A Web-centric CAD System for Collaborative Design

Raymond McCall
University of Colorado, Denver

Key words: net-centric, Web-centric, hypertext, hypermedia, CAD, HyperCAD, knowledge-based critics, collaboration, executable content, argumentative design, IBIS, PHI, design rationale

Abstract: Web-PHIDIAS is a hypermedia-based, intelligent CAD system that delivers both CAD functionality and design information to anyone with Web access. This system is above all designed to facilitate collaborative architectural design. It provides both private (individual) and collaborative (group) drawing and text authoring spaces, with a variety of types of authoring and viewing privileges for groups. This enables a single designer to work in privacy on one piece of a design and later "publish" it to a supervisor or a group. It also enables a group to work in privacy and later publish its work to a different or larger group. This notion of "levels of privacy/publication" is a crucial but too-often missing component of collaborative design systems. With Web-PHIDIAS, all drawings are stored in a central repository accessible from the group server. This means that they are accessible from anywhere in the world to any viewer who has the required viewing and/or authoring privileges. This enables designers to access and modify stored drawings while travelling or when out on the site, even if the site is in another country. It also enables them to create new drawings and store them in the central repository from anywhere in the world. Web-PHIDIAS consists of an interactive Web-based client that serves as an interface to the PHIDIAS hypermedia server. This client, which is implemented in Java, provides basic, 2D graphical editing functionality and as well as display of 3D views. It also provides access to multimedia information useful for whatever design task is at hand. This information includes text and graphical descriptions of design precedents as well as various issues in design of a particular type of building.

1. INTRODUCTION

1.1 Goals of our work

Our work has for more than two decades been aimed at the creation of hypertext systems that support an argumentative approach to design [Rittel 1972]. This approach treats difference of opinion as a positive, democratic force that can improve the quality of things designed by helping designers to explore a wider range of design issues, solution options and evaluation criteria. The argumentative approach is especially well suited to group design processes, both in the sense of collaboration in design teams and participation by clients and users in design projects.

To support the argumentative approach, Rittel originally proposed IBIS (Issue-Based Information Systems) [Kunz and Rittel 1970], a method-not a computer system-for eliciting and organizing design rationale. Our systems support PHI (Procedural Hierarchy of Issues), a method that attempts to improve on IBIS by making it more general and more scalable [McCall 1991]. PHI, like IBIS, organizes information around the deliberation of design questions, referred to as *issues*. Various *answers* are proposed to these issues, and *arguments* are then given on the answers and on other arguments. Where PHI differs from IBIS is above all in the way issues are linked together. While IBIS used a variety of inter-issue relationships, PHI uses only relationships that indicate ways in which the answering of one issue depends on the answers given to other issues. By concentrating on these *dependency relationships* between issues, PHI creates more orderly networks of issue-based discussion. This in turn enables the creation and management of issue networks that can easily be ten to a hundred times the size of typical IBIS networks.

We have sought to create software systems that support the argumentative approach. Our systems have included a range of technologies, including hypermedia, CAD graphics and knowledge-based computation; but they have not included the sort of support for collaborative design that the argumentative approach calls for. Until recently, creating such support was limited by the limited nature of networking technology for computing. In past few years we did develop a multi-user prototype that worked in local-area networks [McCall Argumentative Agents article in Construction Automation]. But this system did not support the work of highly distributed teams or participation-at-a-distance by users or clients.

In recent years the explosive growth of Internet technologies has dramatically changed the potential for computer supported collaboration. In particular, the World Wide Web, Java, and other tools for creating Web-based client-server applications enable us to support group design in a far more full-blown manner. This article describes the first prototypes of Web-PHIDIAS, a software system that uses such tools to support collaborative design. We begin by briefly describing our prior work on hypermedia-based CAD. We then describe four stages in which we modified our system to support the delivery of design information and group design over the Web. This includes a description of a computational mechanism that we have devised to support group work by providing different contexts for collaborative communication. Finally we describe our plans for future prototypes.

1.2 Our previous prototypes

1.2.1 Our first hypertext systems

Our work began with the creation of single-user, text-only hypertext systems to support argumentative discussion in design, using PHI as a more comprehensive and scalable version of IBIS. Our first systems, PROTOCOL [McCall 1979] and MIKROPLIS [McCall, Mistrik and Schuler 1983; McCall 1989] were intended to support the creation of project-specific databases of issue-based design rationale-or *issuebases*-from scratch. MIKROPLIS, in particular, was designed to eliminate most of the extensive and tedious "secretarial" work of recording and updating design rationale in an organized and retrievable fashion. Up to that time, it appeared that this "secretarial" work was so great that it was preventing effective implementation of both IBIS and PHI.

1.2.2 Domain-oriented issue bases

Extensive experience with use of MIKROPLIS by end users revealed that reducing the secretarial work of issue-based documentation was not enough to make it practical to develop project issuebases from scratch. MIKROPLIS did make it possible to develop issuebases many times the size of pervious issuebases; nevertheless there still remained far too much conceptual and editorial work to justify the creation of issuebases for all but the most important projects.

We therefore adopted the strategy of constructing *domain-oriented issuebases*, i.e., collections of issues, answers and arguments that recur frequently in a given problem domain-for example, the design of a particular type of building. Here, the enormous work of creating an issuebase could be justified, because a domain-oriented issuebases could be used in many projects-potentially all the projects in a given domain. Such an issuebase can 'jump start" the creation of project-specific issuebases by enabling designers to create the rationale for a given project simply by adding comments and judgments to the generalized issuebase for the domain-a process we call *differential reasoning*. Thus, for example, a generalized issuebase for the domain of house design could easily be tailored to create an issuebase for design of a particular house on a particular site.

Domain-oriented issuebases have another advantage. They can provide designers with specialized information for the problem domain. They do this by serving as a sort of souped-up FAQ (Frequently Asked Questions) that provides useful information about a specialized domain. Over the years we have built extensive issuebases for a number of domains, including kitchen design, house design, layout of computer networks in buildings, and NASA's design of space-based habitats-plus one very large issuebase on health care policy.

1.2.3 Integrating CAD graphics: JANUS

By 1984 it had become clear that our text-based information systems would not be sufficient for architectural design. Support was also needed for graphical design and for ways to provide architects with information as they designed with graphics. Our next system, JANUS [McCall, Fischer and Morch CAADF 1989] presented the designer with two different faces (thus the name JANUS), one for CAD-graphic form development and the other for argumentative reflection on design decisions. To connect the graphical and argumentative aspects of the system, JANUS used knowledge-based critics that critiqued partially developed graphic solutions using rationale drawn from a domain-oriented issuebase. Schoen's theory of "reflection-in-action" [Schoen 1983] provided a useful theoretical framework for JANUS and all our subsequent systems. This theory sees design as a continual alternation between two types of thinking, which Schoen refers to as *knowing-in-action* and *reflection-in-action*. We interpreted these respectively as form making (using CAD graphics) and *argumentative deliberation* (using PHI-based hypermedia).

1.2.4 Intelligent HyperCAD

JANUS employed a heterogeneous system architecture, with three, loosely coupled major subsystems: a CAD system, a hypertext system and a knowledge-based system. Added on to these were a number of minor subsystems. As we added new functionality to JANUS, it tended to "grow like Topsy," becoming progressively more complex and, thus, more difficult to debug and extend. We therefore devised a simpler and more integral system architecture that we dubbed *Intelligent HyperCAD*. This offered the same basic threefold functionality of JANUS-i.e., CAD graphics, hypermedia browsing and knowledge-based computation-but did so using a common, computational substrate of fine-grained hypermedia. This architecture became the basis of PHIDIAS [McCall, Bennett and Johnson 1994]. In this system, all data, information and knowledge were represented as hyper-nodes and hyper-links. Complex graphical objects were represented as composite hyper-nodes, and knowledge-based computation was done by using the hypermedia network as a semantic network. A typical screen from PHIDIAS is shown in Figure 1.

Because of its integral architecture, PHIDIAS proved far easier to extend than JANUS. In particular, it became much easier to inter-relate and combine the major types of functionality to create new functionality. This greatly facilitated our efforts to create the context mechanism, described below, that our latest prototype uses to support collaborative communication.

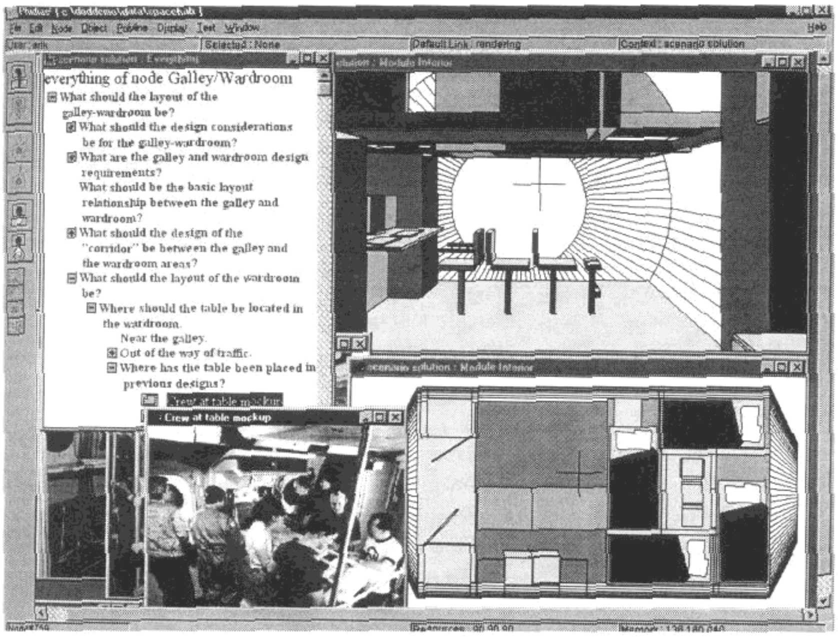

Figure 1. This screen of the stand-alone version of PHIDIAS shows its capabilities for displaying text, raster images and vector graphics. Text and vector graphics could also be created and edited with the system. Additional capabilities of PHIDIAS included knowledge-based critics, voice annotation and video display.

2. WEB-PHIDIAS

2.1 Significance of the Web for our project

The growth of the Internet and, in particular, of the World Wide Web is enormous significance for our work. For one thing, the "gospel" of hypermedia that we and others had been preaching for years is now widely understood and accepted. As the Web first began to take shape, hypertext pioneer Ted Nelson-who come the terms hypertext and hypermedia in the 1960s-was, as he put it, "promoted from 'mad man' to 'visionary." By the beginning of 1995 it was already clear that his then 30-year-old notion of a world-wide network of hyper-linked information on computers was not merely a visionary's futuristic dream; it was an explosively emerging reality. At the end of the century we find that hypermedia, in the form of the World Wide Web, has become a major cultural and economic force. The influence of the Web will continue to grow over the first decade of the 21st Century..

The success the Web made it easy for us (for the first time) to persuade people of the value of the hypermedia concepts on which our prototypes were based. At the same time the Web's success also caused many people to question the need for a system like PHIDIAS that has its own approach to hypermedia-an approach that differs from the Web in a number of ways. "Were we seriously proposing to compete with the Web?" they asked. Some argued that while the Web proved our point about hypermedia, it also effectively put us out of business.

For us, the issue was never how to compete with the Web, but rather how to use it-and other aspects of the Internet-to give a boost to our efforts to implement the argumentative approach. To understand how the Web helps us achieve our goals it is useful to look at what networked computing is likely to offer in the way of general future functionality.

We subscribe to the widely repeated prediction that networked computing will ultimately offer us *anything, anytime, anywhere* (AAA), where "anything" should be understood to mean 1) any information, 2) any communication, and 3) any computation. Let us look at what each of these means and then relate this to the goals of the argumentative approach.

2.1.1 Any information

This means that whatever information we need we will be able to find it whenever we want and from wherever we happen to be. Access to a vast amount of high-quality information has emerged as one of the first and most obvious things that the Web offers us. The bandwidth limitations we now experience will disappear over the next decade. The quantity and quality of information available on the Web will increase greatly. It is not inconceivable that all documented human knowledge will be accessible online within a decade.

Significance for the PHIDIAS project: A goal of the argumentative approach is to improve design-e.g., improve the quality of buildings-by providing designers with useful information. To do this, PHIDIAS could use the Web/Internet as a vehicle for distributing information. Secondly, PHIDIAS could be use information that is on the Web to augment its own information. Among other things, PHIDIAS can serve as a means for finding and organizing the design-related information that is distributed on the Web. Perhaps the hallmark of information on the Web is that is comes from a plurality of points of view; so the fit between PHIDIAS and the Web is in this respect very good.

2.1.2 Any communication

This means that we will be able to communicate with any person we want to, whenever we want to and from wherever we want. Internet-based communications is headed in this direction with email, newsgroups, interactive "chat", instant messaging, and so forth. As wireless communication-including satellite communication-and various portable computing devices proliferate over the next decade, it will become increasingly possible to communicate with any person in any location and at any time-at least asynchronously. Communication for collaborative work will be ubiquitous-so much so that the notion of *groupware* will likely fade away as all software becomes group-aware, i.e., comes with collaborative communication capabilities built in.

Significance for the PHIDIAS project: Collaborative design communication is the central idea of the argumentative approach. Documenting and managing such communication are what IBIS and PHI were created for. Documenting design rationale has proved very difficult to do in the past, because it required extra work from designers. But it can happen easily in the future by being made a side-effect of routing design communication through computers; once the text, voice, drawings and video used for computer-mediated communication are in digital form, we only need to keep persistent records of them. PHIDIAS can use this fact to better capture and distribute design rationale. Collaborative design using networked computers is going to create highly detailed records of design thinking for the first time in history. We will need systems like PHIDIAS to manage the deluge of data that will inevitably result.

2.1.3 Any computation

This has two related meanings: one for hardware and one for software. On the hardware side, it means that whatever hardware resources we need to do computation will be available to us wherever and whenever we need them. The computer that we use to access the Internet will not be the only one we use for computation. We already see this trend in the movement toward Web-based client-server systems. These appeared first in the form of database systems that manage large Web sites. Next came "personalized" Web sites with selected news, email and calendar functionality-all with content tailored to the individual users and having "pages" that are computed "on the fly" at display time. A host of new technologies is waiting in the wings to give users additional access to the computational capabilities of other computers, including CORBA, RMI, ASP, JSP, and servlets. In addition, currently experimental uses of Internet-networked computers as distributed supercomputers could also accelerate dramatically-turning Sun's motto, "The network is the computer" into a reality.

On the software side, "any computation" mostly refers to *executable content*, i.e., content of Web pages that is not merely text and pictures, but is interactive software. Whether implemented using Java, ActiveX, Dynamic HTML, or some as-yet-unimagined technology, the trend toward Web pages whose content executes on the user's (client side) computer seems certain to accelerate. Limitations of current computers and network technologies now impede use of executable content, but the advances of the next decade will change this situation dramatically. Ultimately users will have *software on demand*; any type of software the user might want would be available on the World Wide Web for immediate use without any complex installation process. "Software" might well become a type of *service* rather than a type of *product*. The notion of going to a store to buy software in a bubble-pack and then struggling to install it on a hard disk might in the near future be considered a bizarre ritual from the Stone Ages of computing.

In summary, nearly any type of software and hardware capabilities that any user could need will become available for use over the Internet. The limitations of the desktop computer and its installed software will in no way limit the computation that users can do.

Significance for the PHIDIAS project: The argumentative approach is most easily and strongly supported when collaborative design is supported. The best way to enable collaboration in design by a wide range of people is to eliminate distance as a barrier to participation. This can be done by providing an interactive interface to a collaborative design system that is available over the Internet-e.g., as executable content in a Web page. Such an interface must provide communication with other participants in the design project-regardless of their number or location; and this can best be done by means of a central server that acts as a switchboard for all communications. Using a "thin client" strategy relieves the client computers of computational burdens and therefore increases the types-and thus the number-of Internet devices that can be used as means for participation in design. This server-based, thin-client strategy exploits both the hardware and software aspects of "any computation."

Using the server-and-thin-client strategy, the PHIDIAS functionality-including CAD graphic editing and display, knowledge-based critiquing, information retrieval (with task-based indexing)-can be made available to every desirable participant in a given design project-from anywhere in the world they happen to be, from a wide range of Internet-aware devices, and at any time of night or day that they wish to participate. It will possible to use a library computer, a Web-enabled television or even a PDA to access project drawings, modify them, store them in the central PHIDIAS database and send them to any of the members of the design team or any clients or users. Collaborative design will be possible anytime and anywhere.

2.2 Four stages of implementing Web-PHIDIAS

2.2.1 Stage 1: Putting NASA's MSIS online using Web-PHIDIAS

Our first opportunity for exploring the transition to a Web-based version of PHIDIAS came from our work with NASA. Two years ago, people at the Flight Crew Support Division of the Johnson Space Center became convinced that they needed to put their guidelines and requirements for design of space-based habitats on a Web site, where they would be available to NASA contractors and others who might be interested. These guidelines and requirements are collected in a document known as the Man-Systems Integration Standards (MSIS) and currently take up somewhat less than a thousand pages of paper per version, with a separate version of being created for each new habitat. Thus, the generalized MSIS version is labelled NASA STD-3000, while the specific version for the International Space Station is labelled SSP 50005.

The MSIS was originally created and is still maintained using a desktop publishing program. Since this software can automatically produce an HTML version of any document it creates, it might seem to be a simple task to produce a reasonable Web site containing the MSIS. In fact, for a variety of technical reasons, the NASA staff found it to be impossible. Since no commercial software seemed to offer the needed functionality, the PHIDIAS team-i.e., the author and several of his former students-was therefore hired to create a Web-based MSIS using PHIDIAS. We succeeded by using an approach of first translating the MSIS data PHIDIAS proprietary format and then using PHIDIAS as a hypermedia-database engine that created Web pages "on the fly." Creating a manageable and malleable Web site for long, hierarchically constructed documents such as the MSIS is no simple task, but it is one for which PHIDIAS is especially well suited. The HTML pages produced by PHIDIAS are simple, but the task of managing MSIS information, which is hyper-linked both hierarchically and cross-referentially, is not. Storing information as separate Web pages is not a good approach when that information is updated frequently. The PHIDIAS approach stores information on the level of the paragraph. Figure 2 shows a screen image of the Web-PHIDIAS version of the MSIS.

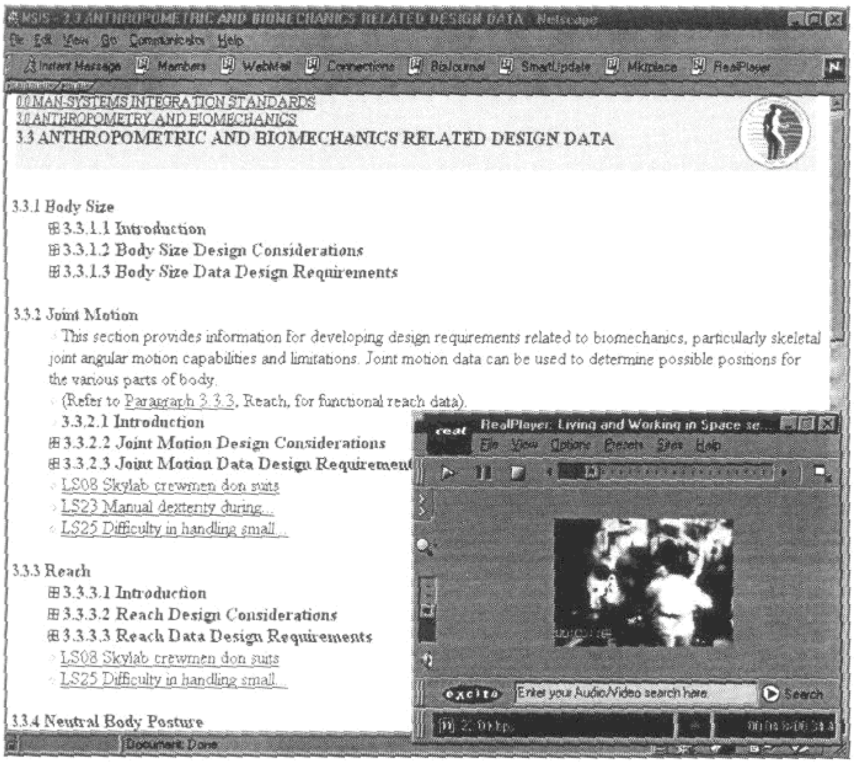

Figure 2. A page from the Web-based MSIS is shown here with one of its many streaming video clips. The Web pages are generated on demand by Web-PHIDIAS from its internal model of the MSIS document. The "plus buttons" represent hierarchical links. The underlined "Paragraph 3.3.3" is a cross-reference link. The remaining links are video clips.

For the MSIS project, we ignored the PHIDIAS user interface and instead used the PHIDIAS hypermedia database engine as a system for managing hyper-linked information. Every database management system uses some language for retrieving data. We used LINQ (Language for Inference, Navigation and Query), the functional (i.e., applicative) hypermedia language that has been the basis of our prototypes since the early 1980s [McCall et al. 1990]. What the Web-MSIS project did was enable us to exploit the inherently language-based approach that we have been using for many years to create a client-server system that would function over the Web. In particular, clicking on a link displayed on a Web-MSIS page sends a LINQ query to the PHIDIAS database system, which then retrieves the data specified in the LINQ query and displays it using a specified HTML page-template. The irony here is that our decision in the early 1980s to use a language-based approach to hypermedia became the basis for creating the Web-based functionality of our current prototypes.

2.2.2 Stage 2: Presenting and critiquing design proposals online with Web-PHIDIAS

In the next stage of our work we moved a significant step closer to our goal of creating a fully functional Web-based HyperCAD system. In this stage we created a system for online presentation and critique of design proposals. This new version of Web-PHIDIAS enabled designers to present examples of their design work-in the form of VRML models-over the Web together with the rationale for their design choices. Each VRML model was represented as a node in the PHIDIAS system. Rationale was stored in linked collections of nodes-as in the stand-alone version of PHIDIAS. Rationale for a design proposal was presented in a Java interface that enabled users at remote locations to call up the rationale and then make comments in response to the design and/or the rationale. These comments were then immediately added to the PHIDIAS database of rationale and posted to the Web, where other users-including the original designer-could respond to the comments. A screen image of this system is shown in Figure 3.

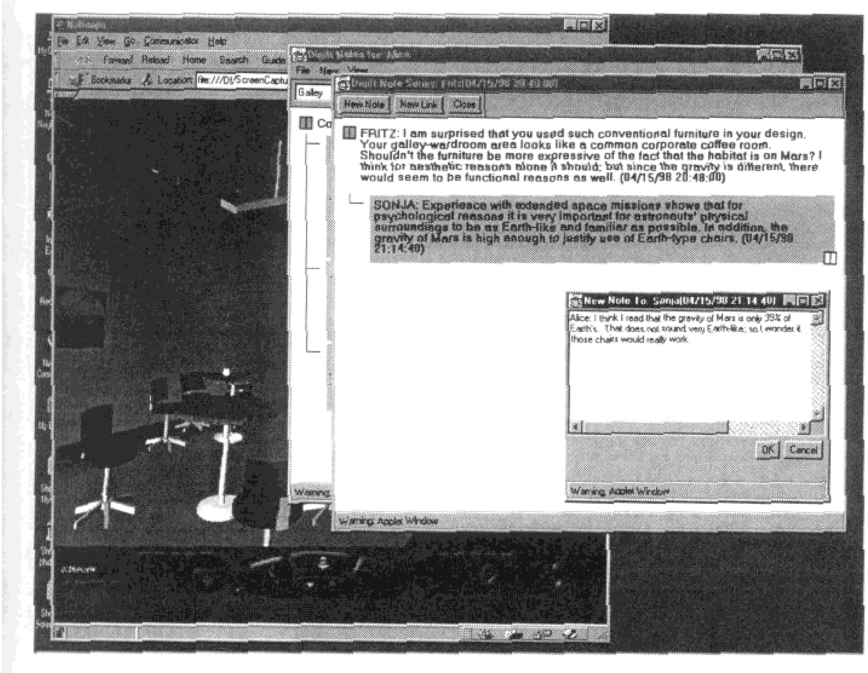

Figure 3. Here we see that a student (Sonja Holmes) has created a VRML model of her design for a Mars Habitat and then displayed this in a Web page created by Web-PHIDIAS. Fritz, a visitor to the Web site, has made a comment on Sonja's design and she has responded. A second person, named Alice, is shown here using the authoring interface to create another comment on Sonja's design. The interface for display and authoring of comments is written in Java. Once entered, comments are stored in the PHIDIAS database, which then displays them on the Web.

2.2.3 Stage 3: Collaborative design online with Web-PHIDIAS

The third stage in creating the Web-PHIDIAS system was to create a Java interface that enabled users to utilize more of the functionality of the server-side PHIDIAS system. In particular, we created 1) an interface for the display of rationale in an expandable! collapsible outline format. 2) an interface for display and editing of 2D representations of 3D forms in plan, elevation and section, 3) 3D perspective representation of 3D forms, and 4) interfaces for the display of additional media types, such as raster images and streaming video. A screen image of this first real HyperCAD prototype Interface is shown in Figure 4.

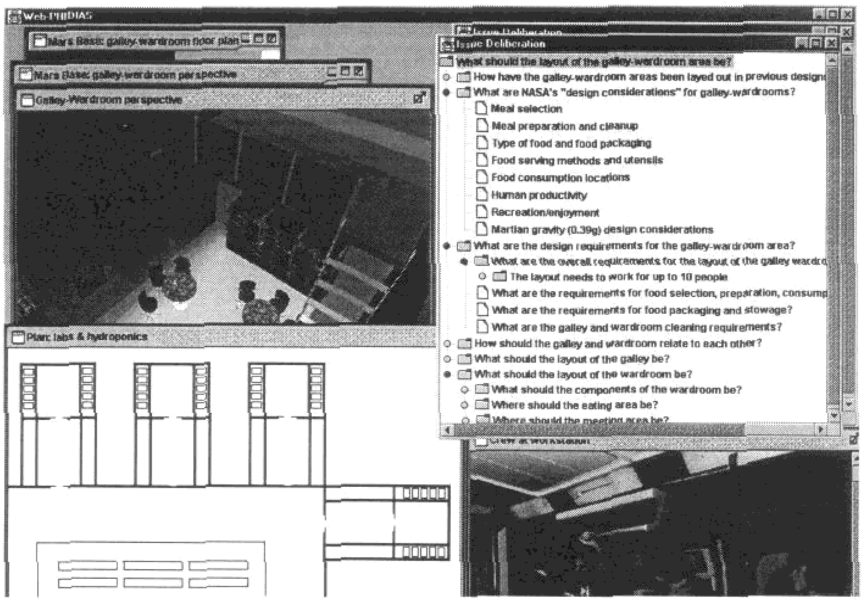

Figure 4. The Web-PHIDIAS Java interface enables use of the inherent functionality of the PHIDIAS hypermedia database. This functionality includes capabilities for display and editing of vector graphics and hierarchically structured design rationale as well as display of a variety of media types including raster images and streaming video.

2.2.4 Stage 4: Supporting collaboration with virtual copies of hypermedia networks

The object-oriented functionality of PHIDIAS provides inheritance of nodes, links and even entire hypermedia networks. This has many uses, but the most important for this paper is the support it provides for collaboration. In particular, it is the functionality for inheritance of hypermedia networks that provides the main means for managing collaborative communication in Web-PHIDIAS.

In Web-PHIDIAS, one hypermedia network inherits from another by creating a *virtual copy* of the network inherited from. To understand what a virtual copy of a network is, imagine a sheet of clear acetate laid over a graph representing the node-link structure of a hypermedia network. We could create a "new" network by drawing new links and nodes on the acetate and by "whiting out" old links and nodes on the acetate. To the casual observer it might appear that we had modified the original graphic image. But picking up the sheet of acetate reveals that we have merely created on the acetate a description of the differences between the original network and the "new" network. When the acetate containing this *differential description* is overlaid on the original image, the result is called a *virtual copy.* With PHIDIAS we say that the original network and the virtual copies of that network are each in a different "hypercontext." When it is used to support collaborative communication we call each hypercontext a collaboration context. (Though this functionality and its implementation strategy are unique to our system, they elaborate basic ideas about virtual copying of non-hypermedia networks in [Mittal, Bobrow and Kahn 1986]. Our own approach to virtual copying is described in [McCall 1993].)

The crucial point for understanding how hypercontexts - including virtual copies of hypermedia networks - can function as collaboration contexts is to see that each hypercontext is a view of the underlying hypermedia database. In any given hypercontext we see some nodes and links but not others. Thus, a hypercontext both enables and restricts our viewing of the database. From this, we can see that it is trivial to associate viewing privileges with hypercontexts: all we have to do is password-protect each hypercontext. It should also be clear that it is a simple matter to give more than one user viewing privileges for any given hypercontext and to give each user viewing privileges in more than one hypercontext.

We can in a similar manner assign various levels of authoring privileges to specific users in specific hypercontexts. (Here by "authoring" we are referring to the creation of both texts and CAD drawings.) The persons authorized to view a drawing or text can be assigned various levels of authoring privileges. They might have no authoring privileges. Or they might be entitled to add to a drawing or text but not edit it; this is like being allowed to draw on the acetate but not "white out" anything. They might be allowed to edit a virtual copy of the text but not the original; this is like drawing and "whiting out" on the acetate. Or they might have full authoring privileges, in which case they can alter the original; this is like drawing and "whiting out" on the original. It is only when we have assigned both viewing and authoring privileges to hypercontexts that we have created collaboration contexts.

Inheritance hierarchies of collaboration contexts can be used to support level of privacy/publication of information. This is very important for supporting collaborative work by larger groups. To see why, consider the following considerations for viewing and authoring privileges in a three-level collaboration of 1) individuals 2) workgroups and the whole design team, containing several workgroups:

There are notes and drawings that I create but for various reasons do not want to share with my workgroup. Similarly, there are things that I and my workgroup collaborators create but might not share with our boss or the larger design team of which we are part. On the other hand, things agreed to in the larger design team ought to be used in the work of the smaller design workgroup; and things that this group agree to ought to be seen and used in my individual work. But I should not be able to alter things agreed to by my work group; nor should this workgroup-or I-be able to alter things agreed to by the larger design team.

The crucial point here is that inheritance of collaborative contexts naturally supports the sorts of restrictions of viewing and authoring-such as those described above-required for multi-level collaboration. If a workgroup inherits its collaboration context from the context of the larger design team, then by default the workgroup can see and use (add to and combine) all the data in the team context but cannot alter it. At the same time, the other team members cannot see anything in the workgroup's collaboration context. When the workgroup is ready, it can publish a selected subset of its work by assigning to this subset viewing privileges for everyone in the team. The same kind of authoring/viewing relationships exist when an individual's context inherits from the workgroup's collaboration context.

Several things should be noted about the collaborative context mechanism. One is that its concept of inheritance hierarchy does not restrict its use to the strict tree structure for the structure of a large design team. It also enables formation of arbitrary ad hoc group structures-such as in so-called "ad-hocracies." A second point is that collaborative contexts still allow conflicts to arise whenever authoring privileges for a given data item are given to multiple users. To deal with such cases, our strategy is to adopt "node locking." In other words, if any person wants to alter a node, he or she "locks" the node until finished making the desired changes. This prevents anyone else from making changes while the node is locked. Further conflicts between users over changes to a given information node must be handled by means not described in this article-such as, for example, the "argumentative agents" that we described in [McCall and Johnson 1997].

3. CONCLUSION AND FUTURE WORK

We have implemented the first full prototype of Web-PHIDIAS, a Web-centric version of the PHIDIAS Intelligent HyperCAD System. This system has a Java client and uses PHIDIAS as a server-side hypermedia database that manages both information and communication for collaborative design. Web-PHIDIAS employs a special context mechanism that uses virtual copies of hypermedia networks to support complex, collaborative communication in potentially large design groups.

Web-PHIDIAS is a major step toward our goal of supporting the argumentative approach to design first proposed by Rittel. It does this by using the Internet and the Web to distribute and capture argumentative information, to distribute computational aids to design, and thus to eliminate distance as a barrier to participation in design projects. Web-PHIDIAS is the first system to demonstrate how the information, communication and computation capabilities of networked computing can be used to promote the pluralistic view of design inherent in the argumentative approach.

Much work remains to be done to flesh out and refine the Web-PHIDIAS system. Our current prototype is still fairly primitive both in its user interface and its failure to provide access to the full functional of the PHIDIAS server-side system. For example, our Java client does not yet make use of the PHIDIAS functionality for knowledge-based critiquing. Other functionality found in previous prototypes-e.g., "argumentative agents"-will have to be re-implemented, perhaps from scratch. We also plan to incorporate voice-based communication and collaborative sketching into future prototypes; and these will require the addition of entirely new code.

In conjunction with the creation of additional functionality, we plan to do a great deal of testing of our system with end users. For this purpose we plan to focus our work on more conventional architectural domains than space-based habitation. We have for a number of years worked on issuebases for the design of houses and this domain is likely to become the focus of our user testing with individuals and small groups. Testing with larger groups will probably require a more complex domain.

REFERENCES

Kunz, W. and Rittel, H. "Issues as Elements of Information Systems," Working Paper # 131, Center for Urban and Regional Development, University of California, Berkeley, 1970.

McCall, R. *On the Structure and Use of Systems in Design*, Dissertation, University of California, Berkeley, 1978, University Microfilms, 1979.

McCall, R.; Schaab.; Schuler, W. and Mistrik, I. "A Database Model as the Basis of a Workstation for the Designer," in *Representation and Exchange of Knowledge as a Basis of Information Processes*, H.J. Dietschmann (ed.), North-Holland, Amsterdam, 1984.

McCall, R., MIKROPLIS: A Hypertext System for Design," *Design Studies*, Vol.10, No.4, 1989, Butterworth Scientific Ltd., Surrey, UK, pp. 228-238.

McCall, R.; Bennett, P.; d'Oronzio, P. Ostwald, J.; Shipman, F. and Wallace, N. "PHIDIAS: Integrating CAD Graphics into Dynamic Hypertext," in *Hypertext: Concepts, Systems and Applications* (Proceedings of the 1990 European Conference on Hypertext: ECHT '90), Rizk, A.; Stretz, N. and Andre, J. (eds.), Cambridge University Press, 1990, pp. 152-165.

McCall, R.; Morch, A. and Fischer, G. "Supporting Reflection-in-Action in the Janus Design Environment" in *The Electronic Design Studio* (Proceedings of the CAAD Futures '89 Conference), Mitchell, w.; McCullough, M. and Purcell, P. (eds.), MIT Press, 1990, pp. 247-259.

McCall, R. PHI: "A Conceptual Foundation for Design Hypermedia," *Design Studies*, Vol.12, No.1, 1991, Butterworth Scientific Ltd., Surrey, UK, pp. 30-41.

McCall, R. "The Locks and Keys algorithm for virtual copying of networks," Working Paper 73, Center for Environmental Design Education and Research, University of Colorado, 1993.

McCall, R.; Bennett, P. and Johnson, E. "An Overview of the PHIDIAS II HyperCAD System," in *Reconnecting: ACADIA '94* (Proceedings of the 1994 Conference of The Association for Computer Aided Design in Architecture), A. Harfman and pp. 63-74.

McCall, R. and Johnson, E. "Argumentative Agents as Catalysts of Collaboration in Design," in *Proceedings of the 1996 Conference of the Association for Computer Aided Design in Architecture (ACADIA '96*), Patricia McIntosh and Filiz Ozel (eds.), Tucson, AZ, 1996, pp. 155-163.

Mittal, S.; Bobrow, D.; and Kahn, K. "Virtual Copies: At the Boundary between Classes and Instances," *OOPSL4 '86 Proceedings*, Association for Computing Machinery, September, 1986, pp. 159-166.

Rittel, H. "On the Planning Crisis: Systems Analysis of the 'First and Second Generations" Bedriftsokonomen, No. 8, October 1972 (Norway).

Schoen, D. *The Reflective Practitioner*, Basic Books, 1983.