An Overview of the PHIDIAS II HyperCAD System

Raymond McCall  
Patrick Bennett  
Erik Johnson  
University of Colorado, Boulder  
University of Colorado, Boulder  
University of Colorado, Boulder

The PHIDIAS II HyperCAD system combines the functionality of CAD graphics, hypermedia, database management and knowledge-based computation in a single, highly integrated design environment. The CAD functionality includes both 3-D and 2-D vector graphics. The hypermedia includes support for text, raster images, video and sound. The database management enables persistent storage and interlinking of large collections of text, images, video, sound and vector graphics, i.e., thousands of vector graphic objects and drawings in a single database. Retrieval is provided both through use of “associative indexing” based on hyperlinks and through use of an advanced query language. The knowledge-based computation includes both inference and knowledge-based critiquing.

A highly unusual feature of PHIDIAS II is that it implements all of its functions using only hypermedia mechanisms. Complex vector graphic drawings and objects are represented as composite hypermedia nodes. Inference and critiquing are implemented through use of what are known as virtual structures [Halasz, 1988], including virtual links and virtual nodes. These nodes and links are dynamic (computed) rather than static (constant). They are defined as expressions in the same language used for queries and are computed at display time. The implementation of different kinds of functions using a common set of mechanisms makes it easy to use them in combination, thus further augmenting the system’s functionality.

PHIDIAS supports design by informing architects as they develop a solution’s form. The idea is thus not to make the design process faster or cheaper but rather to improve the quality of the things designed. We believe that architects can create better buildings for their users if they have better information. This includes information about buildings of given types, user populations, historical and modern precedents, local site and climate conditions, the urban and natural context and its historical development, as well as local, state and federal regulations.

We have recently concentrated on design domains that are so information intensive that complex information is required for even minimally adequate design. One such domain is the design of space-based habitation—including space stations, lunar and Martian habitats. For several years we have been working on this domain with the Johnson Engineering Corporation (JE), one of NASA’s contractors, and more recently with the Flight Crew Support Division of NASA’s Johnson Space Center in Houston. The aim of these efforts is to use PHIDIAS to provide information support for the design of interiors of space habitats.

No one person knows enough to design a space-based habitat. Expert knowledge of many different and highly specific types is needed even for “simple” decisions: ergonomics, psychology, space medicine, crew training, crew selection, detailed knowledge of the history of prior space missions, knowledge of the volumes of design guidelines and regulations devised by NASA, and so forth. The

As the applications of communication technologies to design processes rapidly expand, research in computer-aided design is shifting focus from the medium to the purposes and processes of design thinking. The impacts of these technologies will challenge design thinking to evolve to new levels of sensitivity and efficiency. Conversion of computer data, information to communication, will
problem for any designers of such habitats is how to proceed when they simply do not have in their heads crucial knowledge needed for design.

One suggestion might be to provide the designer with an information system containing the knowledge needed for design. While such a system is unlikely to be capable of containing everything a designer needs to know, it could nevertheless help the designer to avoid many major errors. Unfortunately, giving designers a conventional information system would do little to help them, because such systems can help you only if you ask them the right questions—i.e., pose the proper queries. This does not work for the fundamental reason that the designers do not know what they do not know. They thus do not know when to ask questions or what questions to ask, much less how to phrase the questions or when to stop searching. They do not know when they need information, much less how to go about getting it. Even if designers were able to query a conventional information system, they would find that only a fraction of the information they retrieved was relevant to the specific design tasks they have to deal with.

PHIDIAS II has been designed to overcome the major limitations of conventional information systems. It provides several ways for designers to retrieve information for design tasks without having to formulate and pose queries. It delivers information for design tasks when that information is needed in design. And it retrieves just that information that is relevant for the design situation at hand.

In the remainder of this article we explain the functions of PHIDIAS II—i.e., what it does for the designer—and how they are implemented. We begin by describing the way in which PHIDIAS II provides information for a design task before that task has begun. We then show how it provides information during design. After this we explain how information about the design project, e.g., design rationale, can be obtained after the project is completed. Finally, we discuss the basic principles of the PHIDIAS II system architecture, including how both complex vector graphics and knowledge-based computation are implemented using only hypermedia mechanisms.

The Scenario Project: A Preliminary Lunar Habitat

For several years we have been working with JE on the development of information support for the design of lunar habitats. The examples in this article are taken from that domain. JE asked us to work on the design of a preliminary lunar habitat for the years 2010-2020 that would serve as a prelude to development of a larger lunar base in the 21st Century. One idea JE employees had was to start the preliminary habitat by placing on the Moon a single module of the type to be used in the Space Station. In particular, we were asked to look at a scenario of a mission in which four astronauts of both sexes and from multiple nations would live and work in such a module for 45 days at a time. Since the module is only 23 feet long and 14 feet wide, this presents a “tight quarters” situation in which social, ergonomic, psychological and aesthetic factors are likely to play a crucial role in the success of the mission. Figure 1 shows a PHIDIAS II screen image with several views of a partially designed lunar habitat for this scenario (Figure 1). The plan shows a “pinwheel” configuration of bunks (on the right in the plan view) above a work station area. The galley/wardroom areas (on the left in the plan view) has not yet been fully designed, though some equipment has tentatively been placed in the galley. The examples of this article will show how PHIDIAS II can provide information for this part of the lunar habitat design project.

Information before a Design Task Begins

Before designing, the designer will generally want to obtain information about a variety of issues, including user activities and requirements, costs and constraints.

This obviously applies to the design project as a whole, but it can also apply to each part of a design project. Thus, before designing a special type of area within a building, such as a library room, an operating room or a scientific laboratory, the designer might want to obtain or be reminded of the information relevant to
this part of the design. Similarly, before designing the
galley-wardroom area in the lunar habitat, the designer
might want to look at what the relevant issues are likely
to be and what information is useful for dealing with
these issues.

PHIDIAS II helps the designer obtain such
information before design of the galley-wardroom
begins. If, for example, designers want information
relevant to the layout of this area before beginning to
work on the layout, they can get this merely by double-
clicking on a vector graphic representation of this area
in the plan or any other view shown in Figure 1. The
system would then pop up a window displaying the
issue that represented this part of the design problem.
In this case the issue would be “What should the design
of the galley-wardroom area be?” Clicking with the
mouse on the “plus button” displayed to the left of
this issue will cause display of the “first level” of
information useful for answering this question (issue)
 tossed together with additional “plus buttons.” Clicking on

depend, to an important degree, upon human powers of conception and
perception. Re-examination of conception and perception as co-dependent and
mutually stimulating mental activities may yield additional insights to the potential
interaction of design thinking and media, both electronic and traditional.

A concept can be seen specifically as an abstraction drawn from the
specific. The formation of concepts is the continuing activity that initiates.
these additional buttons will display information at the
next "level." This allows the user to selectively expand
an outline structure of information potentially useful
for design of the galley-wardroom area (as shown in
Figure 2). The information displayed is derived from
various NASA documents as well as videotapes of
actual design sessions with IE personnel.

As Figure 2 illustrates, we currently organize
the information displayed in the form of an IBIS
(Issue-Based Information System) [Kunz and Rittel,
1970] based on the PHI (Procedural Hierarchy of
Issues) approach to IBIS [McCall 1989].

PHIDIAS II is not limited to IBIS and in fact contains
no internal knowledge of the IBIS-PHI method. This

Figure 2: Information for design of the galley-wardroom area is retrieved by clicking with the mouse on a vector graphic representation of this area. The information is displayed in outline format and is organized using the PHI approach to IBIS. A context selector box for different design domains is shown on the upper left.

Howard's End
method is, however, one useful way of structuring highly diverse design information into a consistent set of categories. With IBIS, the categories are issues (questions), suggested answers, arguments, decisions, and several types of related information. The PHIDIAS version of IBIS is especially useful because it emphasizes the way in which answering one issue is useful for answering another. Thus, when confronted with a given issue such as "What should the design of the galley-wardroom area be?" a properly structured PHIDIAS information base points out what other issues should be answered in order to answer the given issue—as is shown in Figure 2. These issues on which the answer to the given issue depends are known as subissues of the given issue. Subissues can, of course, themselves have subissues.

Information for Multiple Projects in Multiple Problem Domains

The upper left portion of Figure 2 shows an open context selector list. This shows a hierarchy of problem domains. For each such domain there is an appropriate collection of multimedia information, including text, raster images, vector graphics, video and sound. This indicates that PHIDIAS contains information not only for a single type of project but for a range of projects in the design of space-based habitats. The general domain context of space-based habitation has its own set of generalized information. This context has been broken down into two more specific domain contexts: microgravity—i.e., zero gravity—and partial gravity; there are information collections for each of these. Partial gravity has been broken down into Martian and Lunar, each with its own collection of relevant information. Under the Lunar domain are the contexts for the habitat and the above-described scenario (4 astronauts for 45 days) under which are separate contexts for several different solution attempts: John’s, Mike’s and Ray’s. This shows that PHIDIAS II not only contains information for use in various projects of various types, it can also contain the solutions to those projects—e.g., in vector graphic form—together with the rationale for those solutions—e.g., in the form of text and voice-based annotation. Thus, unlike conventional CAD systems, PHIDIAS II is not a file-based system for handling individual projects; it is instead a multimedia database management system for integrated creation, storage and retrieval of information about multiple projects. By storing multiple projects in a single database, PHIDIAS II allows designers to examine precedents while designing. The record of precedents can include both the graphical decisions made and the rationale underlying those decisions.

Information while Design is in Progress

Information for Design Tasks

PHIDIAS offers several ways of delivering information for form-making tasks that designers engage in. Its main method for doing this involves the use of domain-oriented construction kits. These are, in effect, symbol libraries for which the system possesses formal semantic knowledge. Typical domain-oriented construction kits for design of space habitats include those for walls, doors, beds, tables, seating devices and equipment of various types.

Information can be delivered for any graphical actions involving elements of construction kits—e.g., for selection and placement of these elements in the configuration. For example, Figure 3 shows in the upper left-hand corner a small part of a construction kit for selection of tables for use in the lunar habitat (Figure 3). (Each of these table shapes was created using the PHIDIAS II system.) The selection of a table for the lunar habitat might seem a simple and trivial decision, yet there are significant complicating factors even in this apparently simple case. Each of these tables was itself the result of a major design effort by NASA or its contractors; each has complex reasoning underlying its design. One of the table designs is a variation on a design used in Skylab, a design for which there is extensive documentation of use, including video footage. Other table designs have undergone simulated acceleration and guides the processes of design thinking. A percept can be defined as an impression, formed in the mind, of something one is aware of through the senses. The concept and the percept are codependent. The specifics from which concepts are formed can only be provided through perceptions of the world of the senses; and the percepts that screen information from the senses are directed by concepts already held in the mind. Frank Lloyd Wright's concept of the
testing. Without an understanding of such background information and a knowledge of the complex criteria for habitat design that NASA has developed, the designer might not be able to make an intelligent choice of table design. PHIDIAS II enables the designer to retrieve relevant information for selecting elements from construction kits by providing each construction kit window with an advice button (as is shown in Figure 3). Clicking on this button with the mouse causes display of this information (also shown in Figure 3). Notice that some of this information also appeared in Figure 2. The crucial difference here is that the advice button for selection of elements gives only that information useful for table-selection decisions, whereas Figure 2 contains substantial additional types of information. Notice also that Figure 3 shows that the information retrieved can include raster graphics images and video. In this we see a NASA photo of simulated use of one of the table designs. We also see a video clip of an actual meal aboard the Space Shuttle.

Knowledge-Based Critics
Advice buttons for form manipulation tasks provide a way for designers to get information that is relevant for the task at hand without having to issue queries to the system, much less to know which queries to pose. The information they retrieve is also highly relevant—in contrast to conventional retrieval systems, which force their users to wade through much irrelevant data to find truly useful information. Yet the advice button concept still has some drawbacks. For one thing, designers must understand their need for advice. Unfortunately, many of the cases in which designers test little need for advice are those in which they are most profoundly ignorant of the complex and non-intuitive consequences of their decisions. Another problem is that the advice button concept cannot deal with cases where designers might fail to provide needed architectural elements or equipment.

Providing knowledge-based critics helps to alleviate these problems. These critics are software agents—of varying degrees of intelligence—that "look over the designers' shoulders" while they design and point out potential problems with their designs. Such critics are highly limited in the problems they can recognize and thus could not effectively substitute for human critics. They are, for example, unlikely to be useful for aesthetic criticism. Nevertheless, these critics can play a valuable role by alerting designers to measurable problems and calling their attention to potentially useful information for dealing with those problems.

PHIDIAS II provides a powerful facility for creating knowledge-based critics. Figure Four shows a situation in which such a critic has fired to alert the designer to the fact that NASA requires that every galley have an inventory management station (Figure 4). To see why NASA created this rule—either to understand it better or to challenge it—the designer need only click on the rationale button provided on the critic window.

Capture of Design Rationale
The Strategy of Differential Reasoning
At any point at which PHIDIAS II displays information the designer can input additional issues, answers, arguments or annotations of any type. This enables the designer to augment the generalized argumentative information for the problem domain and thus tailor it to the specific design project at hand. To create a detailed record of the reasoning for a project, the designer need only record the differences between the domain-oriented reasoning and the project-specific reasoning. There is no need to input all the rationale from scratch in each new project. This greatly reduces the amount of work and disruption required for recording design rationale.

Differential rationale typically emphasizes the recording of choices amongst pre-given alternatives, exceptions to design rules and input of new solution ideas. By requiring the recording only of this differential rationale PHIDIAS II allows the designer to concentrate on describing just those things that are original to the project.

Voice Capture
While much less effort is needed to input only differential rationale, sometimes even the effort of typing in this rationale is too much. In these cases,
Johnson Research Tower as structurally akin to a tree is based in part upon the perception of the structural characteristics of a specific tree. That perception, in turn, may well be derived from a general concept that all natural objects can be seen as structures and that there is something to be gained from the use of analogies based on nature.
Figure 4. Knowledge-based critics alert designers to potential problems with their design solutions and guide them to useful hypermedia information. The sound recording panel on the left can be used for voice-based annotation.

---

Howard's End

ACADIA 1994
Information after Design is Completed

A standard problem with design precedents is that, as Rindt put it, "They do not speak for themselves." In other words, it is typically not clear why the designer who created them made them the way they are. Understanding precedents is therefore greatly aided by a record of the reasoning that underlay the decisions made by their designers. PHIDIAS II enables the retrieval of such reasoning for projects created using it. A designer, historian or anyone else wishing to know what reasoning underlay a given part of a designed object has only to retrieve the vector graphic representation of the design and then double click with the mouse on parts of that design for which an explanation of the rationale is desired. The system will then display the reasoning—in the same manner as shown in Figure 2—including the domain-oriented reasoning and the project specific rationale that the designer recorded.

System Architecture of PHIDIAS II

This section explains the radically integral, hypermedia-based system architecture (not to be confused with architecture in the sense of building design) of PHIDIAS II. In particular, it explains how complex vector graphics and knowledge-based computation are implemented within PHIDIAS. It then explains basically how each aspect of the above-described functionality was implemented using this system architecture. This is meant to show the singular advantages that a purely hypermedia-based approach offers for information environments for design.

Some of the functions of PHIDIAS II are new—multimedia, 3-D graphics, and inheritance of hypermedia networks—but some were contained in our predecessor prototypes, such as JANUS [McCall, Fischer, March 1990] and PHIDIAS I [McCall et al. 1990]. The PHIDIAS II system, however, differs decisively from its predecessors in having a purely hypermedia-based architecture. JANUS got its structure by loosely coupling three types of systems: a knowledge-based system, a vector graphics system and a hypermedia system. Similarly, PHIDIAS I loosely coupled the MIKROPLIS hypertext system [McCall 1989] with a specially written vector graphics system. But PHIDIAS II has no functionally differentiated subsystems comparable to those in JANUS and PHIDIAS I. Instead, it uses hypermedia mechanisms alone to provide knowledge-based computation and CAD graphics as well as standard, navigational hypermedia.

Hypermedia is software for navigating through networks of multimedia nodes and links—such networks being known as hyperdocuments. First conceived in the 1930s by Vannevar Bush, President Roosevelt’s Science Advisor, hypertext and hypermedia systems were first implemented in the 1960s by Douglas Engelbart [Engelbart 1963], who also invented the mouse, the word processor, on-screen windows, outline processing and computer conferencing. In fact, these were all invented as part of Engelbart's hypertext system NLS/AUGMENT. Despite their early beginnings, hypertext and hypermedia only really began to be widely used in such systems as HyperCard, MOSAIC and various hypertext help systems, such as Microsoft’s help system for Windows. PHIDIAS II’s advanced hypermedia mechanisms include 1) composite hypermedia nodes, 2) a special hypermedia language, called LINQ, 3) virtual structures, including both virtual links and virtual nodes defined with LINQ, and 3) a prototype-instance mechanism for inheritance of hypermedia networks.

Complex Vector Graphics as Composite Hypermedia Nodes

We have for a number of years been using a scheme for representing complex vector graphics objects which is based on concepts in [Foley and van Dam 1982] and which have become the basis for the PHIGS standard. This scheme, of course, in no way depends on hypermedia. With PHIDIAS II, however, we have in effect translated this scheme into hypermedia nodes and links.

In our system, all graphical objects, whether simple or complex, are represented as hypermedia nodes. Primitive graphical elements—e.g., polylines—are represented as non-composite
hypermedia nodes. A complex graphic object is represented as a composite node constructed as an augmented part-whole hierarchy in which part-whole relationships are special hypermedia links. The augmentation starts with the addition of so-called instance transform parameters—i.e., parameters for scaling, rotation and translation in 3 dimensions—to the part-whole links. These parameters are used to construct affine "instance" transforms, in the form of homogeneous transform matrices, that are used in the computation of displays. These allow a given object—i.e., vector graphic node—to appear as a part of more than one object. For example, it allows a rectangular polyline to be used as all six sides of a cube. It also allows the cube to be stretched flat and relocated to form a table top, the legs of the table, the legs of a chair, the seat and back of the chair. It also allows multiple instances of the chair and table to be used in a single scene.

A scene consists of a single, composite node. The computation of a vector-graphic display is accomplished by depth-first recursive traversal of the node-link structure constituting this composite node to construct a list of all primitive nodes (e.g., polylines). In our current version of the system, display is done by first sorting the primitive nodes in back-to-front order, then displaying them back to front—an approach known as the "painter's algorithm" for hidden surface removal.

Knowledge-based Computation through Evaluation of Virtual Structures

PHIDIAS II performs knowledge-based computation through evaluation of so-called virtual structures [Halasz 1988]. Virtual structures are computed, rather than constant/static, hypermedia links and nodes. The virtual structures in PHIDIAS II are defined as expressions in the LINQ language and are only evaluated at display time. Virtual nodes have LINQ expressions instead of fixed content, such as text or graphics. Virtual links are named LINQ expressions whose names appear to the user as links where, in fact, no such links exist.

PHIDIAS II represents a knowledge base as virtual and actual (non-virtual) hypermedia network structures. It performs its knowledge-based computation through evaluation of nodes and links while traversing these network structures. Atomic facts are represented as actual link structures rules and higher-level facts are represented as virtual structures. The virtual structures include both virtual links and virtual nodes, with the former generally being both more common and more powerful.

LINQ (Language for Inference, Navigation and Querying)

LINQ is a declarative language combining concepts from functional and logic programming. Its closest relative in the functional languages is Backus' FP [Backus 1978]. Like FP, LINQ provides the capability for defining hierarchies of non-primitive operators that are evaluated in a recursive manner. LINQ also has certain features of logic programming languages such as PROLOG, including clause-like compound conditionals. The functional style was adopted in LINQ because of its usefulness for the traversal of networks. In particular, most of the functional operators in LINQ are concerned with link traversal.

As is shown in the various examples listed below, LINQ is highly English-like. This has important advantages for intelligent hypermedia in general and for the domain-oriented design environments whose creation is our ultimate goal. Intelligent hypermedia systems will in general be loaded with large amounts of knowledge, including many formal facts and rules. If such systems are to empower their users rather than to overpower them, the system's knowledge base must be understandable by its users. To be understandable, formal knowledge must be represented in terms of human concepts rather than in computer concepts. Having an English-like syntax allows the creation of self-documenting code. LINQ's English-like character derives from the happy coincidence that a certain subset of English syntax corresponds closely to a hybrid of functional and logic-programming syntax, a syntax that in turn fits a node-link representation of knowledge. This allows LINQ to provide highly efficient, high-level computation that is "transparent"
for the reader and which closely fits the underlying structure of a hypermedia network. It should be remembered, however, that our intention here is not to make human language comprehensible to computers, but rather to make computer language comprehensible to humans. Unfortunately, the former is still not realistic given the current state of technology.

Implementation of the PHIDIAS II Functions
Implementation of Hyperlinking to Graphical Objects

The PHIDIAS II functions for delivering information before design were made possible by hyperlinking information to graphical objects—in the above example the graphical object was a representation of the galley-wardroom area. Since physical objects as well as the information to be linked are all hypermedia nodes, the linking becomes trivial to implement. It is no more difficult than the creation and traversal of any other hypermedia link.

Implementation of Hyperlinking to Tasks

The PHIDIAS functions for delivering information for design tasks—such as selection and placement of elements from physical construction kits—during design were implemented by hyperlinking information to advice buttons. This function is in a manner similar to the linking of information to graphic objects. The implementation, however, is somewhat different. In the current version of PHIDIAS, neither tasks nor advice buttons are hypermedia nodes. We have therefore arranged to have the pressing of a button cause the evaluation of a LINQ query, and this in turn causes the display of information. In future versions of PHIDIAS we hope to implement tasks and advice buttons as hypermedia nodes, thus simplifying and unifying the system.

Implementation of Knowledge-based Critics

Knowledge-based critics are implemented as virtual structures that function as on-added predicates, firing (being evaluated) whenever an item of a given type—e.g., a table—is added to a graphical scene. As mentioned above, these virtual structures are defined in LINQ. The LINQ expression for the critic shown in Figure 4 is as follows:

If the galley contains no inventory management station, then display the inventory management error message

PHIDIAS II can handle a range of different types of critics. A representative sample derived from NASA guidelines is given below. Please note that these are not the English equivalents of the LINQ expressions but that actual LINQ expressions themselves. Please note also that the expression “SP 38-A” is not part of the LINQ syntax per se, but is the name of a particular scientific package.

If the emergency airlock tools are not next to the airlock, then display the airlock tools error message

If the galley does not contain personal mobility aids, then display the galley mobility aids error message

If the sleeping area is near the wardroom, then display the wardroom and sleeping message

If the SP 38-A experiment package is near a microwave oven, then display microwave error message

Conclusions and Future Work

We have developed a hypermedia-based design environment, called PHIDIAS II, that offers 4 basic types of functions: 1) 3-D CAD-graphics, 2) knowledge-based computation, 3) navigation in a hyperdocument and 4) database management. Designers can use this system to graphically construct solutions; and, as they do so, the system uses knowledge-based critics and hyperlinks to call their attention to useful information for design. This information includes text, sound (e.g., voice-based annotation), vector graphics, raster images and video.
PHIDIAS II is unique in that it has a radically integrated system architecture based exclusively on hypermedia. In other words, it implements not only hypermedia functionality but also vector graphics, knowledge-based computation and database management using only hypermedia-based mechanisms. These mechanisms include composite nodes, a declarative hypermedia language, virtual structures and a prototype-instance scheme for inheritance of hypermedia networks. This multifunctional character of PHIDIAS II challenges conventional notions of what the inherent function of hypermedia is. This has important implications both for the future workings of our design environments and for the future of hypermedia in general.

Considerable work still needs to be done on the PHIDIAS system. We are currently exploring several types of added capabilities. For one thing, we are expanding the power of LINQ to make it a full-blown programming language. We are also extending PHIDIAS for use by distributed groups engaged in cooperative design in a networked computing environment. We are currently putting substantial amounts of NASA video into PHIDIAS and looking at ways of facilitating the input, structuring and indexing of video data. In addition, we have spent the past year implementing a sketching system to use with PHIDIAS on pen-based computer interfaces. Finally, we continue to look for additional ways of implementing all aspects of system operation—including construction kits and other interface elements—using only hypermedia mechanisms.

References


• The critical human tasks of understanding, forming an opinion and decision-making depend upon the links between perception and conception; concepts are only as powerful and dependable as the perceptions of realities to which they are related. The novice designer might perceive a structural metaphor of a tree; Wright based his design of the Price tower on his perception of the tree as metaphor for shelter, landmark, and thematic unity as well as
Howard's End

ACADIA 1994