PHIDIAS: Integrating CAD Graphics into Dynamic Hypertext

Raymond J. McCALL, Patrick R. BENNETT, Peter S. D'ORONZIO, Jonathan L. OSTWALD, Frank M. SHIPMAN III and Nathan F. WALLACE

University of Colorado at Boulder
College of Environmental Design
Boulder, CO 80309, USA

ABSTRACT: PHIDIAS is a hypermedia system for supporting environmental design. It embodies a theory of design as continual alternation between two complementary activities: construction of solution form and argumentation about construction. To support these activities it implements a number of advanced hypermedia concepts. These include an applicative query language providing search by both structure and content, virtual structures, composite graphic nodes, query-based graphic clustering, and "triggered" queries which connect construction acts to relevant sections of the argumentative network. PHIDIAS constitutes a new type of integrated information environment for design.

KEY WORDS: Hypermedia, CAD, IBIS, Virtual Structures

1 Introduction

This paper presents an overview of PHIDIAS, a hypermedia-based computer-aided design (HyperCAD) system. PHIDIAS helps environmental designers--i.e., architectural, interior, landscape and urban designers--by providing support for two complementary design processes: construction of form and argumentation about construction. This is done by using both CAD graphics and IBIS argumentation [Conklin 1987]. Hypermedia provides a unified information management framework in which both text and vector graphics are stored at nodes in a hyperdocument. It also enables events during construction of form to provide entry into the precise section of the hyperdocument where argumentation is located which is useful for the current construction task.

The development of PHIDIAS has been driven by needs of environmental design rather than by the desire to explore hypermedia technology per se. Nevertheless, PHIDIAS addresses a number of leading-edge issues in hypermedia. In particular, it has implications for four of Halasz's "seven issues for the next generation of hypermedia systems" [Halasz 1988]:

1) search and query in hypermedia networks,
2) virtual structures,
3) computation in (over) hypermedia networks,
4) composite nodes.

2 Goals of the PHI

PHIDIAS is the PHI [McCall, Morsch, McCALL et al. 1987]. The PHI is also the PHI [McCall, Morsch, McCALL et al. 1987].

The goals of the PHI are to improve the design process by providing an "argumentative interface" to design. PHIDIAS is an "argumentative environment" that helps designers improve the design process. PHIDIAS is designed to help designers improve the design process. PHIDIAS is designed to help designers improve the design process.

2.1 IBIS

To implement the IBIS, the PHI uses the Information System. IBIS provides a framework for design discussions and deliberations and for the development of ideas. IBIS uses the PHI system. IBIS uses the PHI system. IBIS uses the PHI system.

A number of other systems have been developed that have their own system [Conklin 1985]. The PHI uses the PHI system. The PHI uses the PHI system. The PHI uses the PHI system.

2.2 The PHI Approach

The PHI (Procedural Hypertext Interface) approach broadens the use of hypertext to design. In Rick, the PHI every design...
PHIDIAS also suggests a new approach to the management of large collections of vector graphic data and to the coupling of these with non-graphic data. In particular it offers to replace the conventional concept of "layers" in CAD with a more powerful hypermedia-based concept of graphical clusters.

2 Goals of the PHIDIAS Project

PHIDIAS is the latest stage of a 13-year effort aimed at developing support for the PHI [McCall 1979] approach to IBIS [Kunz, Rittel 1970]. To understand PHIDIAS it is useful to know the goals and history of this effort.

The goals of PHIDIAS originate in Horst Rittel's call in the early 1970's for an "argumentative approach" to design. This approach aims to improve design by improving the designer's reasoning rather than automating it. Rittel advocated the abandonment of efforts to build design expert systems, which he claimed offered only "freeze dried prejudice." Computers should function, he said, like eyeglasses: helping you to see, rather than seeing for you.

2.1 IBIS

To implement the argumentative approach Rittel proposed the IBIS (Issue-Based Information System) method for documenting design discussion. IBIS organizes design discussion around the deliberation of issues. Issues are design questions, and deliberation is the process of arguing the pros and cons of proposed answers to the issues. The components of IBIS are issues, answers (also called positions), arguments, and resolutions. A resolution is a set of decisions to accept or reject answers to an issue. During deliberation, issues are raised, answers proposed, and arguments are given for or against the various answers or other arguments. An issue is resolved by selecting answers on the basis of the argumentation. In IBIS, as originally proposed by Rittel, the separate issue discussions are connected by a variety of relationships.

A number of IBIS hypertext systems have been developed, including Rittel's own system [Conklin 1987], gIBIS [Conklin, Begeman 1988], JANUS [Fischer, McCall, Morch 1989] and the PHIDIAS system, which is discussed here. PHIDIAS and JANUS use the PHI approach to IBIS.

2.2 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 inter-issue structure. In Rittel's IBIS, an issue is a design question which is deliberated; in PHI every design question is an issue, whether deliberated or not. PHI dispenses
with the various inter-issue relationships of Rittel's IBIS and uses instead only relationships indicating dependencies between issues. The most important of these is the serve relationship, which indicates that the resolution of one issue influences the resolution of another. PHI concentrates on one serve relationship called the subissue-of relationship, defined as follows:

Issue A is a subissue of issue B if A serves B—i.e., the resolution of A influences the resolution of B—and B is raised before A.

In PHI, design is represented as a tree-like structure of nested issue-resolution processes. Each project has a single root issue defining the project. Issues are generated top-down by recursively decomposing issues into subissues.

2.3 Developing Computer Support for PHI

The first software for PHI was PROTOCOL [McCall 1979], a system which took users along system-determined paths through non-linear structures of texts. Users of PROTOCOL frequently complained about not being able to determine their own paths through the network.

User control of journeys through the network became a central goal of MIKROPLIS [McCall, Mistrik, Schuler 1981][McCall, Lutes-Schaab, Schuler 1984]. MIKROPLIS (a German acronym for "microcomputer-based planning information system") was originally intended to allow retrieval only by query language. Navigation was also implemented, however, because users of early prototypes frequently pointed to nodes on screen and asked how to retrieve other nodes which were linked to them. An augmented version of MIKROPLIS is now the core of PHIDIAS.

MIKROPLIS is a hypertext system whose nodes are texts of essentially arbitrary length. Its retrieval aids include navigation and a query language allowing both content- and structure-based search [Halasz 1988]. It integrates authoring and browsing, allowing users to enter and edit nodes, add new link types, or restructure the network at any time. MIKROPLIS is outline-oriented, because outline format is ideal for the display of the tree-like PHI issue bases. MIKROPLIS, however, can handle any graph with labeled links, including "network" and cyclical structures. While designed to support PHI, MIKROPLIS comes with no built-in knowledge of PHI or IBIS. It can thus handle other types of text networks instead of or in combination with PHI.

Despite successful application of MIKROPLIS to policy making, lack of vector graphics in MIKROPLIS prevented its effective use in environmental design. Such design centers on the development of form, and its "language" involves closely related methods. In particular, the vector graphics that are familiar in architectural design require capabilities not present in MIKROPLIS.

In 1986, with vector graphics systems becoming more powerful and accessible, it became clear that the process of designing a product was not just a matter of seeking out all unanticipated facts and interfacings within it. Designers began to ask what the product would typically look like? or "What will the users think and feel while they were using the product?" These questions were mapped onto PHIDIAS by asking the designer saying that they are designing in-action.

We found that professional designers act and intervene in-action. According to Co-design as construct form, designers are engaged in the act of reflecting on the design process while acting as designers, and in-action of the designer is studied in-action.

Reflection of professional designers is divided into two major types: design-reflecting and context-reflecting. In action of the designer is studied in-action present, i.e., the design is being taken. To support designers in-action present, designer can be set up with a particular construction tool. The designer builds a construction task that is not interfering with context-reflecting.
involves closely intertwined verbal and graphic expression [Schoen 1983]. In particular, the verbal parts are so heavily populated with deictic references to graphics that they are often incomprehensible without the graphics. Thus, without support for graphics, a PHI-based system cannot succeed in design.

In 1986, we therefore began working on the development of a vector graphics system to couple with MIKROPLIS. But as this work progressed it became clear that the interface between the graphics and text was complicated by unanticipated factors. Originally, we thought that allowing vector graphics to be interlaced with the PHI texts would be sufficient. (In this scheme graphics would typically be answers to issues--such as "What should the facade look like?" or "What should the floorplan layout be?") But we found that design students had difficulties thinking in terms of issues, answers and arguments while they were devising form, i.e., while drawing. Think-aloud protocols indicated, however, that students' thinking processes while drawing could be mapped onto PHI structure in a straightforward manner. Why, then, were they saying that they could not think in the way they clearly seemed to be thinking?

We found the answer to this puzzle in Schoen's theory of design as continual alternation between situated action and the type of reflection he calls reflection-in-action. According to Schoen these cannot be done simultaneously. To construct form, the designer needs to be unselfconsciously and non-reflectively engaged in the action of drawing. Attempting to make the designer reflect while acting paralyzes action, for precisely the same reason that one cannot watch one's fingers while playing the piano.

Reflection-in-action takes place when action breaks down. There are at least two major types of breakdowns. One is when the designer's action results in unanticipated consequences—either good or bad. The second is when the designer is stuck and simply does not know how to act or which action to take.

To apply Schoen's theory to environmental design we operationalized his concepts by dividing design into construction and argumentation. Construction is the kind of action involved in shaping the form of the solution—e.g., with a CAD system. Argumentation is the kind of reflection dealt with in PHI. The task then becomes that of making PHI argumentation a part of reflection-in-action. As Schoen defines this concept it is reflection about action during the action present, i.e., the period during which reflection can still influence the action taken. To support reflection-in-action, the section of the issue base relevant to a particular construction task must be brought to the designer's attention while that construction task is still at hand. But it must not be done in such a way as to interfere with construction. There are two ways this can be accomplished: by
allowing immediate retrieval of this section of the issue base when construction produces surprising side-effects or by allowing such retrieval when the designer is deciding how to act. The former strategy is used by JANUS; the latter by PHIDIAS.

The JANUS system [Fischer, McCall, Morch 1989] provides integrated support for construction and argumentation by combining PHI hypertext with a graphic "construction kit". It uses knowledge-based critics to provide access from the construction system into the hypertext. These critics "look over the shoulder" of the designer during construction and critique partially constructed solutions. If the designer wants to see the argumentative background for these critiques—either to understand them or to argue with them—then a mouse click takes the designer into the exact section of the hyperdocument where this background is found. This greatly reduces the time and effort required to find useful information. It also supports reflection-in-action by augmenting the designer's ability to perceive breakdown situations and by presenting argumentation supporting reflection about them.

One limitation of JANUS' critics, however, is that they provide information only after a construction decision has been made—i.e., after a component has been added to the artifact being constructed. Designers also need information before making even tentative decisions and sometimes find it frustrating to be able to get needed information only after making a mistake.

3 The PHIDIAS System
The general screen layout of PHIDIAS (PHI Design Intelligence Augmentation System) is shown in Figure 1. PHIDIAS provides integrated authoring and browsing, an applicative query language, and dynamic restructuring of the issue base using virtual structures. It also supports construction with composite graphic nodes, a "bootstrapping" construction kit, and database-like "views" of clusters of graphic and textual nodes.

Much of PHIDIAS' functionality addresses limitations of JANUS. While JANUS itself is being improved in parallel with work on PHIDIAS, it will be useful to compare the functionality of PHIDIAS to the version of JANUS presented in the Hypertext '89 Conference [Fischer, McCall, Morch 1989].

3.1 Support for PHI-Based Argumentation
Use of JANUS has shown its static issue base to be inadequate. Designers need to modify the issue base as they browse through it, annotating it and tailoring it to their particular world views and their evolving understandings of problem situations. Even so, eliciting valuation by critics can enter complex interplay.

It is crucial to capture this problem in a way that will be thoroughly integrated into the design process.

Figure 1 -- PHIDIAS

To enter into a hyperdocument with the text or graph, one can enter into a PHIQL (PHIQL). To enter the graph, then use the text option, then use

3.1.1 The PHIQL OPTION
In PHIDIAS, there is an option for argument in PHIQL, which is analogous to a...
situations. Every display of a section of the issue base has the potential of eliciting valuative, corrective and supplementary information from the designer. It is crucial to capture such information as it arises. This requires that authoring be thoroughly integrated with browsing. PHIDIAS was designed to provide such integration.

Figure 1 -- PHIDIAS screen layout.

To enter issues, answers or arguments during browsing one simply points to the text or graphic node to be annotated, indicates the type(s) of link(s) to use, then enters the related text or graphics. In addition to entering single nodes, one can enter complex node clusters defined using the PHIDIAS Query Language (PHIQL). To edit a displayed node, one merely points to it, selects the edit option, then uses either the text or graphics editors.

3.1.1 The PHIDIAS Query Language

In PHIDIAS, the hyperdocument is always displayed as the result of some statement in PHIQL. Thus, the user always sees the hyperdocument in a manner analogous to a data base "view". This is powerful when combined with a
hypermmedia system of a fine granularity, such as PHIDIAS, because it allows nodes to be clustered in many user-specified combinations.

Although PHIQL appears descriptive to end users, it is in fact an applicative language partially resembling a subset of FP [Backus 1978][Baden 1983]. In PHIQL the names of link types are primitive operators. These link-operators may be combined--by 'application', 'composition' and 'construction'--with selection operators and other operators to create complex queries. PHIQL allows both structure- and content-based retrieval, separately and in combination. It is English-like, as is shown by the following examples of queries.

- argument 4 on answer 3 to issue 123
- all issues having answers containing "stove"
- all issues having keyword "cleanup"
- issue 148 with answers with arguments with authors

In these examples the 'with' means 'together with its', not 'having', as in conventional DBMS queries. Thus the last of the above queries displays a cluster of linked nodes (in outline format). The link-operators in the examples are 'answers', 'arguments', 'authors', and 'keyword'. The numbers '4' and '3' in the first query are selection operators which work with link operators. 'Issue' is a node type; '123' and '148' are selection operators that work with node types. "Cleanup" and "stove" are search strings.

Non-primitive operators--called masks--may also be defined, and these definitions may be recursive. Thus, the mask 'argument_tree' might be defined as

arguments with argument_tree.

This would allow us to define a mask called 'deliberation' as

answers with argument_tree.

We might then use the query

all issues with deliberation
to display the current status of a discussion.
to display the discussion of every issue in the hyperdocument.

PHIQL addresses Halasz's issue of providing hypermedia query languages that offer both content-based and structure-based search. PHIQL is advantageous in two additional important respects. First of all, because it is English-like, it is easily learned by end-users. Secondly, its algebra of link operators is particularly well suited to search in hyperdocuments.

3.1.2 Virtual Structures
User input into the issue base often has implications for altering other parts of the issue base. For example, the designer who rejects answer 1 of issue 102 in Figure 2 should not have to deal with issue 104 or the portion of the issue base which serves it. In other words, this information should "disappear" from the issue network when the answer is rejected.

ISSUES:
101: What should the design of the kitchen be?

SUBISSUES:
102: What should the functional areas of the kitchen be?

ANSWERS:
1: eating area
   DECISION:
   reject
2: food preparation area
   DECISION:
   accept

103: What should the design of the food preparation area be?
104: What should the design of the eating area be?

Figure 2 -- Static issue base.

To provide this sort of automatic restructuring of the hyperdocument PHIDIAS uses virtual structures [Halasz 1988], i.e., queries embedded at nodes. When the command is given to display such nodes, the queries are evaluated and the results displayed. This allows automatic updating of changing information. It also makes the structure of the hyperdocument appear to change as a function of user input.

We have implemented virtual structures using PHIQL. Figure 3 illustrates this in the context of the above example. Issues 103 and 104 have been replaced by embedded PHIQL queries, represented by <<bracketed>> text. This makes their display conditional on the acceptance of the answers that they presuppose.
ISSUES:
101: What should the design of the kitchen be?

SUBISSUES:
102: What should the functional areas of the kitchen be?
ANSWERS:
1: eating area
DECISION:
reject
2: food preparation area
DECISION:
accept
<< display issue 103 if decision on answer 2 to issue 102 is "accept" >>
<< display issue 104 if decision on answer 1 to issue 102 is "accept" >>

The effect of virtual structures upon the issue base is shown in Figures 4 and 5. Both were produced with the same query. The difference results from first accepting answer 1 (Figure 4), then rejecting it (Figure 5). This reduction in size of the hyperdocument through elimination of irrelevant substructures greatly facilitates the use of large argumentative networks.

Figure 3 -- Issue base with embedded queries.

Figure 4 -- Unpruned issue base.

Figure 5 -- Pruned issue base.

3.2 Support for Designing PHIDIAS supports the design of physical objects. McCall, Morello, and Dam (1988) have designed object-oriented hyperdocuments which are in turn manipulated by a tool (a "palette") by means of rotation, scaling, etc.

The principle of the design process is computer-graphics object-based (Van Dam et al. 1988). A central idea is that all its components (the "phragments") are all connected to each other. PHIDIAS, the hypermedia configuration manager, allows for the instance transformation of the objects in the hypermedia configuration.

PHIDIAS extends this concept by adding a number of built-in functions and the ability to "bootstrap" its configuration.

The information is organized in a tree-like manner, with nodes in the tree being hypermedia-based. Each node is constructed of a hypermedia object that can be retrieved through a query.

One useful feature is the ability to query the object by its name. For example, "kitchen 101" is retrieved with a query that includes the term "kitchen" and is retrieved with a query that includes the term "kitchen 101".

Like text clustering, the hypermedia configuration manager in PHIDIAS. The configuration system can be queried, similar to text clustering, with a query that includes the term "kitchen 101".
3.2 Support for Construction

PHIDIAS supports construction with domain-specific construction kits [Fischer, McCall, Morch 1989], each containing building blocks for creating a type of designed object—e.g., a kitchen (Figure 1). Each building block is a node in the hyperdocument. Building blocks are assembled into objects, such as kitchens, which are in turn nodes in the hyperdocument. Assembly is done using a direct-manipulation-style graphics editor. Items are selected from the construction palette by mouse and dragged into the construction area, where they can be rotated, scaled and further translated.

The principle behind the PHIDIAS construction kits is the conventional computer-graphic concept of object hierarchies with instance transforms [Foley, Van Dam 1982] (not to be confused with object hierarchies in object-oriented programming). In terms of Halasz's "seven issues," PHIDIAS is significant in that all its construction consists of the creation of composite nodes. Thus, in PHIDIAS, the conventional computer graphics concept of object hierarchies with instance transforms becomes a natural way of dealing with the leading edge hypermedia concept of composite nodes.

PHIDIAS can manage a large number of construction kits each with a large number of building blocks. In fact, new construction kits can be built using graphic objects constructed with the editor, thus the system can in principle "bootstrap" itself to ever-higher levels of construction complexity.

The information management capabilities of PHIDIAS can be used to organize and retrieve building blocks and constructed objects—since both are nodes in the hyperdocument. Thus while JANUS had a separate (non-hypermedia-based) "catalog" of previously constructed objects, PHIDIAS stores constructed objects in the hypermedia network. This means that PHIQL can retrieve these designs using a wide range of query types and search criteria.

One useful result of the PHIDIAS strategy for data management is the possibility of displaying hierarchical clusters of graphic nodes. These are the graphic analog of the hierarchical clusters of text nodes shown in outline format. Like text clusters, graphical clusters are generated by explicit or implicit queries in PHIQL. The graphics image in Figure 1 could have been produced with the query,

```
kitchen 101
   with countertop
       and cabinets
       and sink
   and stove.
```
Hypermedia-based graphical clusters provide an alternative to the conventional CAD concept of "layers", which are usually named but not linked. The PHIDIAS graphical clusters are intrinsically more fine-grained than CAD layers and provide a more natural representation of the relationships among the many objects in any design. Graphical clusters allow the database concept of "views" to be implemented in CAD. In addition, graphical objects can be clustered with textual information. Thus graphics can be annotated with text and text illustrated with graphics.

3.3 Integrating Construction and Argumentation

PHI issue bases contain information relevant to construction. But our experience with users of previous PHI hypertext systems has shown that this information is unlikely to get used unless special measures are taken to tie its retrieval to construction acts. In particular, it must be possible to immediately retrieve just that argumentative information which is relevant to the current construction task. This means the system must know what construction task the designer is currently engaged in and must provide access into the precise section of the hyperdocument where the information relevant to this task is to be found.

We have implemented this in PHIDIAS by allowing mouse button clicks during construction to trigger, i.e. execute, canned PHIQL queries. Certain construction activities can be treated as implicit requests for information. When selecting a building block from the construction kit the designer should get information on which item to select. In PHIDIAS, when selecting from a palette of different refrigerators, a mouse click on the "fridge" button will provide information on the issue "What type of refrigerator should be used?" (Figure 1). A click on a refrigerator in the palette will give just the argumentation on that selection. Once a refrigerator has been selected, a click will retrieve argumentative information on where to place it (Figure 6). Issues about any object already in the construction area can also be retrieved by clicking on that object in the construction area. Once initial argumentative information has been displayed, the PHI links allow navigation into deeper levels of the issue base.

PHIDIAS' strategy is weaker but more general than JANUS' strategy of using knowledge-based critics for providing entry points into the issue base. The critics remind designers that they have violated principles of design but depend on the existence and codification of such principles as rules. Much useful design information cannot be put into this form. The two strategies, however, should not be regarded as alternatives, but rather as complementary.

4 Conclusion

4.1 Future Work

Virtual structure construction and restructuring of the existing construction tools and dishwasher, the user tool

Virtual structure representation in the hypermedia network

Nested virtual structures, created during construction, are allowed to exist and can be used for complex inference.

The creation of a system for augmenting PHIDIAS with a richer set of types of conditions...
4 Conclusion and Future Work

4.1 Future Work

Virtual structures have considerable unexplored potential for intelligent restructuring of hypermedia networks. They might even be used to restructure the construction kit palettes. For example, if the designer decides not to have a dishwasher, then dishwashers would not be available as building blocks.

Virtual structures also provide a mechanism for computation over hypermedia networks which is different from that suggested in [Halasz 1988]. Nested virtual structures, i.e., virtual structures which call on other virtual structures, create the possibility for powerful recursive computation. If queries are allowed to test conjunctions or disjunctions of conditions, virtual structures can be used for basic logic programming. This would allow PHIDIAS to make complex inferences based on user input.

The creation of more powerful virtual structures can be accomplished by augmenting PHIQL to allow compound conditionals, existence tests, and other types of conditions. PHIQL could also be extended to test for conditions in the
contents of composite graphic nodes—such as relative positions of component objects. This would greatly enhance the ability to search among graphic nodes. It could also be used to implement critics such as those in JANUS, most of which are based on relative positions of component objects. This would further extend the range of hypermedia functionality.

For a graphic system to be truly useful for environmental design it must have three-dimensional capabilities. We are currently working on such a graphics system and hope to have a prototype 3-D PHIDIAS within a year.

4.2 Conclusion

Good design requires reflection-in-action. Computer systems for design should therefore provide integrated support for both construction of solution form and argumentation about construction. Hypermedia is a natural vehicle for creating such design environments. PHIDIAS explores the potential of hypermedia to provide a unified architectural framework for such systems.

To support design we have found it necessary to implement a number of advanced hypermedia concepts in PHIDIAS. These include an applicative query language providing search by both structure and content, virtual structures, composite graphic nodes, query-based graphic clustering, and "triggered" queries to connect construction actions to relevant sections of the argumentative network. Each of these has opened new areas of potential future research for PHIDIAS and hypermedia in general.

Acknowledgements

Work on PHIDIAS was supported in part by the Colorado Institute for Artificial Intelligence (CIAI). CIAI is sponsored in part by the Colorado Advanced Technology Institute (CATI), an agency of the State of Colorado. CATI promotes advanced technology, education, and research at universities in Colorado for the purpose of economic development.

References


