A. "Construction Kits and Design Environments: Steps toward Human Problem Domain Communication," in Human-Computer Interaction Journal, Vol. 3, No. Lawrence Erlbaum, 1987-1988, pp. 179-222.

Fischer, G.; Morch, A. "CRACK: A Critiquing Approach to Cooperative Kitchen Design," Proceeding the International Conference on Intelligent Tutoring Systems, June 1988.

Kunz, W.; Rittel, H. "Issues as Elements of Information Systems," Working Paper 131, Institut fuer Grundlagen der Plannung, Stuttgart, 1970.

McCall, R. On the Structure and Use of Issue Systems in Design, Dissertation, University of Californi Berkeley, University Microfilms, 1979.

McCall, R. "Issue-Serve Systems: A Descriptive Theory of Design," Design Methods and Theories, Vc 20, No. 3, 1986.

McCall, R. "PHIBIS: Procedurally Hierarchical Issue-Based Information Systems," Proceedings of the Conference on Architecture at the International Congress on Planning and Desigr Theory, American Sociey of Mechanical Engineers, New York, 1987.

Rittel, "On the Planning Crisis: Systems Approaches of the First and Second Generation," Bedriftokonomen, 8:390, 1972.

Schon, D. The Reflective Practitioner, Basic Books, 1983.