TOWARDS A HYPERMEDIA APPROACH OF DATA ORGANIZATION IN BUILDING-MODELING CAD SYSTEMS

by

Jian Zhao

Bachelor of Architecture
Tsinghua University, Beijing, China
1988

SUBMITTED TO THE DEPARTMENT OF ARCHITECTURE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF
THE DEGREE OF
MASTER OF SCIENCE IN ARCHITECTURE STUDIES
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June, 1990

Signature of the author ______________________________
Jian Zhao
Department of Architecture
May 6, 1990

Certified by ______________ Vale ______________
Lawrence Vale
Lecturer
Thesis Supervisor

Accepted by ______________________________
Julian Beinart
Chairman
Departmental Committee for Graduate Studies

TOWARDS A HYPERMEDIA APPROACH OF DATA ORGANIZATION IN BUILDING-MODELING CAD SYSTEMS

by

Jian Zhao

Submitted to the Department of Architecture on May 11, 1990
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Architecture Studies.

ABSTRACT

The development of architectural CAD systems implies a trend of using a comprehensive building model as the storage space for all relevant data about one design project. Instead of a set of drawing files, a building is represented as a collection of building object data and their relationships. In order to create and manipulate this complex organizational structure, we have to empower the CAD system with a more sophisticated manipulating tool than it now possess. Hypermedia, as an associative way of organizing and presenting information, is able to support the practical requirements of designers.

The thesis approaches the idea from two aspects: one is to represent a building model in a data structure, based on hypermedia nodes and links. During this authoring process, we should create a hypermedia structure which can not only cope with the original building data structure but also provide the structural basis we can use in inquiring for building data in the design process. The second aspect is to browse for building information in a model embedded in a hypermedia structure. The possibilities it provides can go far beyond the constraints of conventional organizational methods. The way we access building data can be greatly broadened. The flexibility and interactivity of hypermedia are leading us to better design environments, with more machine power and intelligence being brought to architectural CAD systems.

From the viewpoint of system analysis, the ideas and methods, together with the potential problems the thesis discusses will be of value to the real practice of creating and managing this kind of data structure in future architectural CAD systems.

Thesis Supervisor: Lawrence Vale
Title: Lecturer

To my grandmother.

If it was not for the constant support from Jim and Hong,
I could have never made the accomplishment.

Thanks to Jenny for accompanying me during the frustration and anxiety of this semester.

Acknowledgements

I would like to express my deep gratitude to the following people:

Professor Gary Hack of his advice and enlightenmentwhen serving
as unofficial Thesis Supervisor;

Professor Timothy Johnson for his guidance and critics
in the whole process of thesis thinking;

Professor Frank Miller of his precious time for reading the thesis
and providing helpful comments;

And the peoples at English Writing Certer
for their patience to proof read the thesis.

Table of Contents

Chapter I: Introduction

I.1: An emerging trend of architectural CAD systems

During the early nineteen eighties, the development of architectural CAD systems depended on the adoption of features from older mainframe computer systems. The established turn-key vendors, faced with the rising popularity of PC-CAD, had focused on bringing new versions of their software to the PC platform, and PC-CAD vendors copied those high-end features and sold them at low-end prices. To some extent, this imitation and adoption process hindered the revolutionary development of new features and functions. And most of these CAD systems are designed to conduct generic design work, mainly targeted at various engineering fields. These features tended to be either too general or too simple to meet the specific demands of architects.

These CAD systems, including those tailored specifically for architects, are basically electronic drafting boards. Traditional hand-made drawings are transformed into digital form with the aid of a computer. Though some editing and organizing capabilities are used, such as copying and pasting, the general approach to organizing project information stayed the same. Each drawing on the digital board is saved as a file. Different drawings are stored in different files, and they can be printed out just like the sets of drawings we stored in a flat drawer. The disks are used as the substitutes for physical storage space. When we want to check some data, first, we need to figure out which drawing hosts the data, and then we look for the drawing file in the disks. We open the file on the screen and navigate through the parts of it to locate the necessary data. This is exactly the process we followed when we were using paper drawings as the storage media for design. We had to remember the data resources by some standards, browse through the drawings in a drawer or a binder, and pull out or open the drawing containing the required information. This kind of data organization method in CAD systems still heavily relies on human minds to do the organizational work, such as querying and retrieval. Computers are only used as the memory boards for various kinds of symbols.

The stagnation of architectural CAD development came to an end in the last few years. One emerging trend of the development is the building-modeling method of architectural CAD systems. It is changing the role of computer from a drafting board to a design medium. First, with the aid of robust, easy-to-use building modeling tools, we can build a collection of 3-D or 2-D building models. Then, integrated with the associated alphanumeric data, this collection becomes a computer building model which stores all relevant data about the project. The models like these can be further refined and improved with the development of design. We may take advantage of the data editing abilities of computers to modify the model in the design process. Once this is finished, we can take snapshots of the model as presentation or construction documents. These documents can be described by a "view" of the model, scale selection filters, and an overlay of annotations, such as text, dimensions, and targets. They are representations of the model from different angles and at different stages. They are the graphic or text reports of the building design.

The potential of integrating design and production within an architectural CAD system has long been a fascinating idea to architects . The building-modeling method is leading us closer to this potential:

1. Instead of symbols, architects can directly manipulate building objects like walls, slabs or doors in the design process. These object-oriented abilities make it possible for architects

to think directly with building elements instead of with the symbols representing these elements. This is closer to the way we conduct building design activities, which are basically the manipulation of various building elements and the definition of their relationships.

2. The building-modeling method integrates design and production into one consistent process. The data we input to construct the building model can be transferred to data for production documents. This transaction process can be done easily, and the system improves the consistency of different representations of the building. With this method, we construct the designed building twice: once on the screen, and once on the construction site. For architects, the making of production documents becomes an integrated part of constructing CAD building models.

3. The graphic representations of design are associated with qualitative and quantitative data through the inherent integrated database in the system. In fact, the integrated database is a consistent part of the building model; it is constructed into the software design of the system. We do not have to create the links between graphic data and alphanumeric data by ourselves; this is done automatically by the computer.

4. The data unit we can manage is object-oriented. Instead of drawings, which contain groups of building object data, we have access to the separate object data directly. So the management of building data is based on each object of the building. Drawing management will be replaced by a comprehensive concept of data organization for building-modeling CAD systems.

Several newly released architectural CAD systems have followed this trend of building-modeling methods, in which the designed building is represented as a coherent set of data instead of sets of drawings. In "Architrion II", this is represented by an integrated system consisting of three modules: "ArchiDesign" allows designers to construct a building with various building elements, using its robust 3-D solid modeling tools; "ArchiDraw" provides the drawing tools to further delineate 2-D drawings projected from the solid model; "ArchiList" creates comprehensive lists of quantities based on the 3-D solid model, and can be used as the basis for other quantitative analyses, like cost estimates. During the construction of the CAD building model, we can bring in some pre-defined building elements from the integrated 2-D and 3-D libraries; the characters of these imported elements can also be reflected in the table of quantities generated through "ArchiList." In another Macintosh-based architectural CAD system, "ArchiCAD," this feature was further enhanced in terms of its abilities to consistently represent building model data. One package of a building-modeling system can store almost all the graphic data and part of the alphanumeric data about a design. This kind of system has taken a great step forward using the 3-D solid model as the coherent basis for organizing building model data. Conventional presentation drawings, construction drawings, and bills of materials can be derived from this solid model, and together with these construction functions, they form an integrated design and production system.

Since these systems imply a more comprehensive and intuitive way of managing building models, the mechanisms involved tend to be more complex than those of the drafting-oriented CAD systems. Unfortunately the abilities of today's CAD systems are still at the level of production-oriented drawing management. Building-modeling CAD systems have to be accommodated with a new organizational structure and a set of management tools based on today's advanced information management technology. This is the area this thesis addresses.

I.2: Thesis Organization

There will be basically two parts of the thesis. In the next chapter, I will focus on the theoretical possibilities of incorporating hypermedia into the data organization of a building model. As an integrated system for both design and production, building-modeling CAD systems should be able to manage building model data in a comprehensive and sophisticated manner. The problems I will explore are: how we access building data for various purposes of design practice, how the available features are limited in accomplishing these functions, and how we can break the constraints of multi-layer drafting systems in terms of their abilities in managing these data.

I will introduce the concept of hypermedia, the associative way of organizing and presenting information, to examine its potential in this particular environment. I will try to justify its application in terms of design thinking, communication with other design participants, perception of building representations and the application meanings in construction. The focus will be on how we approach and manipulate building model data in the design process, and consequently on how we should manage these data effectively and conveniently. I will also examine the available CAD features in terms of their ability to serve the practical demands of building data management.

The second part of the thesis will explore the ways we incorporate hypermedia in the organizational structure of building model data. In Chapter III, I will start with the problem of how data are constructed in representing building object information. The organizational structure of hypermedia is different from conventional data systems. The nodes and links of hypermedia can be defined to represent various building object data. I will deal with the problems of how building object data can be represented as hypermedia nodes, how those relational data can fit into forms of hypermedia links, and how these two kinds of data are associated together in forming a hypermedia-based building model. The discussion will touch on a representational model of windows in a single family house, which has implications for the data structure of other kinds of projects.

In Chapter IV, I will discuss the applications of the hypermedia-based building model in browsing for information about the design. There are many ways for us to access one piece of information, and these can also be managed by the mechanisms built into the model for the convenience of users. I will examine the demonstration model built in Chapter III following some possible browsing processes in design practice. And this will also form the conclusion of the thesis. After a brief review of the ideas and methods we explore in chapter two and three, I will analyze the problems which may occur in a real design environment with a hypermedia-based data structure. I hope that the discussion of these problems, and the ways of incorporating hypermedia, will be helpful to the practice of creating and managing this kind of data structure in future architectural CAD systems from the viewpoint of system analysis.

Chapter II: Hypermedia and Building Data Management

II.1: A problem with the data organizational structure of CAD systems

Information organization in building modeling CAD systems is fundamentally different from those systems based on overlay drafting techniques. We do not have to segregate data into separate drawings or other separate groups. Conceptually, the building model is the host of all data about the project in every phase of design. Since production is no longer the only purpose of the system, the requirements of design will play a more decisive role in conceiving the functions of the system. We can retrieve information from a building model not only as production documents, but also for design operations like comparison, analysis, evaluation and presentation. To conduct these operations, drawings are not necessarily the only representational form. It should be possible for us to retrieve data of building models in all kinds of representational forms; this is similar to the relationship between one physical object and its various representations We can have pictures of the object from different viewpoints and at different scales, we can have scripts to describe the object textually, and we can also use equipment to record our visual or other sensory experiences of it. We can imagine, in the future, in the same way a document is now taken into a postscript printer, a CAD database may be taken into a rendering-type system to produce construction documents, or it may be generated as a movie sequence through an animation system.

However, the relationships between a building model and its various representations are far from defined as simply as that. Conceptually, these representations can be taken separately from the building model. But in reality, they form the integrated parts of it. We can only perceive the model through these representations, such as plans, perspectives or spreadsheet tables. And no matter what kind of media we use, we have to follow the conventional ways of presenting building data. This is especially meaningful in regard to construction documents. The conventions of architectural drawings have to be used as the basis of communication. So, on the one hand, we are freed from the constraints of drawings as the only and universal building data container, and, on the other, we have to adopt these conventional forms of representation to conduct both design and production work.

Another significant difference in data structure between drafting and building-modeling systems is the way relational data are represented. In the drafting systems, the representation of connections among different building objects is largely limited to the scope of one drawing. The power of a spreadsheet lies in the formulas which reveal the relationships among large amounts of numeric data. Similar to this, the most valuable asset of a building model is not the data stored in it, but the ability to dynamically representing the relationships of all the relevant data about building design. Architectural design is basically the manipulation and definition of complex relationships of various building elements. The data we input into the building model should be constructed into the logical relational network. And this should be not only reflected in one single drawing, but also among each piece of object data within the building model. The relationships should be constructed not only as an inherent part of the data but also as an open-ended structure which can be modified and redefined as the development of design.

Compared with this, today's CAD systems have failed to identify the interconnections of building data which allow us to recognize the links and similarities between different kinds or pieces of building objects, the most fertile background for architectural design practice. Obviously, the development of a building-modeling CAD system will tackle this problem more seriously. Since today's CAD systems still manage building data on basis of files, we can partially solve the problem with some housekeeping methods at the cost of efficiency and flexibility. A building-modeling system requires every object data of a building model as an object unit by itself be associated into one coherent system. It is hard to imagine that we can manage the complicated relationships of object data simply by opening, closing or merging different files on disk. That means that we need will new power, which fortunately has come into being with the development of information technology. Hypertext and its broader sense concept, hypermedia, provides us with a conceptual weapon to solve this problem.

II.2: Hypermedia: a choice of design practice

II.2.1: The concept of hypermedia and its application in information organization

Hypertext can be simply defined as the creation and representation of links between discrete pieces of data. When this data are graphics or sound, as well as text or numbers, the resulting structure is referred to as hypermedia. It is simultaneously a method for storing and retrieving data. Hypertext incorporates the notion of linking pieces of information, allowing users to navigate through a network of chunks of information. Information is provided both by what is stored in each node and in the way that information nodes are linked to each other.

It is not difficult to understand the significance of hypermedia to the organization of the building data stored in a building model. Each object can be regarded as a data node within the building database. If we can create the structural links with other piece of data at the same time or after we create and input these data into the building model, we can embed these representations into a structural network, which allows us to take advantage of the flexibilities of hypermedia in conducting further design or production operations within the CAD system. So ideally the process of generating representations from the model and manipulating these representations can be converted to the operations of inputting and retrieving data in a hypermedia database structure. In practice, these two kinds of operations will accompany us in every phase of design.

The hypermedia approach for organizing data within the CAD system is different from the artificial intelligence method of CAD. The AI method tries to represent the designer's knowledge in this field and the real circumstances of design in a form that enables machine to perform sophisticated reasoning, comparing and selecting, while, hypermedia is trying to augment human cognition by providing a dynamic storage medium for project information. Although these facilities alone cannot transform a drafting system into a real design system, hypermedia approach does enable us to use the data stored in these representations more effectively and intuitively, which will finally facilitate a better design environment.

The purpose of data transformation within CAD system is communication; the data are transmitted from machine to designers, among designers and between designers and other participants. They can be transformed in one of the following ways:
1. The data is accessed by designers through retrieval and query mechanisms in the system. After the process of perceiving, evaluating and editing, the modified data is returned back. The cycle may happen many times in different forms, following different paths. The

question for us is what kind of mechanism designers need in accessing data for this review cycle.

2. The data is transmitted among all participants other than designers. In this process, various ways of presenting the same piece of data should be employed. Different participants may have different requirements regarding the way data is presented.

3. The data is presented to people including designers who depend on these representations to perceive the design model; it concerns the perception of the human mind when dealing with building information. The issue is, put the specific professional concerns aside, how people perceive architectural design in terms of the way they access data.

4. During the processes of producing construction documents and supervising construction, we may have another set of requirements for the way the building data are represented. And this will become the main justification for the adaption of hypermedia in CAD applications.

II.2.2: Manipulating building data in the design thinking process

A designer's selection of data largely follows professional logic, but this does not mean the data selection path will be either definable or predictable. In fact, the ways of inquiring data often end up as a maze. The problem not only lies in the complicated relationships of building objects, but also, and maybe mainly, stems from the required flexibility of design practice. Aside from the unpredictability of job commitments, unpredictability is also one of the characteristics of that design problems. This affects the whole process of problem-solving, and may bring in new problems at the same time. To track these nearly random "trouble spots", we have to be able to set up links of data at any point as well as go everywhere in the building model.

Let us further explore the process by which designers discover and solve design problems and at the same time trigger new problems. Suppose we are working on a housing project with an important facade facing the street. After finishing the elevation design, we convert it into a view from street perspective; then we find that there seems to be something wrong with it. Where is the problem? We switch between different views of the same elevation, or go back to plans and sections, or maybe, in the future, will perform an animation sequence of walking down the street or into the building. Is this a problem because of awkward geometric composition, or inappropriate structural types, or fail to acquire the needed visual effects and so on? To target the problems, we may need to go through various representations of the elevation.

To make a simple picture, let us say that it is because some windows look too large compared with nearby buildings, and this causes a visual imbalance of solid and void in the facade. Then we need to move to the representation on which we can directly edit the windows, such as a detailed elevation, or object library. We run to the library to change the size of the window, while, after looking at several other parts of the design, we find that the same window figure used there have also been changed and the modification is clearly inappropriate. We go back to the library to undo the change and choose or create another figure which supposedly will fit into the elevation. When we go back to the elevation again for a while, we suddenly realize that the window may be too small to provide enough natural light for that deep living room behind it. This is a new problem. To solve it, we may go to a representation to perform a light simulation of the living room at a certain time or date. If not, we will have to try to find some other solutions. The cycle of seeking and solving design problems requires us to shift between various representations of the model. The possibilities for browsing through these data are numerous and constantly change.

Hypermedia consists of exactly this kind of linking structure, which can potentially assist us to access those related data during the design process.

Architectural design is a complicated thinking process of analysis and synthesis, which involves constant comparison, evaluation and selection. Every step a designer takes is based on some related data. What mostly happens is that, when we work on one aspect of the design, we have to constantly reference to other related data. Design is partly an inquiry process for related data. And designers should have a complete and comprehensive understanding of the building model in order to make any decisions. We cannot understand the whole picture of a building model if only provided with some isolated representation of it. One representation can only convey the information of one aspect of a building model. But each design decision will have a broad impact on the whole model. To compensate for the incomprehensiveness of a single representation, we have to constantly check with the data representing other parts of the building.

II.2.3: Inquiring building data for design communication

Architecture practice is a multi-disciplinary effort. The subdivision of labor is becoming finer and finer. From space planning to facility management, numerous design participants must become involved. Some tasks are predictable, some may never be thought of at the early stage of constructing a CAD model. Each area of expertise requires own data inquiring method. The data structure in CAD system has to be flexible to meet these practical demands. The data stored in a building model has to accommodate the diversified and constantly changing needs. For example, if we have several rooms in the model, at first we may only have to provide the envelop data for the interior designer,and the HVAC engineer. We set up the retrieval links for the convenience of these two professions. But we find this is also the working base for the electrical engineer and the cost estimator. They also need retrieval links for them too. There should be several ways of retrieving the same piece of data. A building model should be a friendly space in which all these design professionals can work together.

Architectural practice is classified as a producer service business. The products generated must pass through a series of producers' hand, in order to reach their final consumers. A design product has to face a diverse spectrum of spectators or users. All those people may have an influence on design. The skills building designers present the design ideas and the manner in which they organize and present these skills become the means of constituting the practice. The demand for the service is confined more and more to specific features of the product or particular stages of the design process. Many jobs are limited to a feasibility study, program development, schematic layouts, or construction document. Often the performance of these services will lead to further work on the project or another job for the same client. Once designers procure a job they do not always know the tasks the project will involve. This unpredictability can also result from the client's bureaucratic system of management. The final product is not necessarily the construction documents. Presentations at each stage of design become common phenomena in today's practice. The feedback we receive during these presentations is decisive to both project and design development. In the CAD environment, these presentations may take some revolutionary forms depending on the new tools we have. We may use computers to demonstrate the building model in front of these people, or we may simply send them a copy of the model for them to review the design by themselves. So the problem of accessing data may expand to groups of people other than design professionals.

Generally, today's design projects are becoming larger and more complex. They will have a broader influence and affect a larger group of people. Government agencies, communities groups, financial supporters, public interests groups... every one of them may participate in the design process. Design presentation is an interactive communication process. It requires all participants to act and respond actively. The issues we prepare may not be of interest to the audiences. And the issues the audiences would like to raise may not be the ones we prepare. The audiences, who usually play a dominant role in design development, should be encouraged to express their ideas about the project. At this time the design tool we have can often be used as a presentation tool with its super graphic visualization abilities. In order to provide such an interactive environment between the building model and audiences, the computer should be able to perform some flexible and interactive function for manipulating data. During this process, the machine should allow us to take into consideration the understanding levels and points of interesting of each specific audience. What we have in the building model should not a canned slide show, but a set of data ready to be accessed in different ways with some straightforward operations. With the aid of hypermedia tools, it is possible for us to do this simply by clicking on several buttons.

II.2.4: Spatial perception and building model representation

The physical characteristics and properties of the building model data in computers can not be directly perceived by human minds. The only way to access a building model is through its various representations. These representations act as the medium between our perception and the building model. Every representation reveals some aspects of the model, and all of them together help us form a complete mental image. They are the only sources we can depend upon to perceive the characteristics of a designed environment.

Especially in design presentations, the designer tries to convince the audiences that these representations are the reliable basis to predict future reality. But as we will see, even if every possible representation is available and included, (though it is impossible), it does not guarantee our perception can completely reflect the model or the built environment. A large percentage of this transaction problem is decided by whether these representations are organized and fed in the way our perceptual ability digests and absorbs data about the physical environment.

Aside from each person's specific professional background, we share many similarities in perceiving the physical world. The transactional process by which we experience the world is called perception, and while scientifically, perception is in the domain of psychology, the process is open to all of us as perceivers. How we perceive the physical environment is crucial to the problem of organizing data to represent the environment. The perceiving process is partly the way we receive data through our various senses, such as vision, smell, and touch. For the same physical object, the way we receive the data, contents, sequences, media, can make considerable difference in our perceived mental image.

Among all perception activities concerning physical environment, spatial perception is the most frequently encountered activity in the field of environment design. There are numerous media forms we can use to convey data for spatial perception, such as sound. We can perceive a space quite vividly simply by listening to the sound effect in the space. But it is difficult for us to imagine the physical details of the space. Language is also helpful at depicting a space. It may reveal some meanings which cannot be presented by other media forms. But it tends to be too abstract and cannot be separated from subjective factors like past experience and imagination. Comparatively, speaking visual images can be much more effective in conveying data for spatial perception. A picture is better than a thousand words. Spatial perception is largely done through some visual cues, such as surfaces, edges, and various perspectives. In perception of the physical world, one medium form may not be explicit in presenting all aspects of the environment. One form of media, such as graphics, may have a dominant position, but it must cooperate with other forms to convey a complete idea. The potentials for retrieving data from building model in various media forms will be very important to architects. Hypermedia, combined with current media technology may bring this into reality.

The multi-media approach of presenting data does not make sense if the data is not associated in a perceivable way. Gibson in 1966 explains the experience we have in perceiving a physical environment:

> "...One vista leads to another in a set of continuous connected sequences. Overtime, as the individual moves about the house, the street, the town, and the country, the sequences come to be perceived as a scene, and the fact that the transformations all make a group becomes evident to him. The individual is then able to find his way from place to place, but more than that, he is able to see one place behind another on a larger and larger scale. He is then geographically oriented. Even when he is shut into the room he is able to apprehend the house, the street, the town, and the countryside in relation to the room...".

Spatial perception is not only recognizing certain spatial images or characters, but more important, the relationships existing among them. If each image can be thought as a representation of the model, then, in order to perceive the model thoroughly, we have to understand the relationships between each representation. Hypermedia implies is a way of presenting data associatively. When one piece of data is presented, the relationships with other pieces can also be revealed at the same time.

II.2.5: Applications in production and construction

In today's architectural practice, producing construction documents and supervising construction bring in more than half of the incomes of architects. The practical requirements of these work possess significant meanings to the performance of an architectural CAD system.

As we mentioned earlier, the construction documents will become the natural results of a well-finished building model. But this does not mean the transaction from a design model to the construction documents can be a effortless process. In fact, in order to cope with the conventions we adapt in representing a design for construction, we may have to include information from different sources into one document, or to distribute the information from one source into different documents. For example, in a typical detail drawing, there may be many reference nodes revealing the relations of this representation with others, and the background of this drawing may come from a drawing, which simultaneously provides background for several this kind of detail drawings. So during the process of producing construction documents, there are needs of the flexibilities of accessing different pieces of information for cross reference. The process may be similar to the way we produce a text document by merging chunks of text from different documents, or segregate one document into several pieces for clear representation of the information. The potentials of hypermedia in enabling us to manipulate building data flexibly and in a object-oriented fashion can be meaningful and very attractive to architects.

In the future, construction documents may take a totally different form in terms of the media conveying the information. We do not have to transfer a CAD building model into a set of construction documents for construction; we may directly construct a building based on the CAD model. Let us imagine that we can bring one terminal to the construction site. Under this circumstance, it is possible for us to take a different approach of representing a building model. For example, all related information about one building object, such as a room or a story, can be accessed from this object. In the process of supervising construction, we can immediately access the information for different inquiry needs from different people: an electrician, a carpenter or a representative of the manufacturers.

II.3: Hypermedia potentials in current modeling systems

The discussion about building data transaction in design processes explains the incentives for us to adapt hypermedia as a conceptual framework for organizing building data. Just like any other developments of CAD systems, a hypermedia-based building model is not an utopian idea without roots, it will be the direct off-spring of today's data organization methods. As a merging point of information technology and CAD application, it implies a promising bridge leading to an intelligent CAD system for architectural design. To some extent, incorporating hypermedia organization methods into a building modeling system can be regarded as an improvement for the system. It will be the expansion of the organizational facilities, which can perform more comprehensive and advanced features in organizing building model data. Since these data are strongly relational and associative, the prototype of this kind of data organization already exists in the current architectural CAD systems. We will examine these functions from the following three perspectives:

1. What kind of functions are available in the state-of-the-art CAD system.
2. How different can the features be in the future building-modeling CAD system.
3. At what point for hypermedia to be merged into a practical CAD environment.

To provide a kind of practical setting for the discussion, we will use "ArchiCAD" as a representative of today's advanced CAD systems. It may help us to understand the problems in a general way.

II.3.1: "ArchiCAD": an architectural modeling/design package

As early as 1983, "ArchiCAD" was implemented on the Apple,"Lisa." After 7 years' development based on the interaction with its client and dealer network, it became one of the sophisticated architectural CAD packages on Macintosh computer. It adopted a practical approach and interface design which are extremely beneficial to architecture designers. Fundamentally, it is different from most generic CAD systems in three aspect:

1. It is a software package exclusively designed for architects. So, unlike those general purpose 2D/3D drafting systems, we do not have to use the macro command language (like AutoLisp in AutoCAD) to construct those specific features for AEC needs, such as architectural symbol libraries or some editing functions. It can reduce the commands of composing parallel lines, cleaning intersections, filling area and projecting to 3D into the simple operations of selecting wall tools and drawing. The architectural features like wall, slab and roof are treated as the basic commands of the system, and the construction of a building model can be accomplished directly using these commands as opposed to having to assemble macros of drawings of abstract geometric elements, such as

rectangles, circles and squares. This kind of building-object-oriented method of design and modification is similar to the object-oriented structure of hypermedia.

2. It is an architecturally tailored 3D solid modeling software. It calculates all the complex 3D operations which can represent every aspect of a building model. Unlike a surface modeling program, which overlooks the solid character of building objects, "ArchiCAD" allows us to take representations of building elements realistically in terms of their true physical features. We can see through openings in solid walls and roofs, unified walls and slabs that have continuous exterior surfaces for a story-by-story constructed building. We can cut real sections of the solid building model to reveal every hidden part of the model. We can calculate the volume quantities of both solid and void objects in the model. For a designer, the building model in the system is really equivalent to the building to be constructed on the physical site. We cannot apply these building data organizing methods used in multi-layer drafting system. There is a demand for a new way of data organization.

3. It integrates 2D drafting, 3D solid modeling and bills of materials into one package. During early design phases, we use it to quickly construct and visualize design in a robust fashion. After we further refine the design ideas and represent them as conventional drawings, we make presentations with the help of its various drawing and drafting functions. Construction documents result naturally from the building model with integrated database functions. Most of the information about the design can be included into the file which contains the designed building model. So the building model in this system can be regarded as a comprehensive database for all building model data, it possesses the potential to accept hypermedia as a new way of organizing these data.

II.3.2: Data representation

During the process of inputting data into a building model, data about building elements are categorized into six types: walls, doors, windows, fixtures, slabs and roofs. Both the whole project and each type of building elements can be represented as an object representation. For example when we construct a section of wall in a building model, we are prompted to a dialog window to decide all characters of the wall, its name, materials, finishes, dimensions, etc. These wall-related data will be associated with the wall graphics we input into the model, and each time we highlight this wall, we have a chance to access this related information. They will also be presented as part of the related information in the scenario of the whole project. At any time in constructing the building model, different representations, such as plan, section, elevation, perspective and bill of material, are available when required. Since every representation is derived from one building model file, the relationships are automatically created and maintained during the processes of construction and modification.

In terms of representing building data in object-oriented fashion, "ArchiCAD" may have only set up an obscure skeleton for the organization structure. Between one project and the building elements such as windows and doors, there are numerous levels of object data existing in a building model. They form a delicate relational network. If every element of one room, from roof to floor, can be represented as an object, why cannot the room itself be represented as an object, and why cannot the floor of all the rooms also be represented. If each window can be represented as an object, why not all the windows in the room, in the floor or in the building. In fact, in a complete object-oriented data representation structure, any identifiable entity or group of entities should be represented as an object. Just like a door contains all related data such as its name, type, dimension, and position, a room object should be able to reveal the information like the room's name, number, volume, area, enclosure, and fixtures.

The object representation of building elements in "ArchiCAD" is exclusively directed to construction purposes. We access the wall setting dialog window mainly because we are constructing or editing a section of wall. Virtually, the purpose of these settings is not to represent the associative data of the building objects. Its representation features are only the by-products of the construction process. For example, when we need to input a window into a wall section, we first decide all features of the window by selecting through the window library, then insert it in the designed position. After the insertion, the window still represented as an identical one used in other parts of the building. The computer only remembers it as one item from the window library.

II.3.3: Data browsing and inquiry

There are two kinds of operating tools in an architectural CAD system: the view control system and the integrated database. The view control system is the navigational mechanics we use in controlling viewpoints, scales and contents of each representation. The integrated database is the facility to manage the connection between graphic and alphanumeric data of building model. In building data browsing, "ArchiCAD" shares many similarities with other systems. We will discuss the problem in a more general sense.

Data browsing through the view control system basically defines the contents of representations. Although many improvements have occurred-such as the change from operating with 2-D drawings to manipulating a 3-D object and from freely assigned layers to the standard layer system-the basic concepts remain the same.

With the aid of the coordinate system, we can choose any standard viewpoints or define a specific viewpoint to make a projection of the model. These projections can be either used as a view to examine design or as a base to produce production documents. Each projection can be regarded as a representation. In a solid modeling system such as "ArchiCAD", when we cut a horizontal section of the 3-D model, we see a plan view of the building. When we simply choose view points to look at the sides of the model, we have a set of elevations or some 3-D views of the building. Zoom-in and zoom-out have similar effects in organizing the building data. We can control the view scope and detail level to segregate information. A wall detail and an elevation can convey totally different information, while the zoom-in or zoom-out operation can set up the links between them.

Multi-layer system is the typical way of organizing information in draft-oriented CAD system. For each drawing we can input different kinds of information into different layers. And then by turning on or off each layer we can show or hide the information at our intention. One of the popular issues in CAD development is the standardization of layer system. Although file transformation problem may be the major concern for standardization, to provide an object-oriented tool for organizing building model is also a strong incentive. In "ArchiCAD", each layer is endorsed with a specific label. This label can act as the handle of the information stored in this layer. Also layer is not only holder for 2-D object, we can put 3-D information into different layers. This may not fit with the word "layer" which itself is a 2-D concept. In fact the concept of layer is becoming an information holder with specific usages and can be accessed separately or collectively. Each layer contains a sub-set of information. By turning on a particular layer, we can see a particular representation of the model.

We can browse through the building model by making certain layers visible and others invisible. We can retrieve certain kind of information by referring to the name of its label, like we pull the handler of a drawer. Also these layer manipulating tools can be used to connect any part of the sub-set data together. When we have several layers visible at the same time, we visualize the links among these different representations of model. If we bind together several "handlers" of layer, the links among them are set up.

The concept of integrated database started from intelligent graphics, which associates a graphic entity with text. When we instruct computer to draw a circle, the word "circle" is stored with the graphic. So next time when we refer to this graphic, computer automatically understand it is a circle. If we combine intelligent graphic figures with database records together, they constitute an associative figure or group. So when we add intelligent figures to building model, we are also building the graphic database. Door schedule for a drawing can be derived from the doors we include in this drawing. One of the unique features of integrated database is that it can be manipulated using a combination of graphic and text-editing techniques. We can create a building model, modify it as necessary, and finally, we can extract the needed data with various retrieval and query methods. This process is greatly simplified in "ArchiCAD". We can browse between a text table and a drawing simply by choosing a menu item. The modification in a drawing can be reflected in the table automatically.

But the same as other CAD systems, the browsing paths are strongly directional and limited. They are lack of the flexibility and versatility required by design practice. Combined with its insufficient object-orientation functions, we are constrained by a stereotype of organization structure. Although we may find some features with the taste of hypermedia, basically they are still based on a conventional organization method. In next two chapters, we will continue this topic by exploring the solution based on an incorporated hypermedia building model.

Chapter III: Representing CAD Building Model Data in a Hypermedia Environment

As we discussed in Chapter II, incorporating hypermedia concepts with the data organization of CAD systems is becoming a great possibility with the development of the building-modeling systems. From the viewpoint of design practice, we require interactivity, flexibility and machine intelligence in the manipulation of building data. The managing abilities of today's drafting systems are faced with a complicated and sophisticated CAD building model. In the next two chapters, we will explore the possible representational structure of data for building-modeling CAD systems based on the concepts of hypermedia. The focal point is how hypermedia, as an information authoring and presenting tool, can be incorporated into data organizational structure of architectural CAD systems.

In an architectural CAD system, hypermedia provides not only a flexible way of browsing for information about a building model. But it is also an extremely flexible data representational environment. A CAD building model can be represented in numerous ways. By identifying the contents, scales, media types, presentation purposes or phases of design, we can retrieve various representations of the model, such as a plan, a mock-up drawing, a perspective, or a text document. In the future building-modeling systems, the forms of these representations will largely remain the same due to the conventions they involve. But the organizational structure of the data contained in these representations will be fundamentally different.

There are three components of the data structure for meaningfully representing data about building object:

1. There are data representing the building object itself. All building elements—such as a handle, a door, a room, a story and a building—can be represented directly by this kind of data-object. Although there may be several representational forms for each of them, they all belong to one data-object. Here the term "data-object" refers to that body of information which is used to represent a particular building element.

2. Another part of the structure is the attribute values of these building objects; for example, for the data-object of a door, we may have a combination of numbers like "7'0 x 4'6" to represent its dimensions, a text string like "D#3" to represent its name, and a graphic symbol to display the relations between it and other data-objects, like the section of wall which is holding the door. A value of a data-object can be either an "atom", something like the color "white" which can not be further defined as a data-object in the architectural senses, or it can be a collection of other data-objects, which can be called the elements of this data-object: for a data-object of a door schedule, for example, all types of doors in it are the elements which can be further defined as data-objects.

3. For a data-object, we may have a set of attributes which defines its characters or its relations with other data-objects. "Color" can be an attribute of a door which represent one of its visual characteristics. The attribute, "components", reveals its connections to a series of data-objects such as the frame, the board, the leaf and the hinges. The significant meaning of the attributes of a data structure is that they reflect the organizational relations among all kinds of data-objects, rather than simply describe a data-object in further detail.

The relationships among these three components of the data structure can be represented as: "A (O) = V ." "A" stands for the attributes of a data-object, "O" is the data-

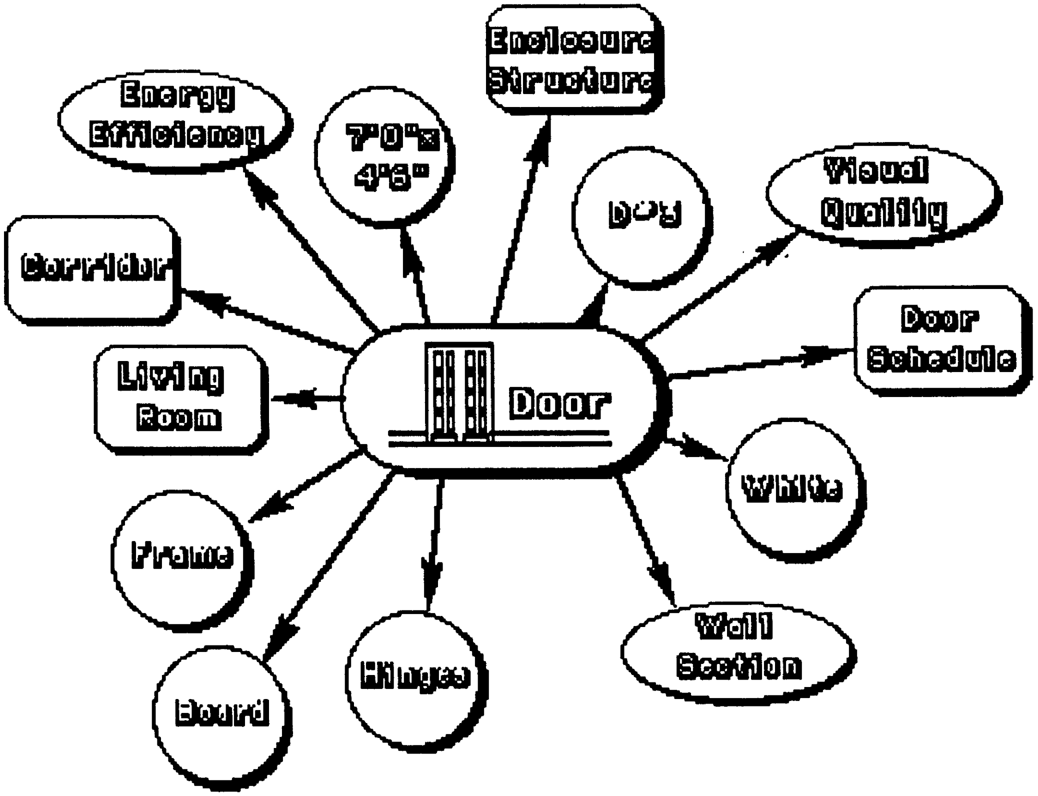


Figure 3.1: Data structure for representing a door as a data-object.

object itself, and "V" is the values of a particular attribute. From this kind of data representational structure, we can retrieve information statements like: "the dimension of the door is 7'0" x 2'6," [Dimension (Door) = 7'0" x 2'6"]," "the color of this door is white" or "the position of the door is between the corridor and the bedroom." The relations of these components can also be represented in a directed graph form. The data-object and its values (elements) are represented as nodes (vertices) and the attributes of the data-object are represented as the arcs, which are the lines connecting those nodes. Since the attributes are the properties of a data-object, the attributes with their various elements have a directional meaning. They are represented as arrows in the arcs (Figure 3.1).

The characterization of a data representational structure is not concerned with the way that such structured information can be represented in memory, but rather with the abstract properties associated with the various operations which are appropriate for our manipulation of the information. Through this kind of structure, we can represent the information more effectively and clearly.

In many ways, hypermedia is analogous to this kind of representational structure. It consists of nodes (objects) and links. Different forms of knowledge representation can be implemented in hypermedia by structuring and defining the basic types of nodes and links. So in order to create a building model which can fit into hypermedia's data representational structure, we may analyze this problem from the perspective of a building model; by representing building data with the nodes and links of hypermedia.

III.1: Representing building object data as nodes of hypermedia

A building model may contain a large number of data-objects for a practical design project. For example, in this single family house (Figure 3.2), there are approximately 100 basic construction elements, which include 38 sections of walls, 14 doors, 11 windows, 32 various pieces of fixtures, and etc. This model is only for demonstration purposes, the number of the building elements is still very limited compared with a practical design

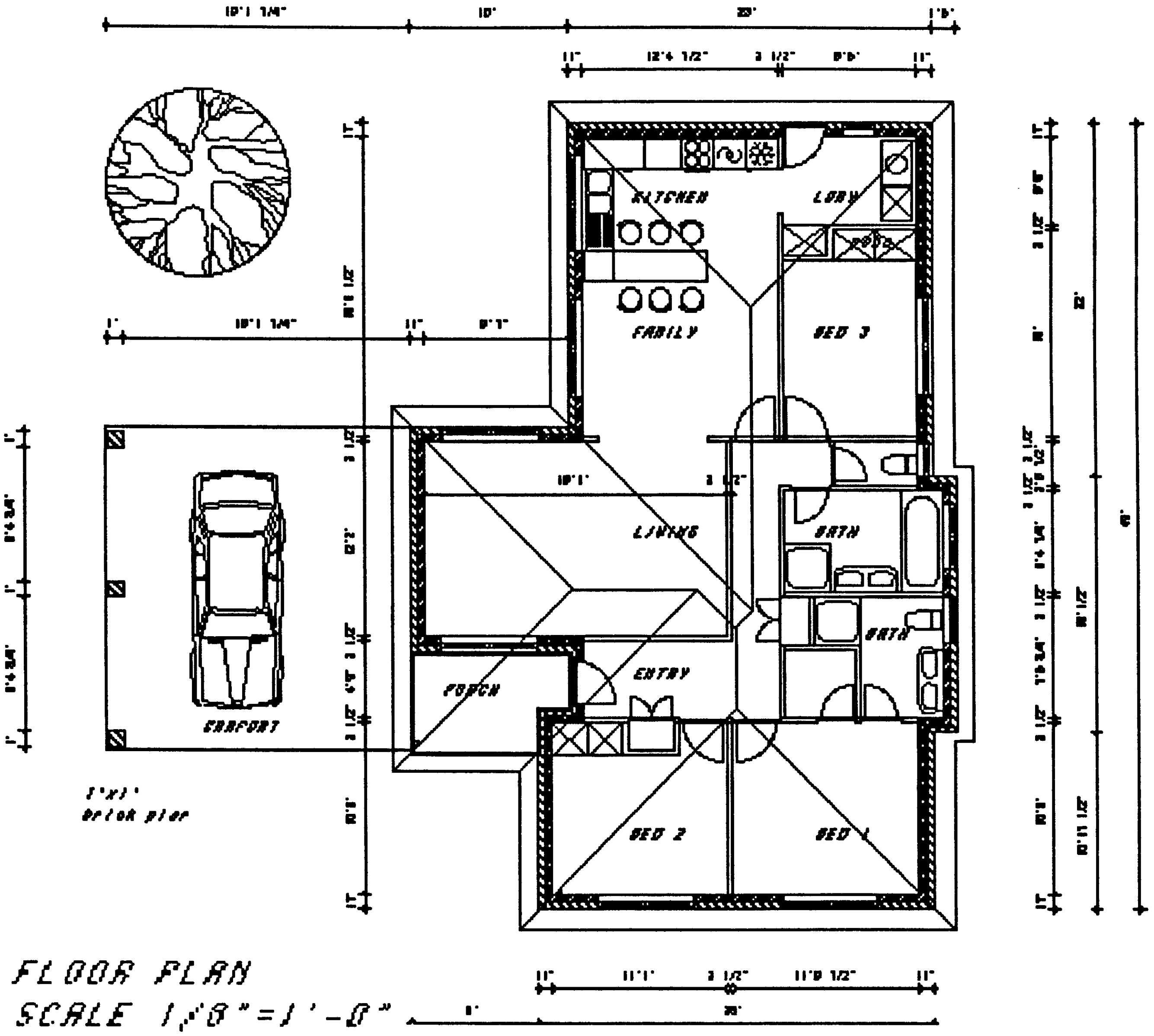
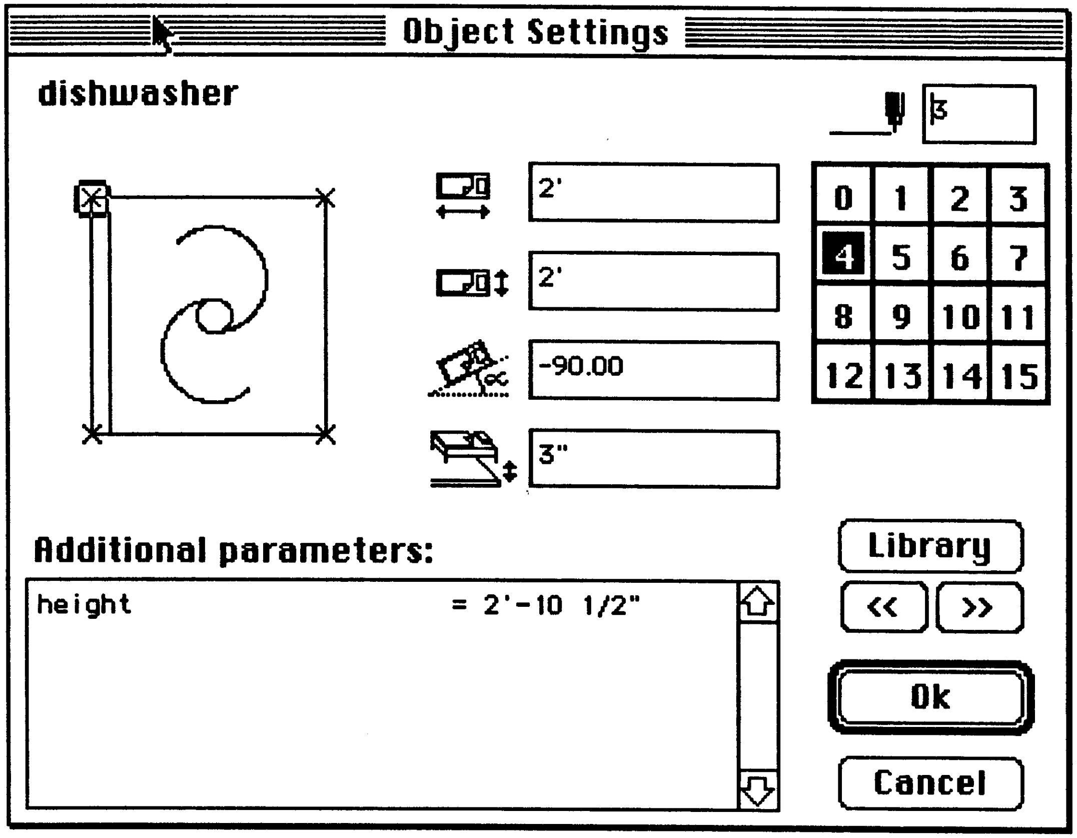


Figure 3.2: Plan drawing of a single family house

model. And this number includes only the basic construction elements, which are physically separate units, with which we assemble a building. In a hypermedia environment, the number of data fragments of a building model which can be represented in a separate data node is much larger.

Hypermedia is an object-oriented data representation environment. Every identifiable entity in a building model should be regarded as an object node. In a directed graph representing the data structure, generally every node point can be interpreted as a representational node of hypermedia. The door, each of its components and its name—"D#3", for example—can be represented as an object node. In the same house with 100 basic construction elements, we may easily identify a number of entities which can be represented in the hypermedia nodes to a level far beyond the scope of the manipulating abilities of a human mind. Simply for one room, we may have the room, the contents of it, the enclosure structures, the physical characters, the relationships with other parts of the building, and etc. If only the data-object can be defined as an entity with any architectural meanings, we can make it a hypermedia node.

In a building-modeling CAD system built upon hypermedia concepts, the main function of data-object nodes is to display building information. The way of representing information in the node form is more flexible than traditional data structuring methods in that it allows data in a variety of forms to be attached to these nodes. A data-object node of a building model may consist of a graphic, a text string, or even a sound. Every representation of the building model can be represented as a node. Some nodes may be similar to the drawings we used to have in hard copy forms; the plans, the elevations, the perspective drawings. The mixed-media nodes with graphics and text together represent one specific aspect of the building. A hypermedia node may also have some different forms, such as an animation sequence representing the experience of walking down a designed corridor. A node is virtually a universal container; all kinds of building data can be stored into it.



dishwasher

3

2'

2'

-90.00

3"

Additional parameters:

height = 2'-10 1/2"

Figure 3.3: Represenatation of a dishwasher.

A hypermedia node does not intend to be an explicit representation of a data-object. This is different from conventional representation methods. The data displayed in these nodes, such as the dishwasher symbol in the plan drawing of the house, usually are not interpreted; conventional methods assume that we, as viewers of the drawing, can interpret this symbol as a dishwasher. However, if we include this symbol in a hypermedia node, it can be taken simply as a graphic representation of something. There will be some other representational nodes which may further display the information about the symbol (Figure 3.3). One node is used only to represent one aspect of the information. The group of nodes, together with some hypermedia links, conveys the whole idea of a data-object. The knowledge interpretation of the data-object is partially manipulated by the computer instead of human minds. For a complex data-object, this approach will finally end with some better-informed users.

There is a special kind of hypermedia node which contains the organizational and functional information about a building model. It represents the relationships of data-object nodes. In the data representational structure, it may be equivalent to attributes data. There is a distinction between this kind of node and the relationships they represent. The links connect pairs of data-object nodes in a building model, whereas the organizational and functional nodes execute the procedures implied by the links represented in these nodes.

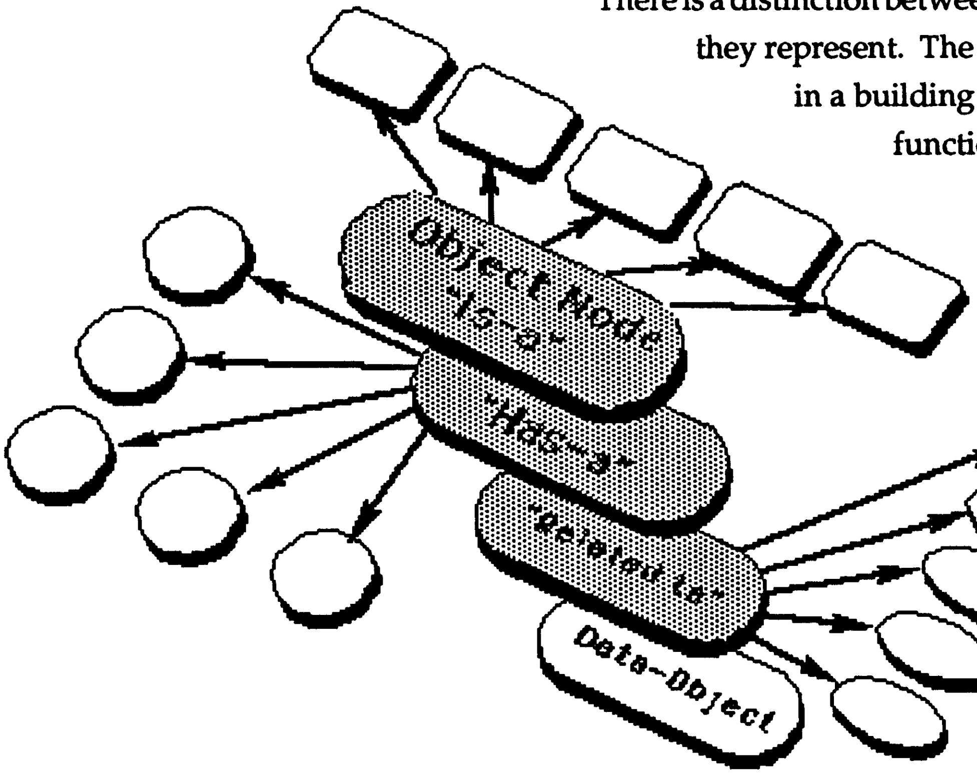


Figure 3.4: Representation of an object node.

One type of organizational nodes is the object node. In general, we can think of every node in the data structure as an object node. For the sake of clearness, we can imagine the object nodes are overlapped with ordinary data-object nodes. The purpose of the object nodes is not to display object data, but to display the attributes information about the data-object. For each data-object in a building model, the organizational links with other data-objects may be represented as a node. By sending a message to the object node, we can access the organizational links which are not contained in an ordinary representation (Figure 3.4).

Another type of special node is the index node. It usually represents a "go-to" link, which connects any two or more data topics in a building model. For example, when we look at a door schedule, we may suddenly think about checking for a construction detail. The links between them is not provided by the organizational structure of a building model. In order to use the model, we need to have control over these kinds of functional links. We can store the links in the index nodes and use them when required (Figure 3.5).

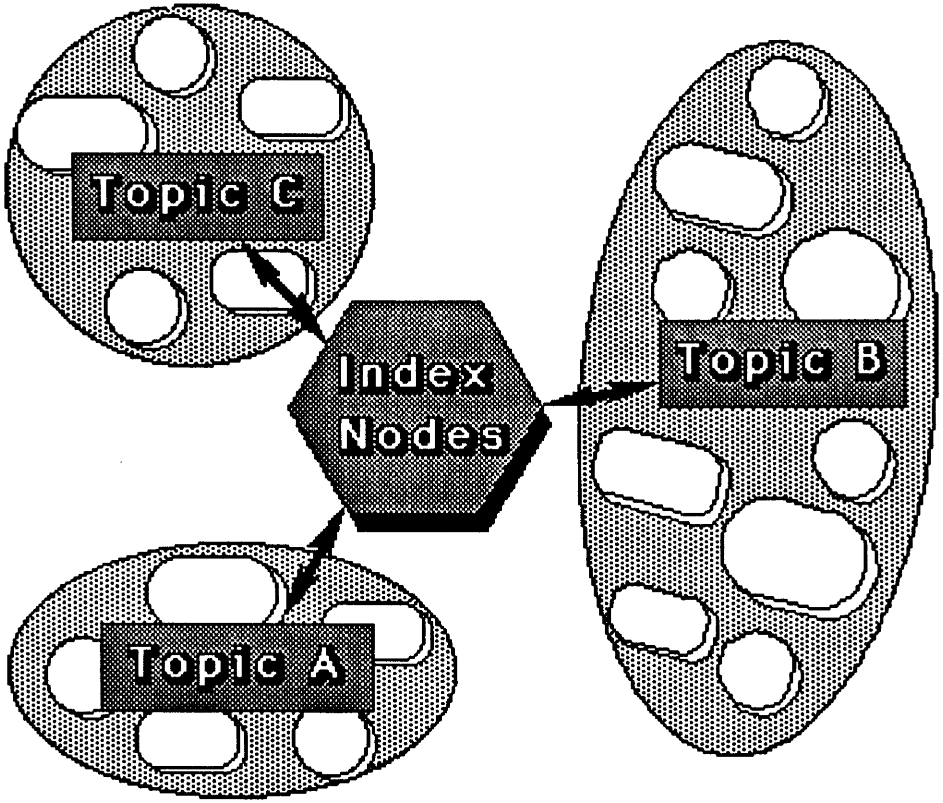


Figure 3.5: Representation of index nodes.

Another type of functional node is the rule node. These nodes list rules and reveal organizational or functional links that satisfy the rules. They represent some kinds of "if-then" relationships and allow a certain level of intelligent manipulation of the building data. A selection, that is to say, an inference process is stored in a rule node. Once we activate it, it will do the intelligent screening work for us. The rule node depends on a high level of object-orientation. We cannot take advantage of the efficiency and the machine's intelligence without full development of the other types of knowledge representation nodes.

III.2: Representing relational data as links of hypermedia

The strength of hypermedia lies in both the nodes, which store segregated and categorized information, and the links, which define the structure and provide the capability of browsing and exploring these nodes. In the architectural CAD systems, hypermedia links have significant meaning in data organization and manipulation. This is partly because building model data are originally more object-oriented and segregated than other data types, such as a text document, which tends to be linearly organized and is difficult to segregate into chunks of data. Conventional design drawings can be thought of as the prototypes of these nodes. But representing relational data as links of hypermedia is a completely new kind of work, which has not been used in the drafting-oriented CAD systems. Another reason for the significance of the links is that there are more relational data in a building model than other types of data. It is impossible to look at a building model as merely an aggregation of building elements. Most elements of building information are contained in the relationships of those elements and their context.

Similar to the object-oriented nodes, ideally, hypermedia links are created to serve a project-specific environment. Although most links may have the same contents in a variety of different projects, such as those organizational links for a specific type of building, each of them is unique in terms of the specific context within which it is created. But, just like a "HyperCard" address book stack, the organizational conventions decide most parts of the link structure. The work left for us is mainly to input each data item to an appropriate node (card), then we can browse it with the built-in links. Building design representation is based on a sophisticated code system. Most links in it are defined by a set of strict patterns. We do not have to create a whole set of links for each project. We can build some of these links into the CAD systems to deal with most of the organizing and manipulating work. Sometimes, in a "HyperCard" address book, we need to endorse a special meaning between two data nodes, for example, that A is the father-in-law of B. A link could be created to represent this relationship. It is impossible to create this kind of link during the stack design process of the address book. In a building model, there are lots of nodes relationships which are similar to this, and they are project-specific. As a result, we need the potential to create links besides using the build-in links of the CAD systems. We will further discuss this problem in terms of its application meanings in Chapter IV.

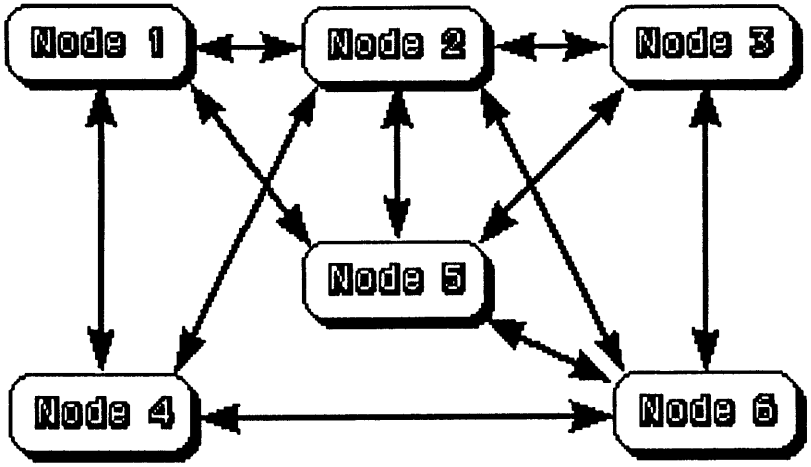


Figure 3.6: Static representation of the relational links.

Flexibility and efficiency are other major concerns in creating hypermedia links. A CAD building model is organized to face the unpredictable tasks and user requirements. This should be reflected in the ways we can not only constantly edit and add links but also in the ways we construct them. In a building model with nodes connected by the static links, a stereotype of relational links is authored throughout the whole model (Figure 3.6). The best part of the structure is that it represents the relationships unambiguously. But it often ends with a rigid and over-complicated link structure. For a building model with thousands of nodes, the number of links we need to represent the building may easily reach an astronomical level. It is practically impossible to create this kind of structure for a building model, and it is also very inefficient for both the system structure and the users. For a building design, the creation of links is an accumulative process. It has to cope with the evolutionary development of design ideas. We can not make the links connect nodes which have not been created, or represent relationships which are not clarified yet.

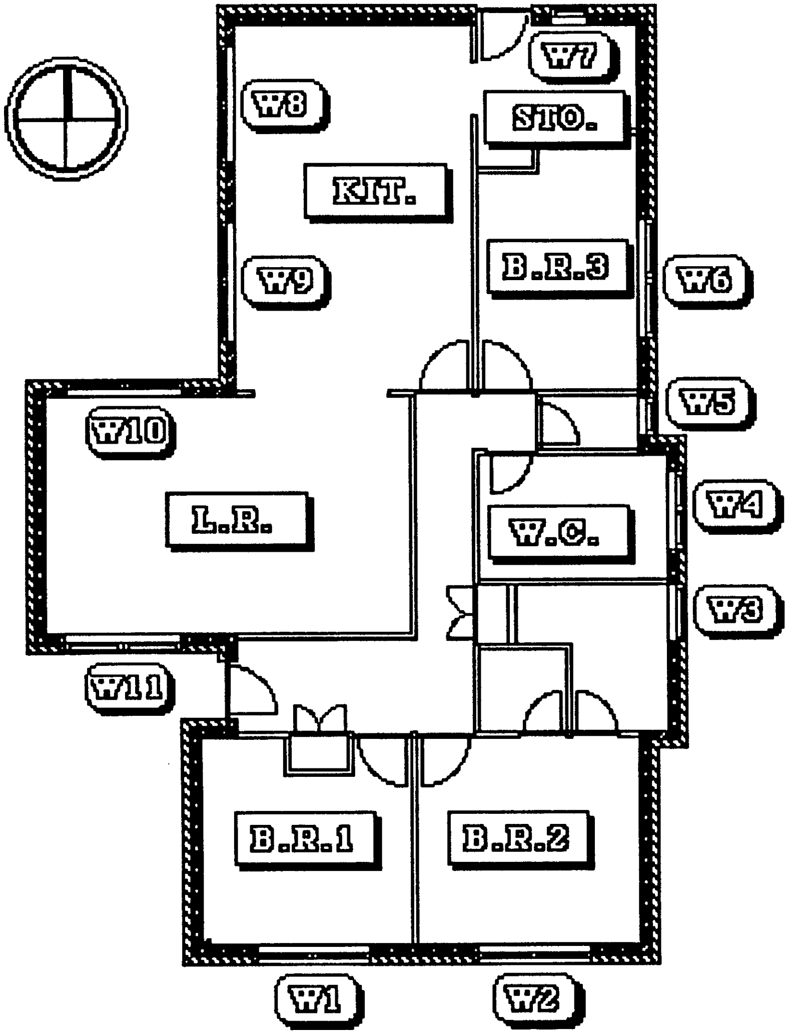
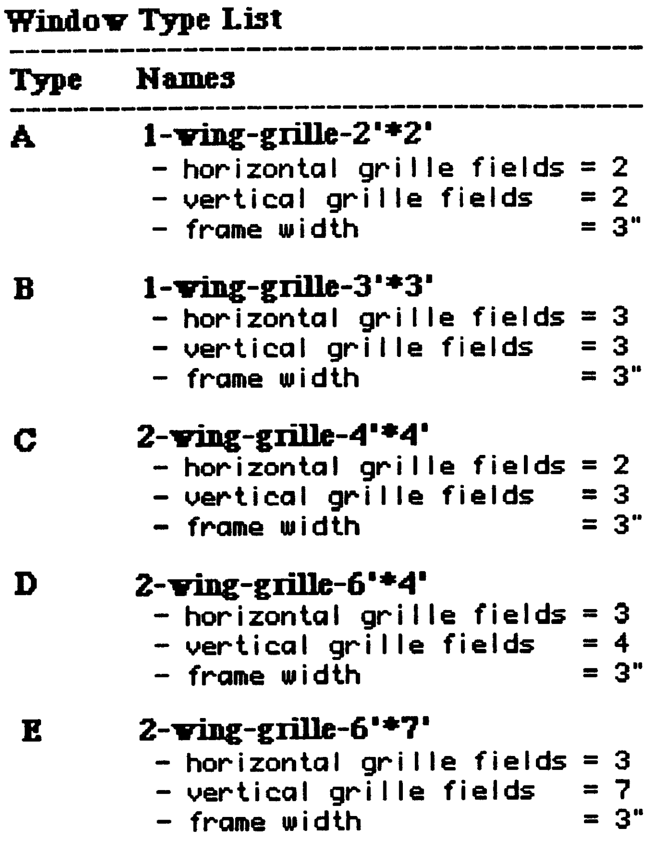


Figure 3.7: Windows in a single family house.

In a hypermedia-based building model, there could be three different types of links to represent the various kinds of relationships which may exist there. Let us use the same examples as shown in figure 3.2 to demonstrate the link types we may use in constructing a hypermedia representational structure. Since the structure to represent each domain of the building is similar and it can be very complicated and tedious to represent the whole building, we will only create the structure which represents the windows.



```
Window Type List
---------------------------------------
Type    Names
---------------------------------------
A       1-wing-grille-2'*2'
         - horizontal grille fields = 2
         - vertical grille fields   = 2
         - frame width              = 3"

B       1-wing-grille-3'*3'
         - horizontal grille fields = 3
         - vertical grille fields   = 3
         - frame width              = 3"

C       2-wing-grille-4'*4'
         - horizontal grille fields = 2
         - vertical grille fields   = 3
         - frame width              = 3"

D       2-wing-grille-6'*4'
         - horizontal grille fields = 3
         - vertical grille fields   = 4
         - frame width              = 3"

E       2-wing-grille-6'*7'
         - horizontal grille fields = 3
         - vertical grille fields   = 7
         - frame width              = 3"
```

Figure 3.8: Window types in the house.

There are altogether eleven pieces representing five types of windows in this house (Figure 3.7, 3.8). If we only choose the window types and their positions in each room of the house to create the links, the relationships of the windows can be represented in this diagram (Figure 3.9).

The first type of link is the organizational link. It represents the ways these windows are assembled in the building. Combined with the representational nodes, it represents the information statement like "there are two Type-A windows used in the building: W5 and W7" or "The position of W7 is the storage room." The first link conveys the message of "Usage (Type-A) = W5 and W7," the second one conveys the message of "Position (W7) = Storage Room." Each of them has a directional meaning. In a sophisticated data representation structure, the information is conveyed in two directions. We can also have "W5 and W7 is Type-A windows," and "a component of the storage room is window W7." In all these cases, the links constitute the organizing

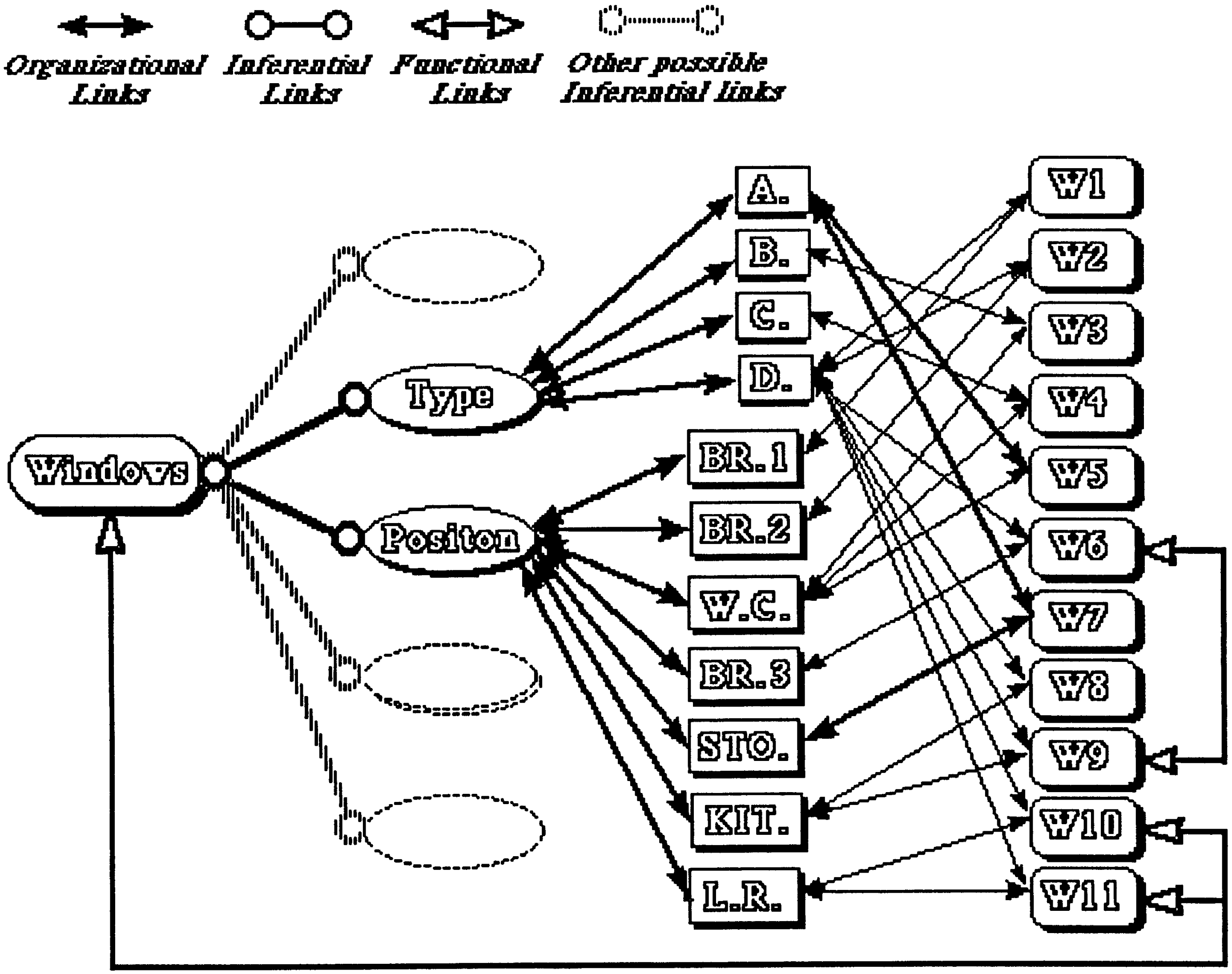


Figure 3.9: Various Links in representing relational data.

structure which connects the nodes representing related data-objects, or different representations of one data-object. In the same way we can have organizational links to represent "story N has room A, B, C, and D," "room A is a conference room on story N," and etc.

It is possible to create most organizational links in an architecture-oriented CAD system, but it is almost impossible to embed the second type of links in a system without a specific project going on. A building model is an organic entity. All data-objects in it are mutually related and influenced. We can only pick up the most direct and common relationships through the organizational links. There are still many overlooked relationships which may be critical to a specific project or user. We may call the links representing these kind of relationships functional links. They are directly related to the operation of a building model, for example, when we think about the windows of this house, we may directly focus on the windows in the living room because the windows in this room add some special scenic or characteristic meaning to this building. We jump over the representations of "Position" and "Living Room" or "Types". Another example is that, for a designer, there may be a need to connect the nodes of W5 and W9, since these two windows are almost aligned on the same line. There are no direct organizational relationships between them; for them both, the other is only "another data-object in the house."

The third type of link is the inferential link. In representing the windows of the house, we may have numerous ways of organizing them. An inferential link between the windows and one organizational approach can create a conditional connection, which possesses intelligent meaning for the users of the building model. For example, the inferential link between "Windows" and "Type" may contain something like: "if we are talking about the window schedule..." and the link between "Windows" and "Position" contains "if we are considering the design of..." (Figure 3.9). This can be further defined by another layer of inferential links (Figure 3.10). If we combine these links, we can derive a message from the building model like "if we are considering the positions of windows in regard to the HVAC equipment of the house, we had better go to the windows in the kitchen, the W.C. and the storage room."

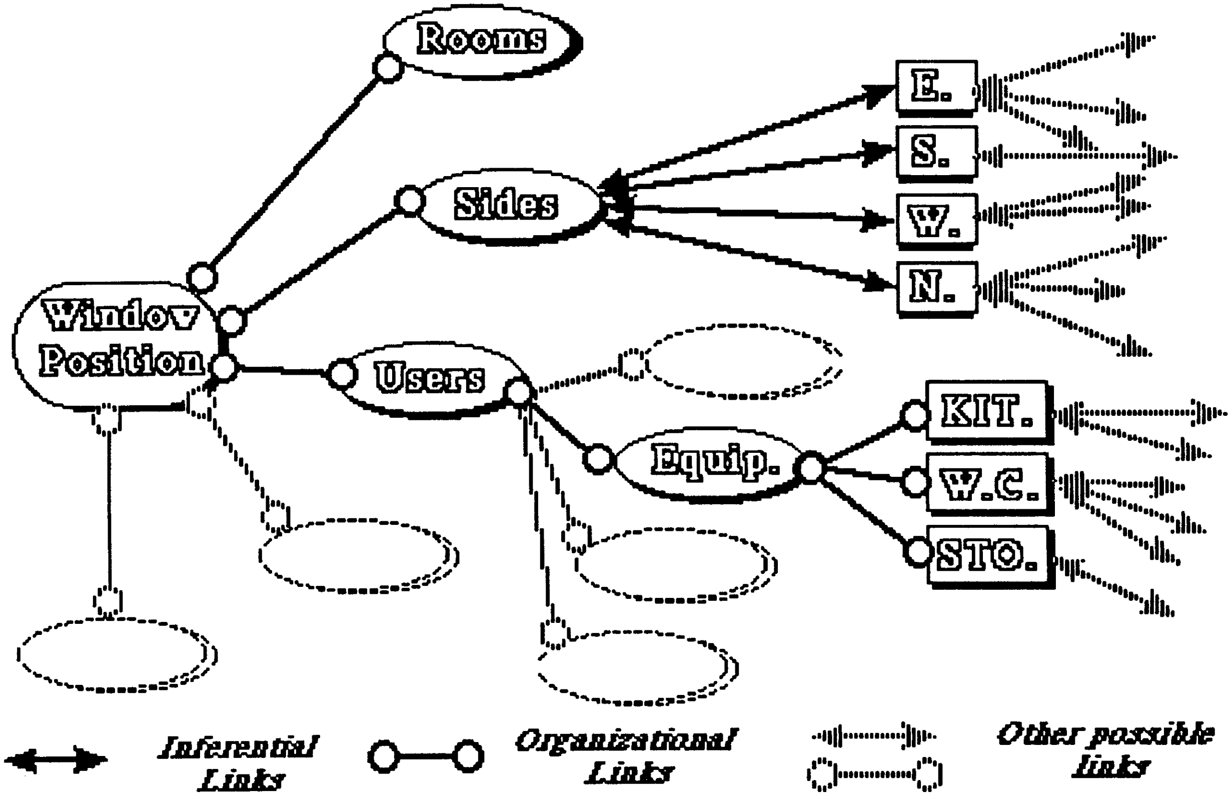
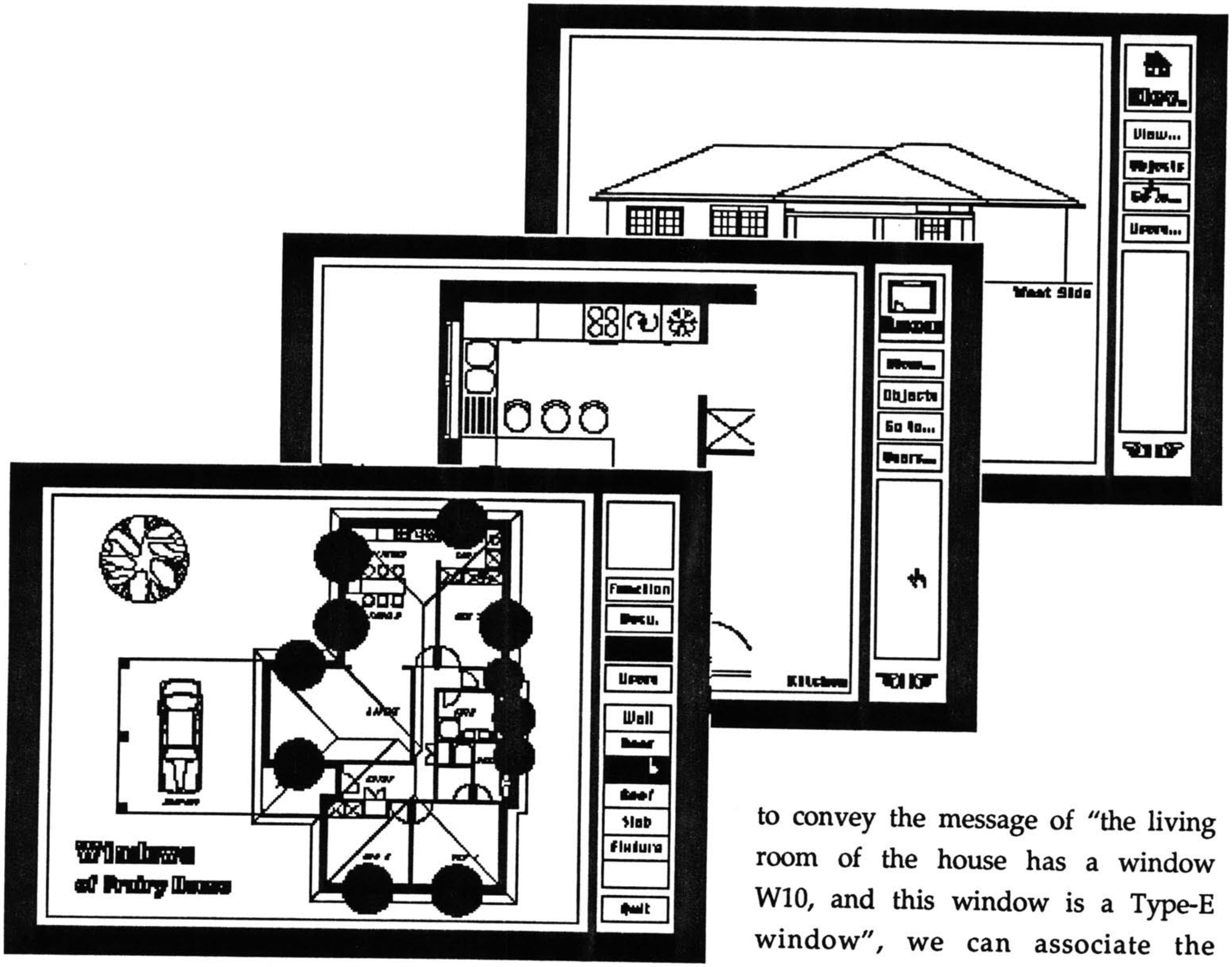


Figure 3.10: Inferential links in representing intelligent information.

1. Windows are represented as objects of the house. There are numerous ways to do this. For demonstration purposes, we may have the window represented only in the scope of the house, the sides of the house, and the rooms of the house (Figure 3.11).
2. Each window of the house is represented as an object by itself. A window object is unique in terms of its differences from other building objects, including other windows, such as windows "W1" and "W11" These are identical window types (Figure 3.2), but they can be represented as different windows with different positions, functions and relationships with other objects of the house. So each window can be represented as a representational node (Figure 3.12). We choose the plan and elevation in order to explain how the window is related to the wall of the building and we choose the perspectives to display the appearance of the window in this building.
3. The window types are categorized by their physical and visual characteristics. We may represent a window only by its own character, disregarding its usage in the building (Figure 3.13). These representation nodes are the same as items in a window library. They can be totally separated from a particular building and exist as window object by itself. The graphic and text data describe its physical characteristics and some related information. We may further include some detailed drawings to display the construction details of this type of window.

In order to represent the window domain, we need to create further object nodes through which we can access the organizational links among these different representations. We can use buttons or graphics of "SuperCard" to imitate these nodes and embed the links into the button or graphic scripts. Then we can access different representations through organizational links by choosing these buttons or graphic. For example, if we want



to convey the message of "the living room of the house has a window W10, and this window is a Type-E window", we can associate the graphic "Living Room" on the house

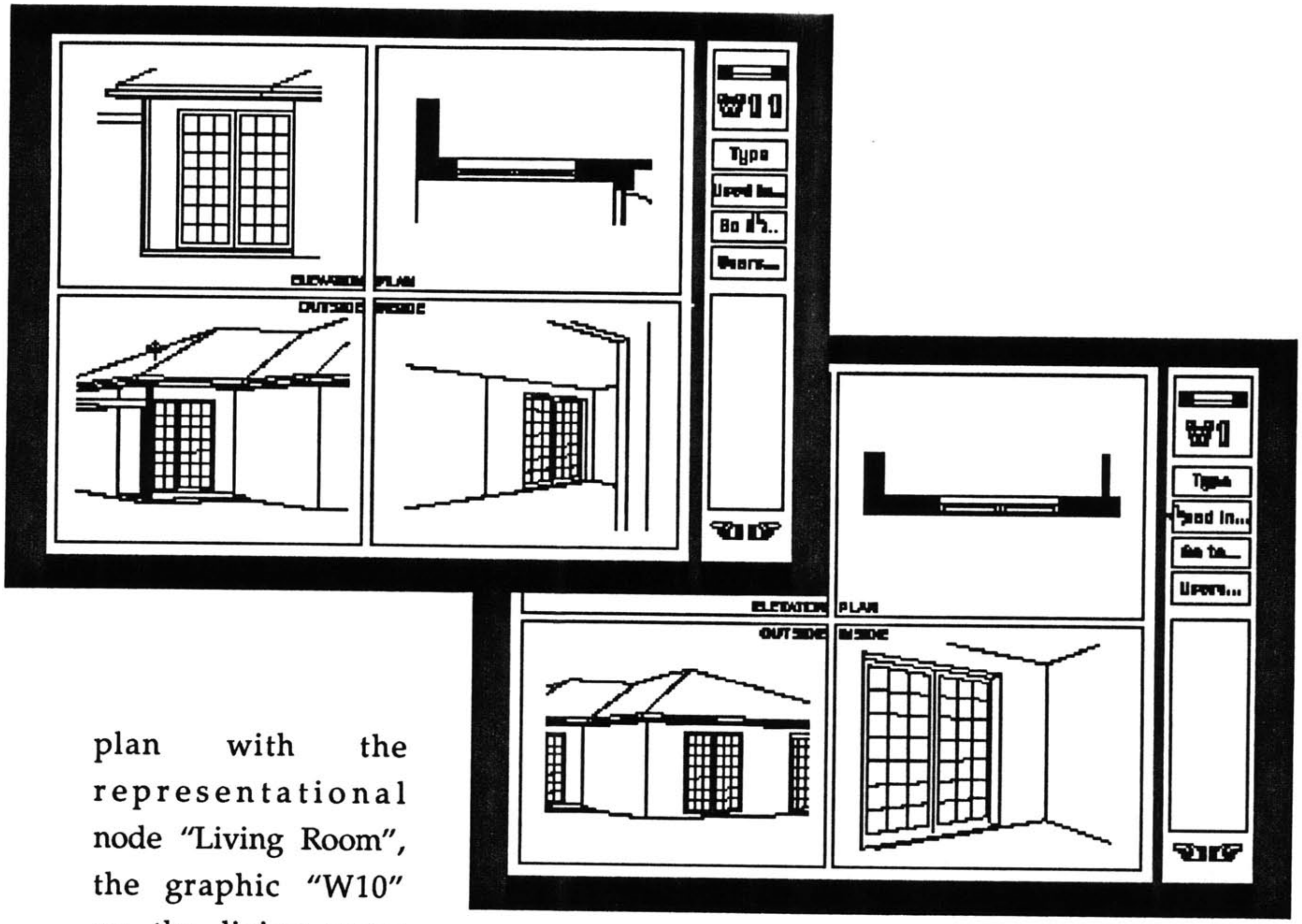
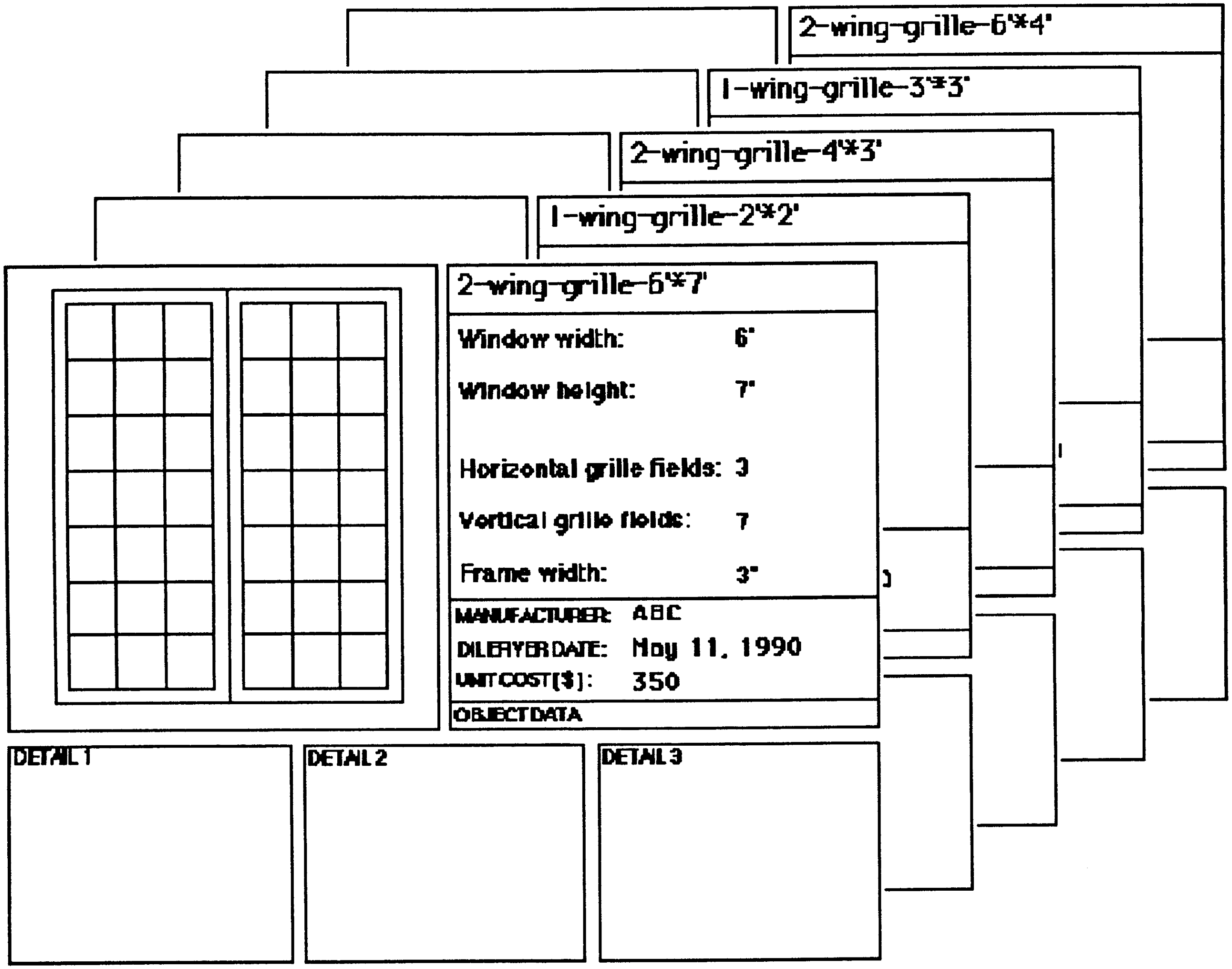


Figure 3.12: Two identical windows are represented as two different data-bojects.

plan with the representational node "Living Room", the graphic "W10" on the living room node with the representational node of "W10", and use button "Type" to connect the representational node "W10" with the representational node "Type-E". This can also be done in



Window width: 6'

Window height: 7'

Horizontal grille fields: 3

Vertical grille fields: 7

Frame width: 3"

MANUFACTURER: ABC

DILERYER DATE: May 11, 1990

UNIT COST($): 350

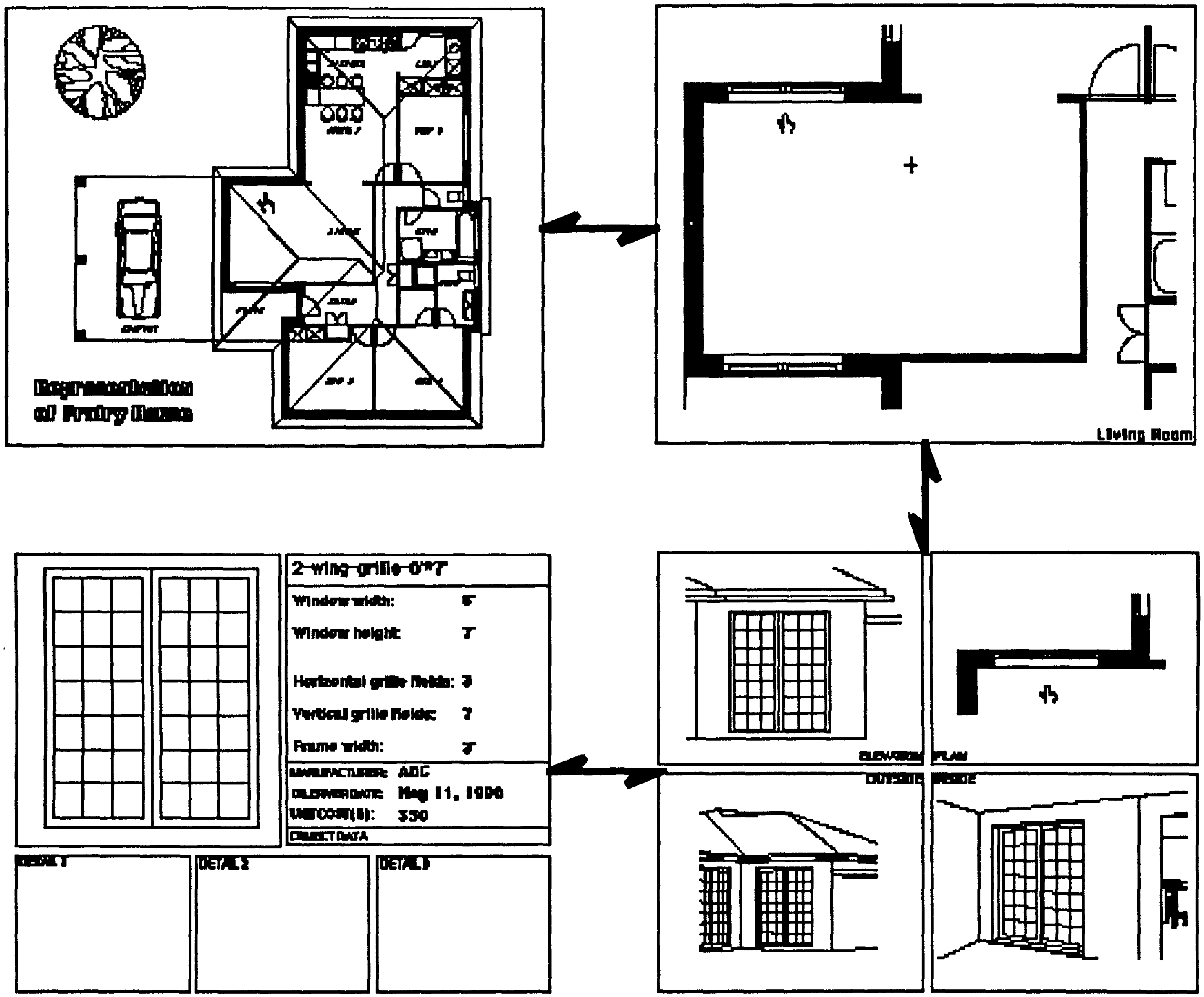
OBJECT DATA

DETAIL 1

DETAIL 2

DETAIL 3

Figure 3.13: Window types are represented in different representational nodes.



2-wing-grille-6*7'

Window width: 6'

Window height: 7'

Horizontal grille fields: 3

Vertical grille fields: 7

Frame width: 3"

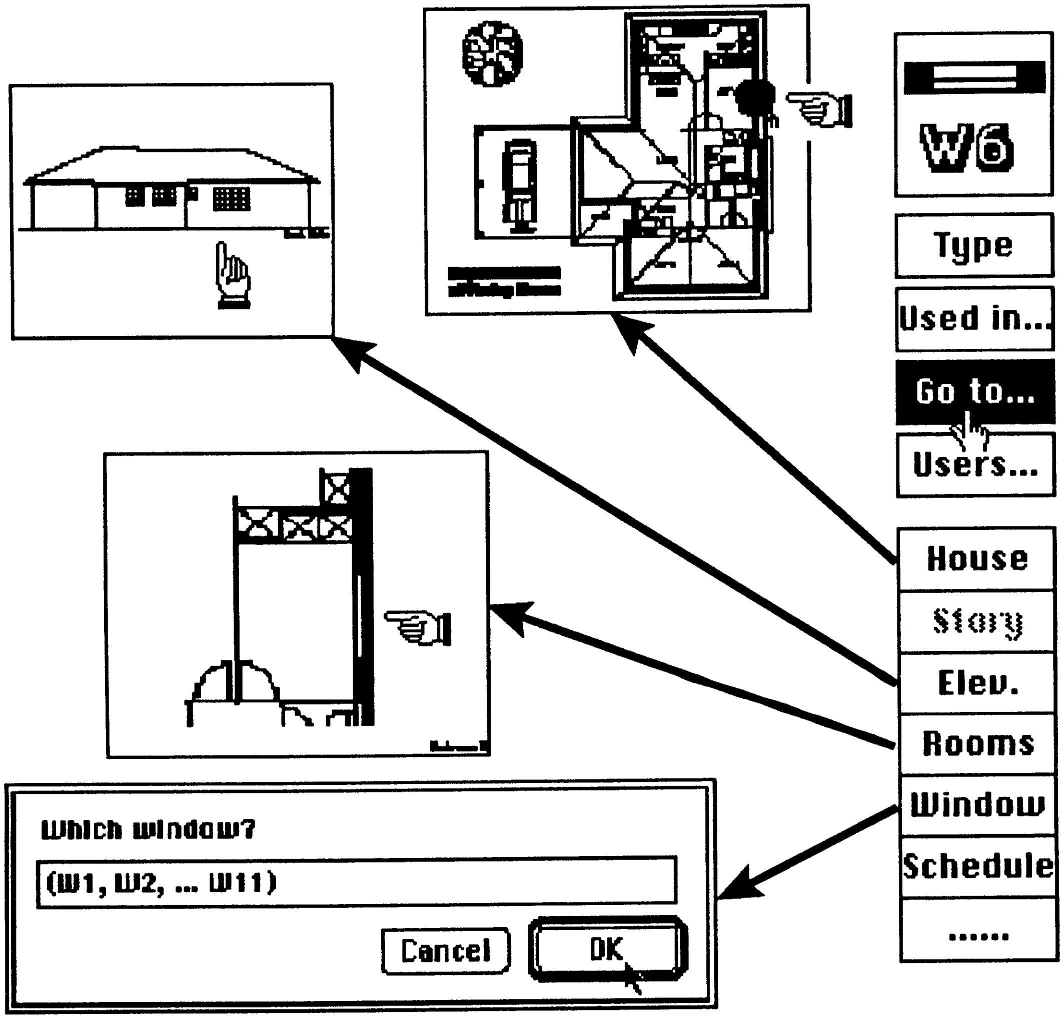
MANUFACTURER: ABC

DELIVERY DATE: May 11, 1990

UNIT COST($): 350

OBJECT DATA

Figure 3.14: Using organizational links to connect different nodes.



Which window?

(W1, W2, ... W11)

Figure 3.15: Following the functional links to browse through different representational nodes.

reverse: if the message we want to convey is "'Type-E' window 'W10' is part of living room of the house", we can use buttons to connect node "Type-E" with node "W10", node "W10" with node "Living Room" and node "Living Room" with the house plan node (Figure 3.14). These buttons are the object nodes in terms of their ability to reveal the organizational relationships of these data-objects. In this way, we can construct various organizational links among these different representations.

Functional and inferential links can also be represented in button nodes of "SuperCard." We may need several ways to navigate through these representations. Some of these operations only check for different representations of a data-object, but some of them may move from one data-object to another or one topic of data-objects to another; we may want to see "W9" after reviewing "W5" or after examining the washing closets of the house, we may want to check for locations of all windows. To do this, we can create some navigational buttons with script like "go to..." (Figure 3. 15). These button nodes can be organized and controlled by other buttons; in studying the house, we may have several approaches to access the building model. We may need some index nodes to further define the access area. The inferential abilities of the links are decided by the sophistication of the script. For each window representation node, if we tell the computer what kinds of problems the user is considering about this window, it can automatically show the nodes containing information directly concerned with the specific problems.

In this part, we have created a simple representational model, which incorporates the concepts of nodes and links to represent the windows of a single family house. In the next chapter, we will use this model to examine the hypermedia abilities from the view-point of data inquiry and access of building design.

Chapter IV: Application and Problems

In Chapter III, we explored the ways of incorporating the concepts of hypermedia with the data organizational structure of a CAD building model. We embedded the representational structure of building data into the network of hypermedia nodes and links. Based on this idea, we created a simulated representational model for the windows of a single family house. In the first part of this chapter, we will use this model to demonstrate the abilities of a hypermedia-based building model in providing its users with various data access possibilities. Then, after a brief review of the ideas discussed in last two chapters, we will analyze the problems which may occur in a real design environment with a hypermedia-based data structure. This may become the point of departure for further exploration and discussion of this problem.

IV.1: Browsing through a hypermedia-based representational model

The process of accessing information in a hypermedia-based building model can be defined as two related sub-processes. First, we reach the representational node which contains the information we are interested in by browsing through various links of the model. This can be a navigating process starting from any one representational node and ending with another in the model. Second, we ask for information about the data-object represented in the node. This browsing process can lead us to any representational nodes which convey relevant messages about the data-object. The first sub-process may define the manipulation of the second one, and from any point in the second sub-process, we may start the first one searching for a new data-object.

From the perspective of data representational structure, we can retrieve information in two reversive directions. After defining the attributes of a data-object, we can ask for information about the value of one particular attribute of the data-object, and we can also acquire the data-object information from its value. If we explain the processes using the formula " A (O) = V " (Page 28), the first one can be described as " A (O) = ? "; the second one is " A (?) = V ." Among them, "A", the attribute, is defined by the users in solving a particular problem. These two directional retrieval operations are intertwined with the two sub-processes of accessing information in a hypermedia-based building model.

IV.1.1: Searching for a required data-object

Since the model we built in the last chapter is mainly about the windows of the house, let us choose one window data-object as the one we are interested in. In fact, it can be any data-object in a building model—a door, a room, a story or a part of the building. But the idea is the same: from the standpoint of the data-object, we can come from everywhere and go everywhere in the model. For a simple data-object like a window, this kind of accessing process may look trivial and unnecessary, but it is not difficult to understand its significant meaning in accessing a complicated data-object like the story of a multi-story building.

Suppose the window we are interested in is window "W8." Depending on how we describe it, there are several different ways to approach it:

1. "That is a window in the house." We may start with a representation of the house (Plan), make a statement that we are interested in windows, (choosing button "Window" under menu "Element"), and then, by clicking on window graphic "W8", we reach the representational node of "W8" (Figure 4.1).

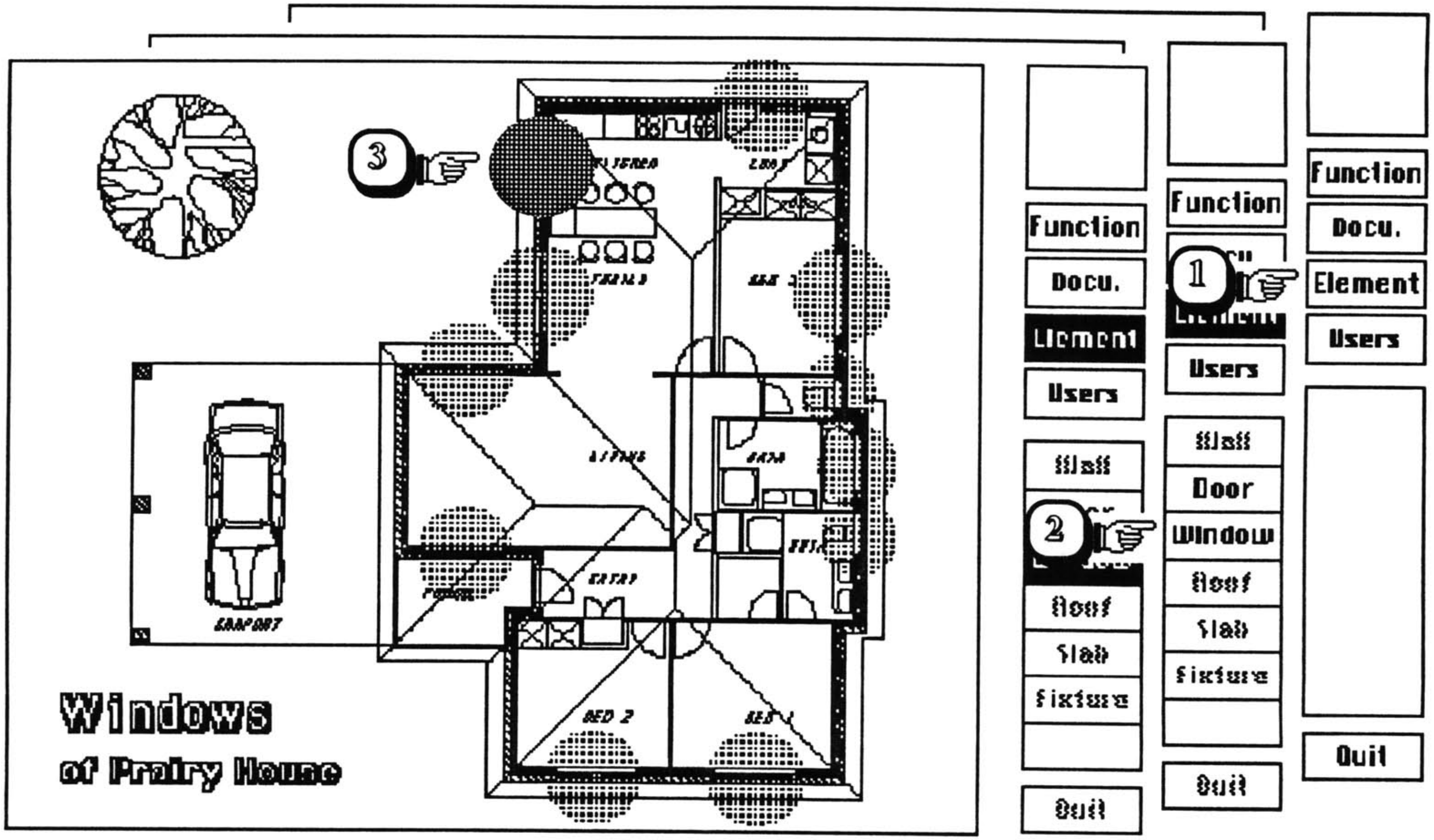


Figure 4.1: "That is a window in the house": after pushing these buttons sequentially, we can reach the representational node of the window we are interested in.

2. "That is a window in the west side of the house" or "that is a window in the kitchen." We may start with a representation of the west side of the house (West Elevation) or a representation of the kitchen (Plan). After we recognize the window we can simply click

on it, or choose the "Window" button under menu "Go to...," this will lead us to the node of "W8" (Figure 4.2, 4.3).

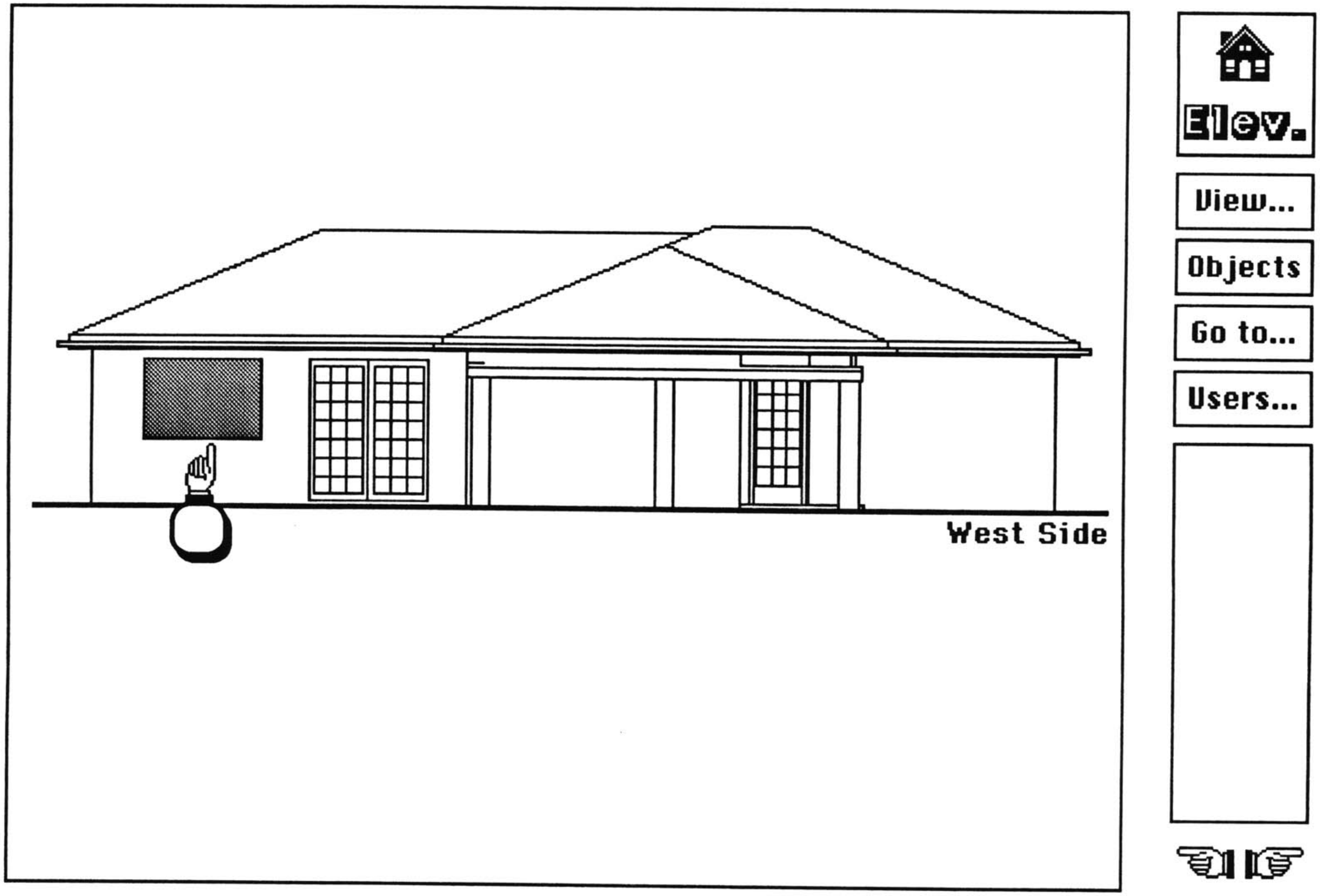


Figure 4.2: "That is a window in the west side of the building": we can reach the window representational node by clicking the window in the west elevation.

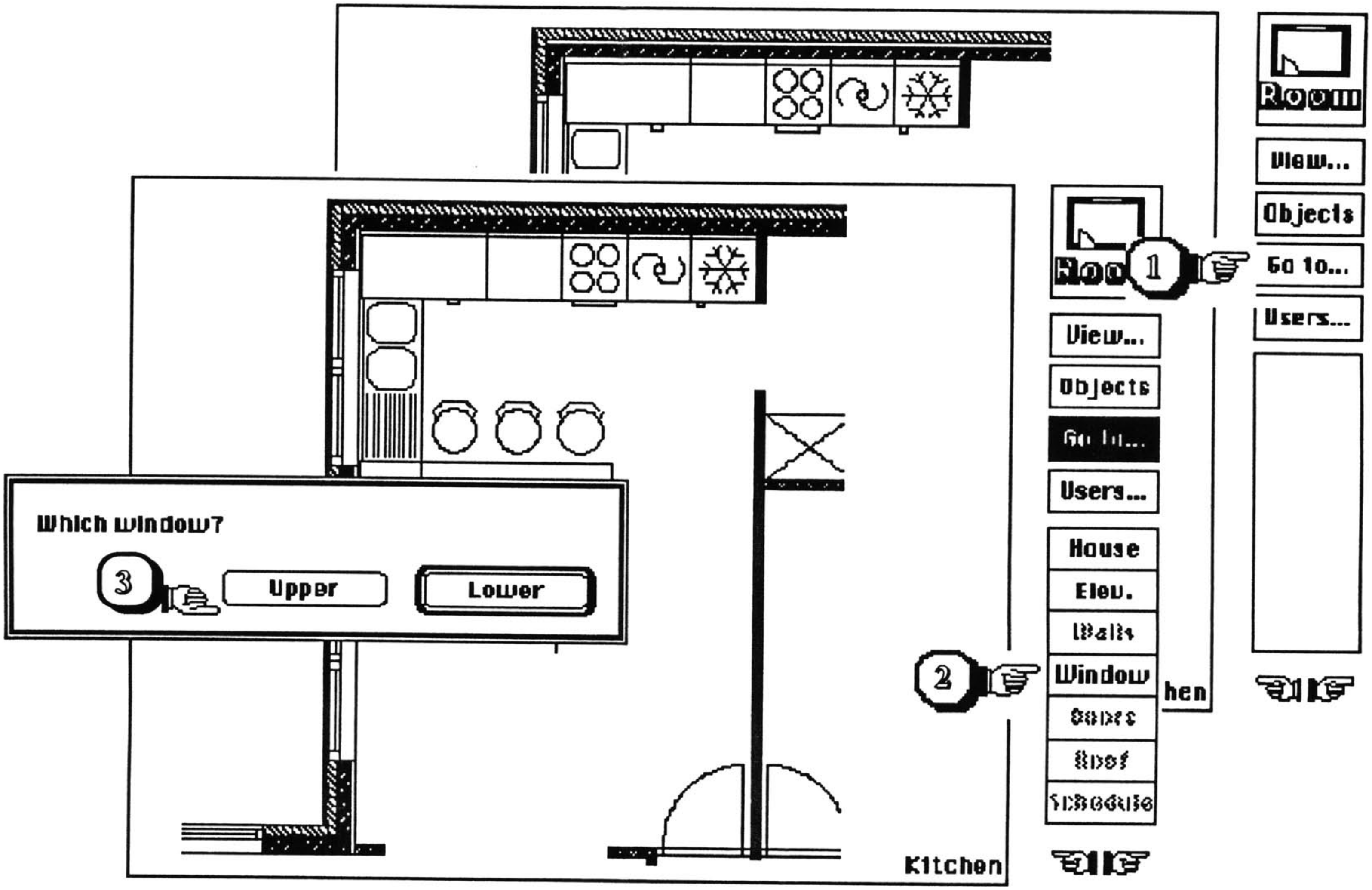
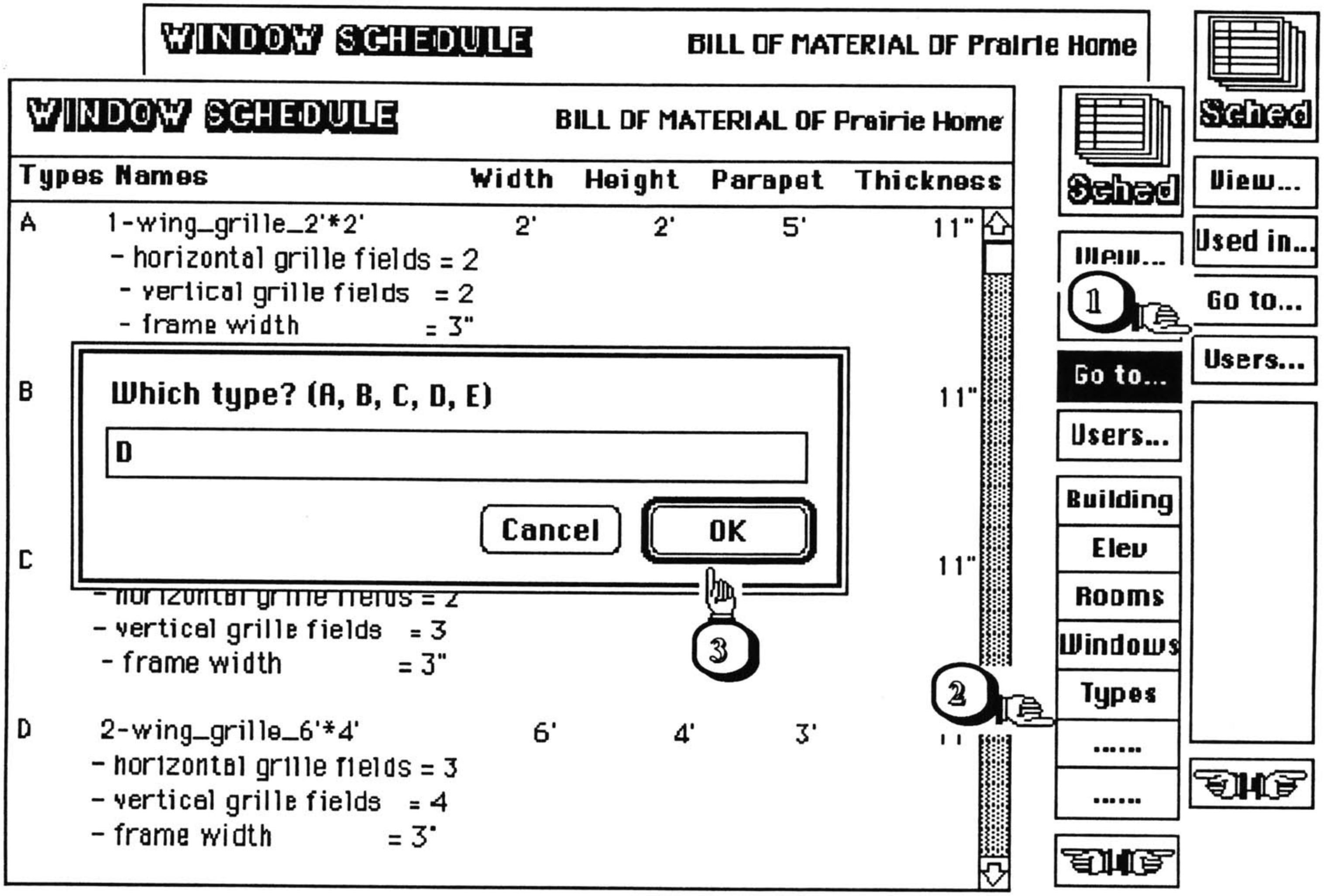


Figure 4.3: "That is a window in the kitchen": we can also reach the window item by pushing the buttons in the control menu.

3. "That is a type of window in the window schedule." Every record in the schedule can be linked to a representation node of window type (Figure 4.4), and this can further lead us to the windows which belong to the type (Figure 4.5).



WINDOW SCHEDULE — BILL OF MATERIAL OF Prairie Home

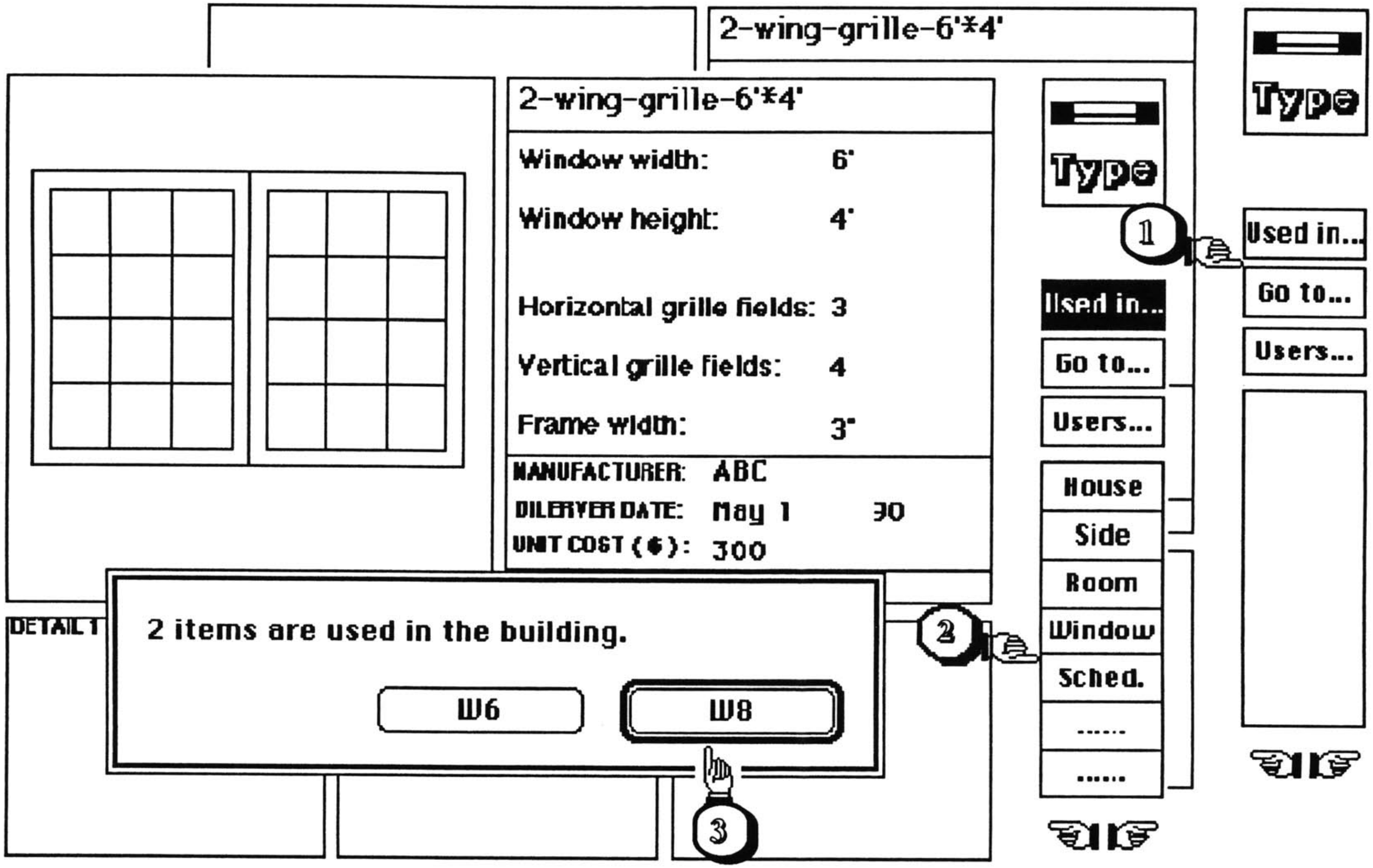
WINDOW SCHEDULE — BILL OF MATERIAL OF Prairie Home

	Types Names	Width	Height	Parapet	Thickness
A	1-wing_grille_2'*2' - horizontal grille fields = 2 - vertical grille fields = 2 - frame width = 3"	2'	2'	5'	11"
B					11"
C	- horizontal grille fields = 2 - vertical grille fields = 3 - frame width = 3"				11"
D	2-wing_grille_6'*4' - horizontal grille fields = 3 - vertical grille fields = 4 - frame width = 3"	6'	4'	3'	1 1

Which type? (A, B, C, D, E)

D

Figure 4.4: "That is a type of window in the window schedule": pushing these buttons sequentially, we can reach the representational node of the type of window.



2-wing-grille-6'*4'

2-wing-grille-6'*4'

Window width: 6'

Window height: 4'

Horizontal grille fields: 3

Vertical grille fields: 4

Frame width: 3"

MANUFACTURER: ABC

DILERYER DATE: May 1 90

UNIT COST ($): 300

2 items are used in the building.

Figure 4.5: By further pushing these buttons sequentially, we can reach the window item.

4. "That is another window in this elevation" or "that is a window in the same room." Supposedly we are in one window representational node and want to go another. We click

on the button "Elev." or button "Room" under the menu "Go to...," and then choose the window we want to see (Figure 4.6, 4.7). And we can also click on button "Window" to go to any window items we are interested in (Figure 4.8).

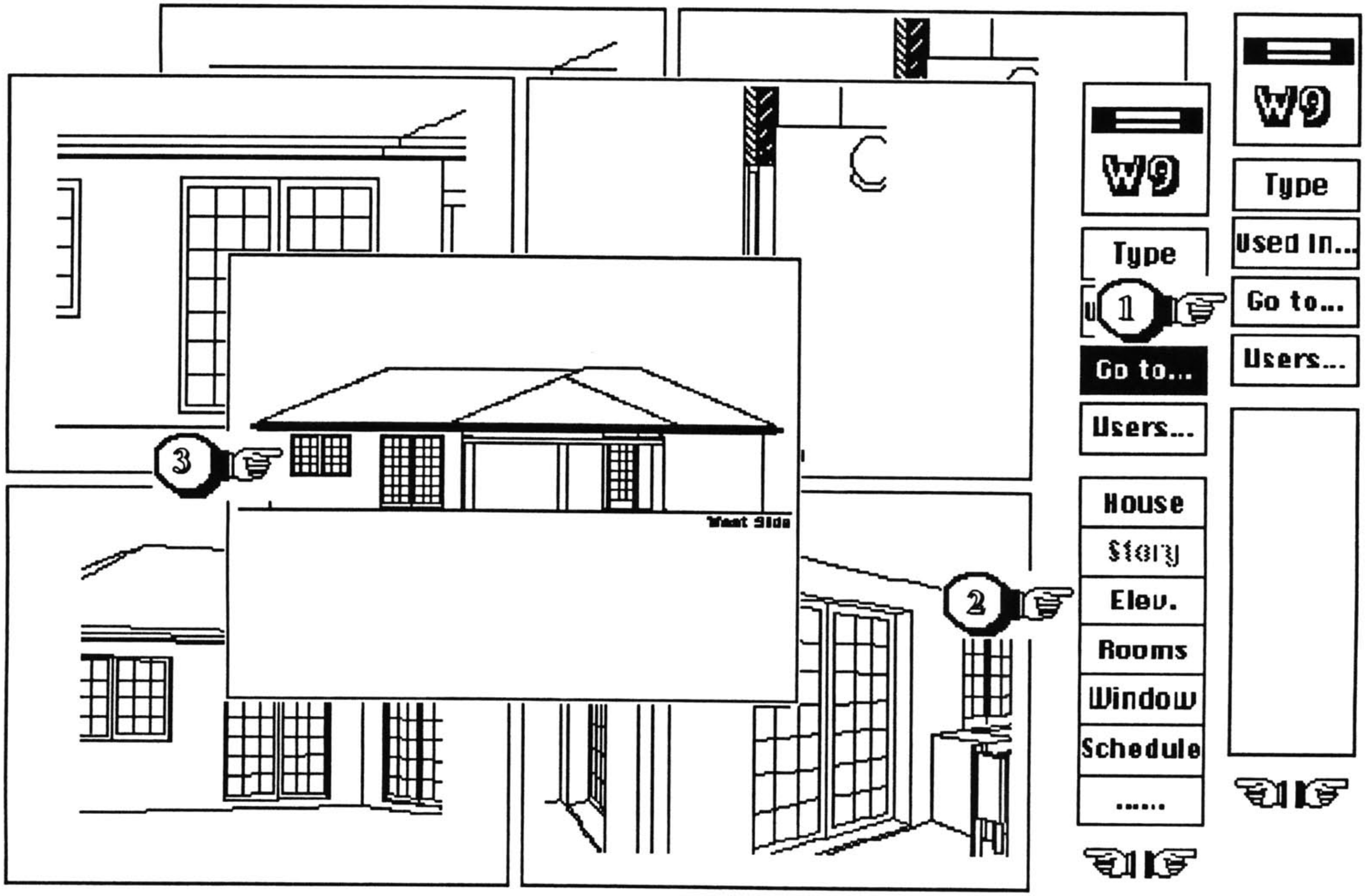


Figure 4.6: "That is another window in this elevation."

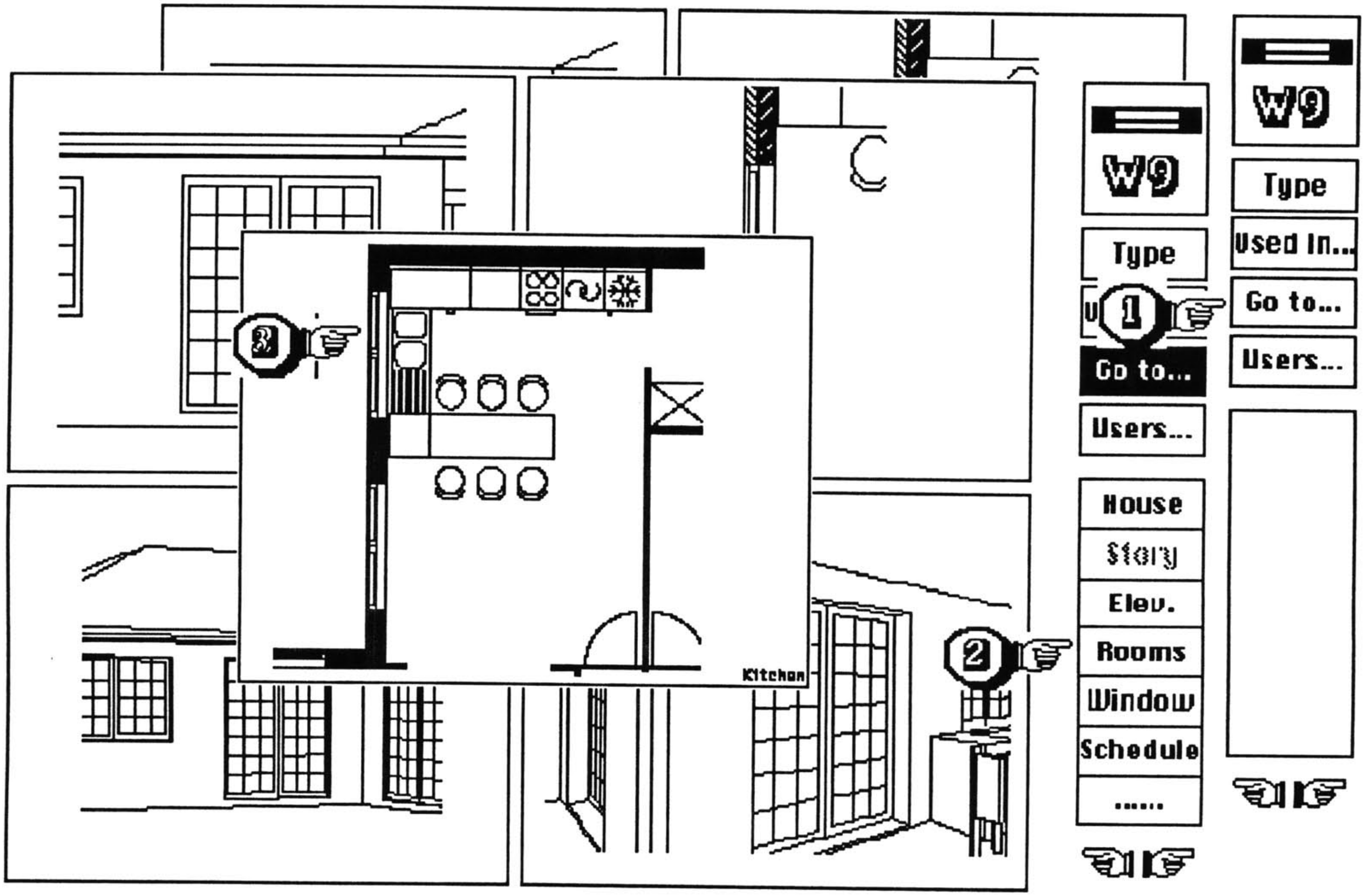
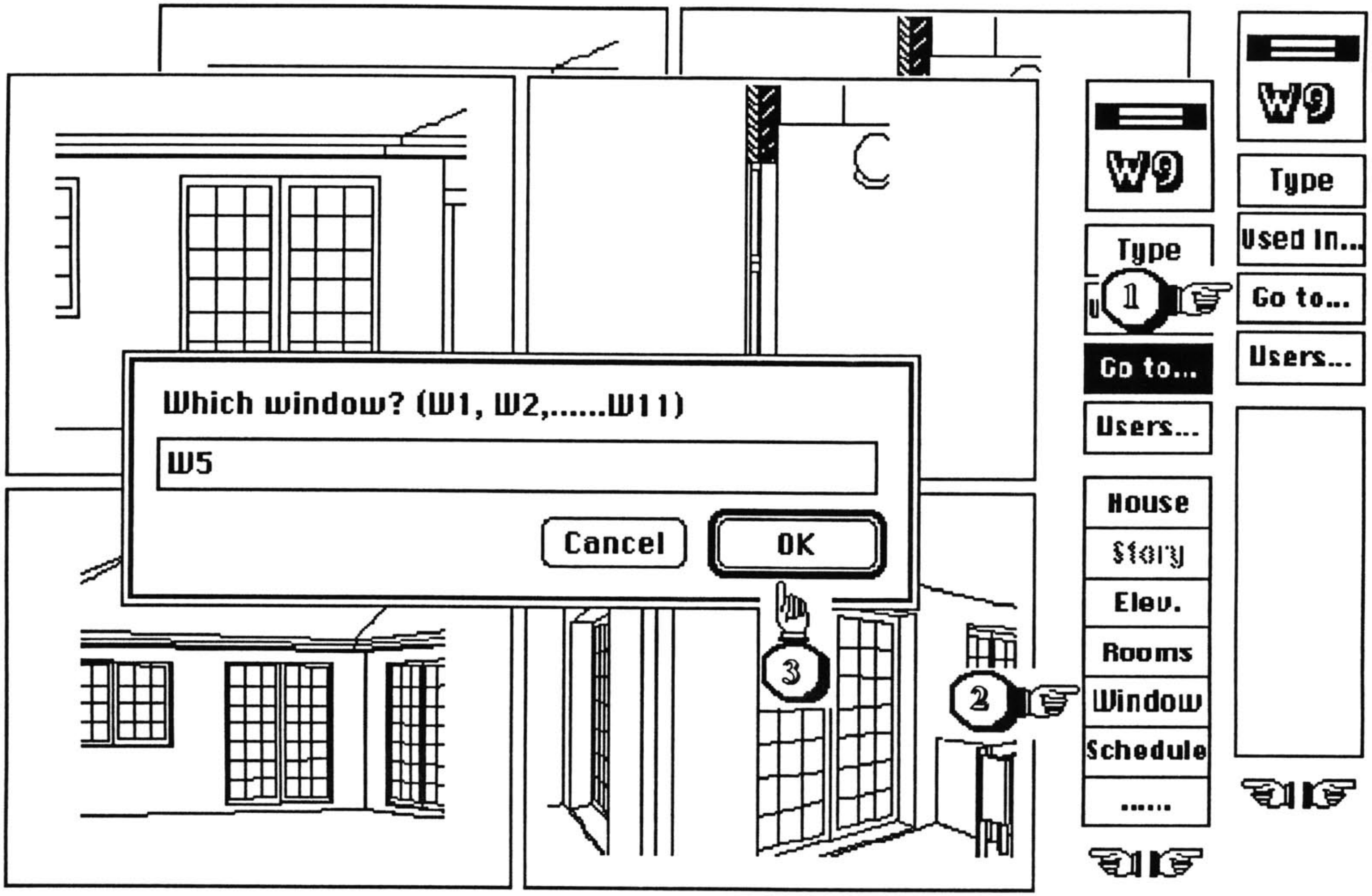


Figure 4.7: "That is a window in the same room."



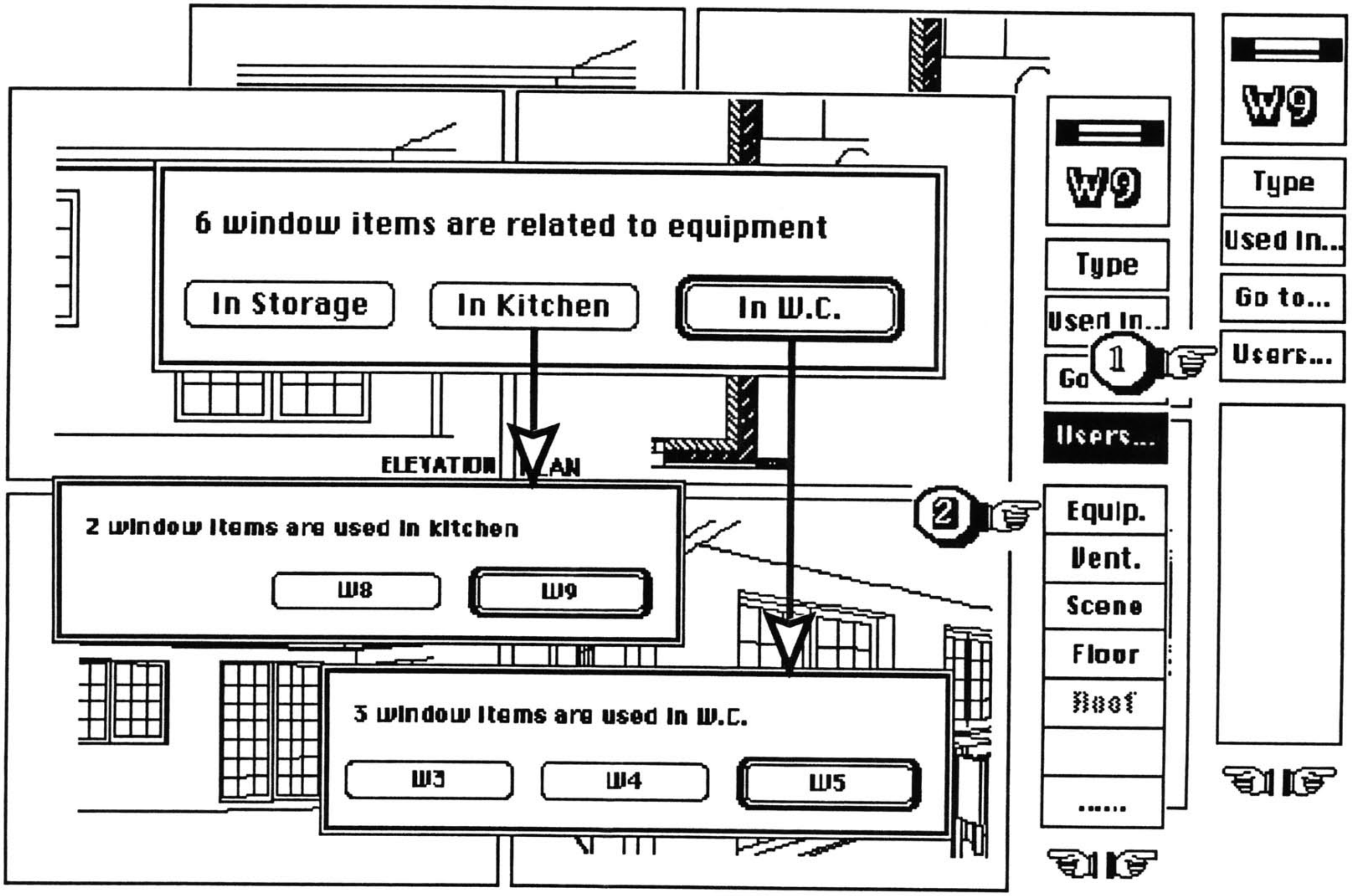
Which window? (W1, W2,......W11)

W5

Figure 4.8: Pushing these buttons sequentially will lead us to any other window items.

5. "That is the windows which may have some problems with the equipment design." In any window representation node, we can choose the "Equip." button under menu

"Users...". This will lead us to the window representation which may relate to the equipment design (Figure 4.9).



6 window items are related to equipment

2 window items are used in kitchen

3 window items are used in W.C.

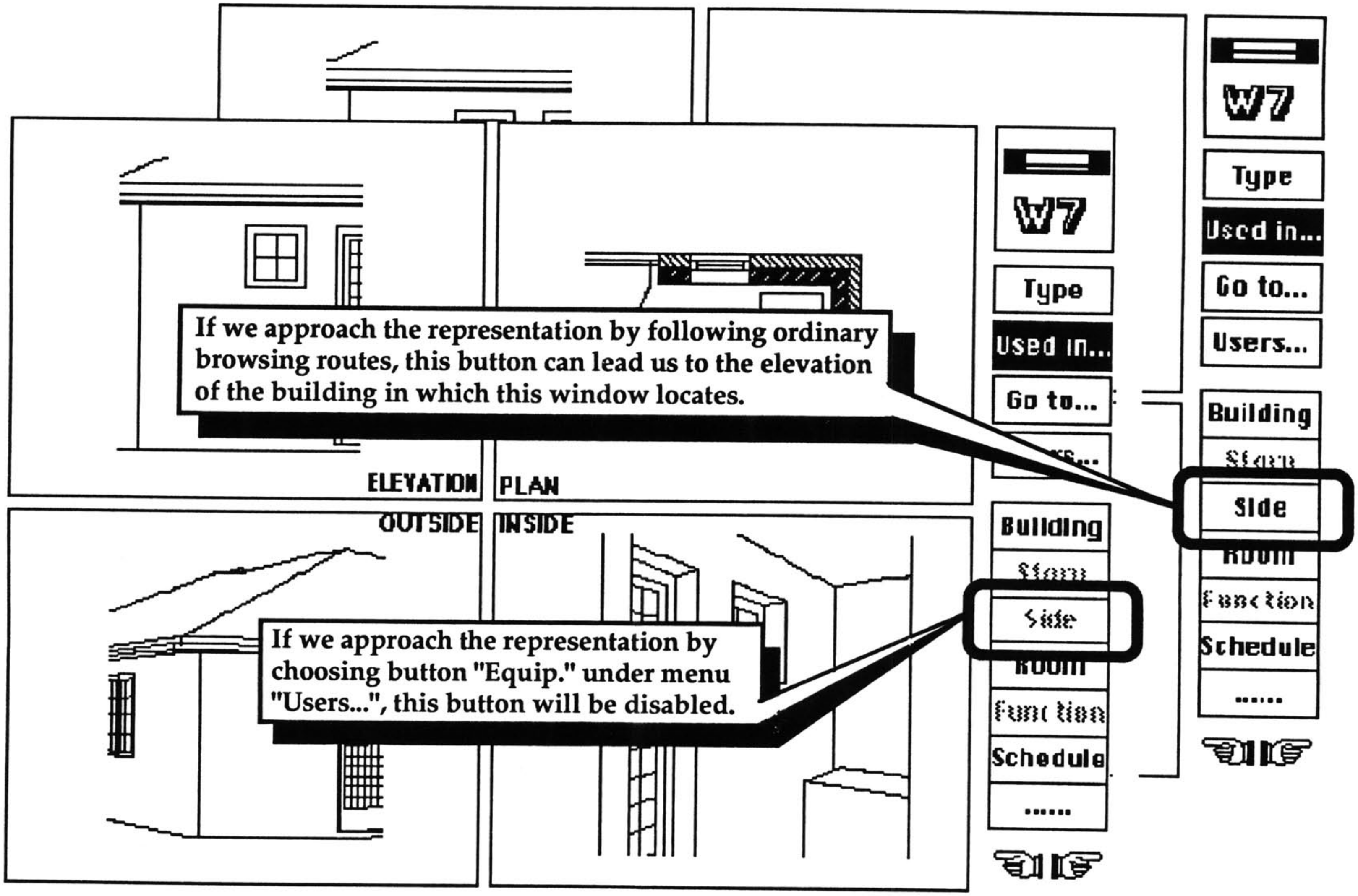
Figure 4.9: "That is the windows which may have some problems with the equipment design."

The way we access the data-object may also have a control effect on the ways we can ask for information about the data-object. This is programmed in the inferential links of the model. For example, if we approach the window representation node from the viewpoint of equipment design, the inquiry path to the building elevation may be unnecessary. So if we choose to check the equipment, the button node representing the links to the elevation representations will be disabled (Figure 4.10). But if we want to edit the window, this connection can be resumed. In this way we can adjust organizational relationships to the requirements of design operation. It gives the system some intelligent ability in coping with design and avoids making the building model overly complex and causing confusion to the users.

From the representational node of "W8", it is also possible to go to any other representational node in the model. Under the menu "Go to...", we have several choices of places we may need to go during the browsing process (Figure 4.6, 4.7, 4.8).

IV. 1. 2: Asking for information about a data-object

After we access the data-object, we can perform various inquiry operations to obtain the information we need in solving a problem concerned with this particular data-object. Since it is impossible to include the values of all the attributes of this data-object in one representational node, we may need several different nodes to completely describe every aspect of the data-object. So the inquiring process becomes a browsing process pursuing the attribute values of the data-object. At the time we define the problem, we define the attributes information of the data-object we should know in order to solve the problem, then by choosing the object node which store the link between one particular attribute and its value, we can reach the node containing the value information.



If we approach the representation by following ordinary browsing routes, this button can lead us to the elevation of the building in which this window locates.

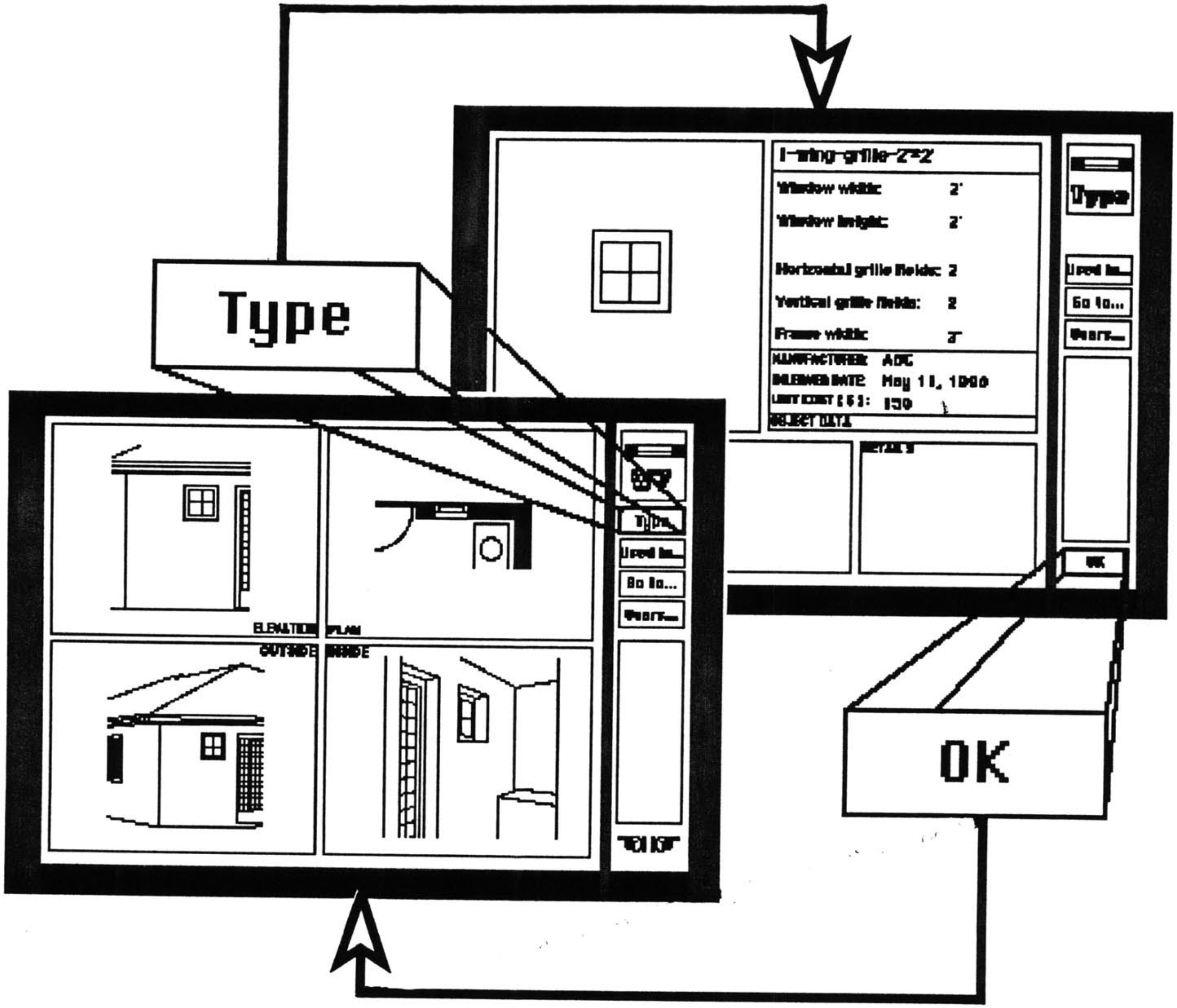
If we approach the representation by choosing button "Equip." under menu "Users...", this button will be disabled.

Figure 4.10: If we choose to check the equipment, the button node representing the links to the elevation represetations will be disabled.

Unlike browsing for a data-object of a building model, asking for attribute values is round-trip. We start from a node representing the data-object, go to various representational nodes containing information about its values and end with the same node we started out with. This is reflected in the demonstration model. If we are browsing for the related representations of one data-object, there is always a path available to us to go back to the original representational node.

This time if we choose window "W7," after reaching the representational node of this particular window, we are provided with a set of buttons which contain the organizational relationships of the window and its various attribute values; if we click on button "Type," we can see a representational node display the information about this particular type of window (Figure 4.11). Under the menu "Used in...", we have several buttons which can lead us to representations showing the position of this window in the house, the side of the house and the room it locates. So we can derive the statement of "the position of W7 is in the storage room" (Page 39). In this case, we think of "position" as an attribute of data-object "W7" and "Storage Room" is the value of the attribute (Figure 4. 12).

But if we take a view from the domain of rooms in this house, "W7" can be regarded as a value of the "Storage Room" under the attribute of "component". So the same browsing process can be regarded as "after defining the attribute "component" and the value "W7", we ask for the information about data-object "Storage Room"". Both browsing directions are supported by this simple model. In the representational node of "Storage Room", if we choose button "Windows" under menu "Objects", the node "W7" will be displayed (Figure 4.13). So we can derive the statement "a component of the storage room is window W7."



I-wing-grille-2*2'

Window width: 2'

Window height: 2'

Horizontal grille fields: 2

Vertical grille fields: 2

Frame width: 2"

MANUFACTURER: ADC

DELIVERED DATE: May 11, 1990

UNIT COST [$]: 150

OBJECT DATA

Figure 4.11: The browsing loop in inquiring information about window type.

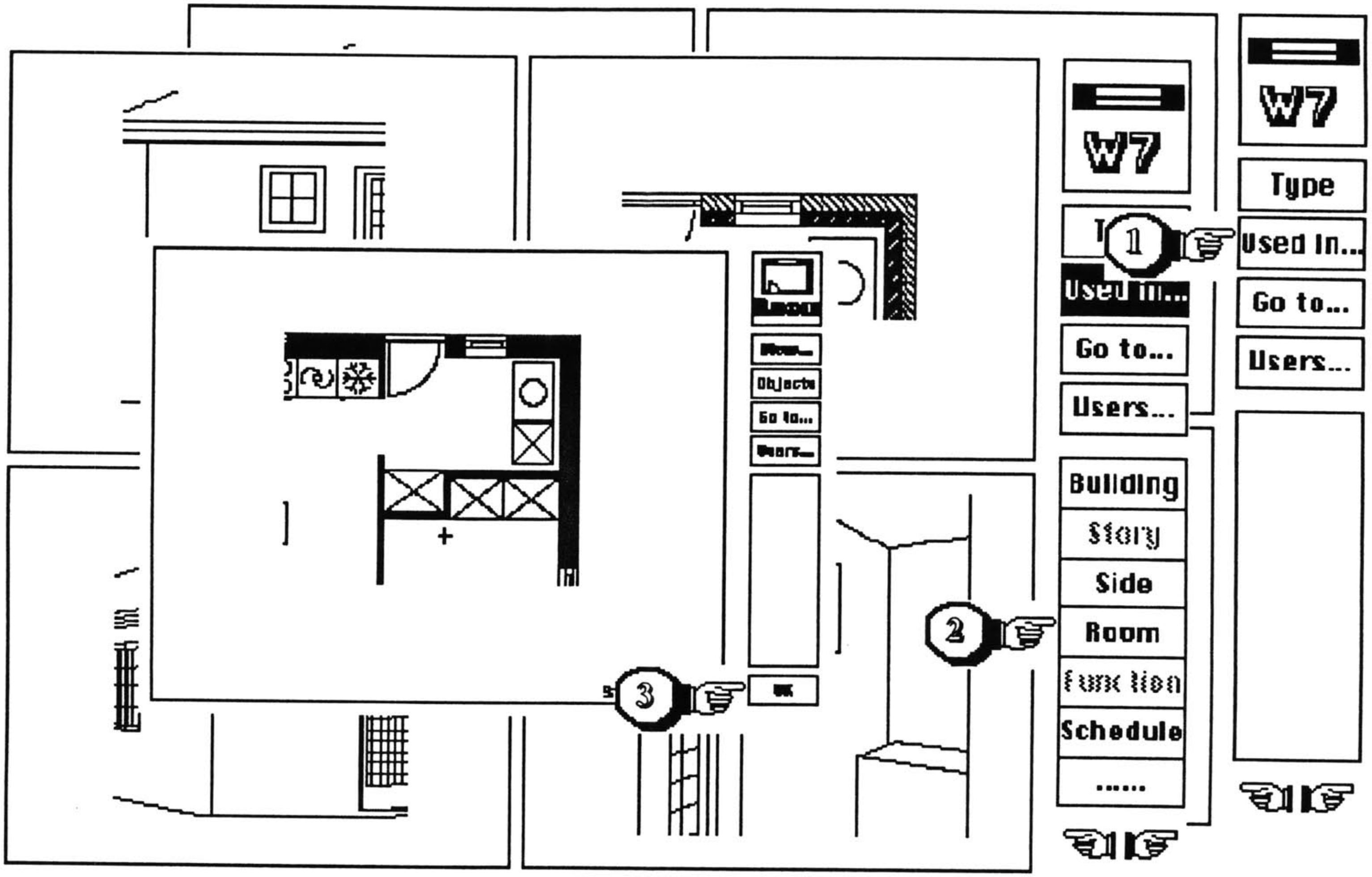


Figure 4.12: Inquiring information about in which room this window is located.

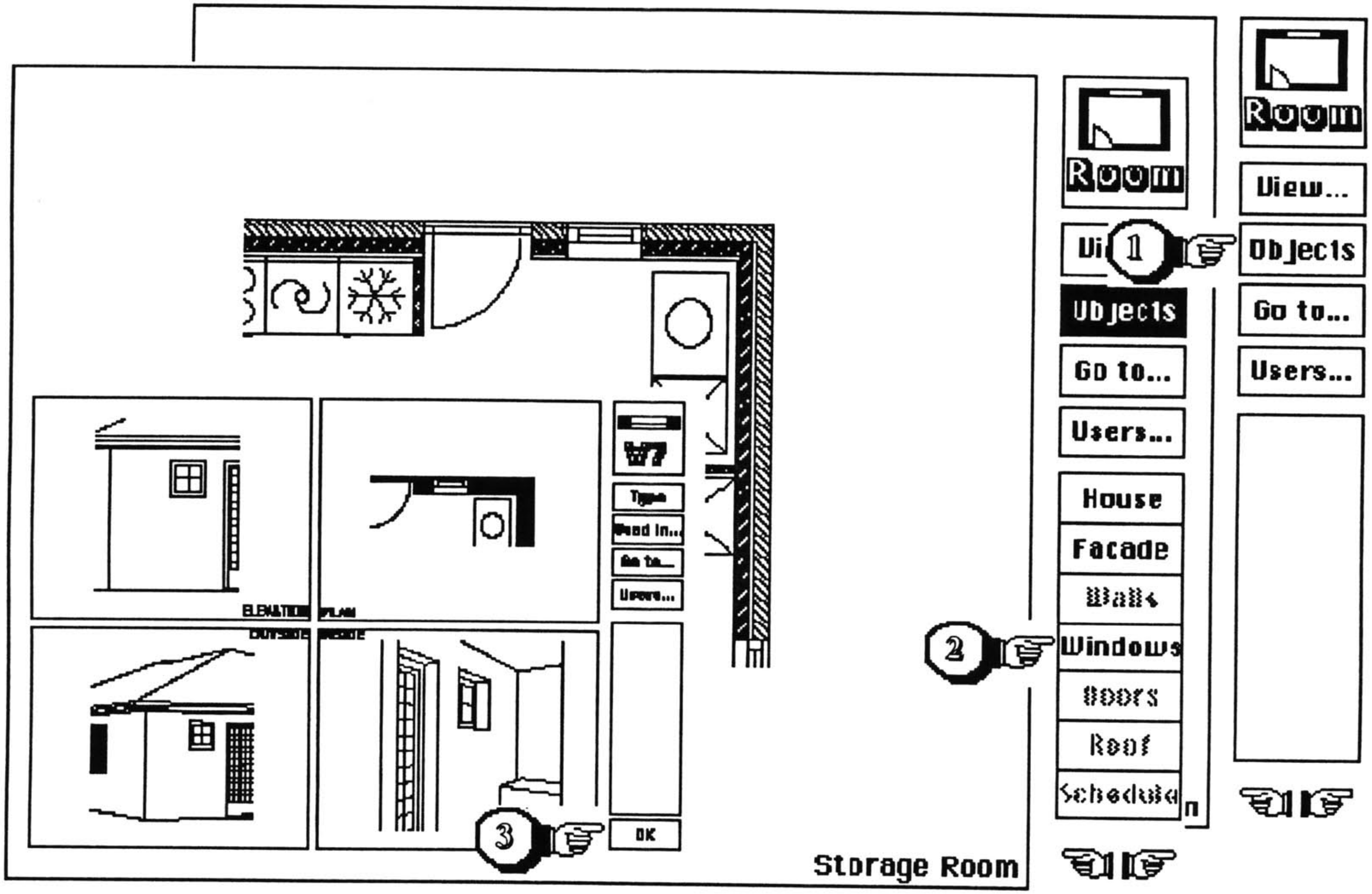


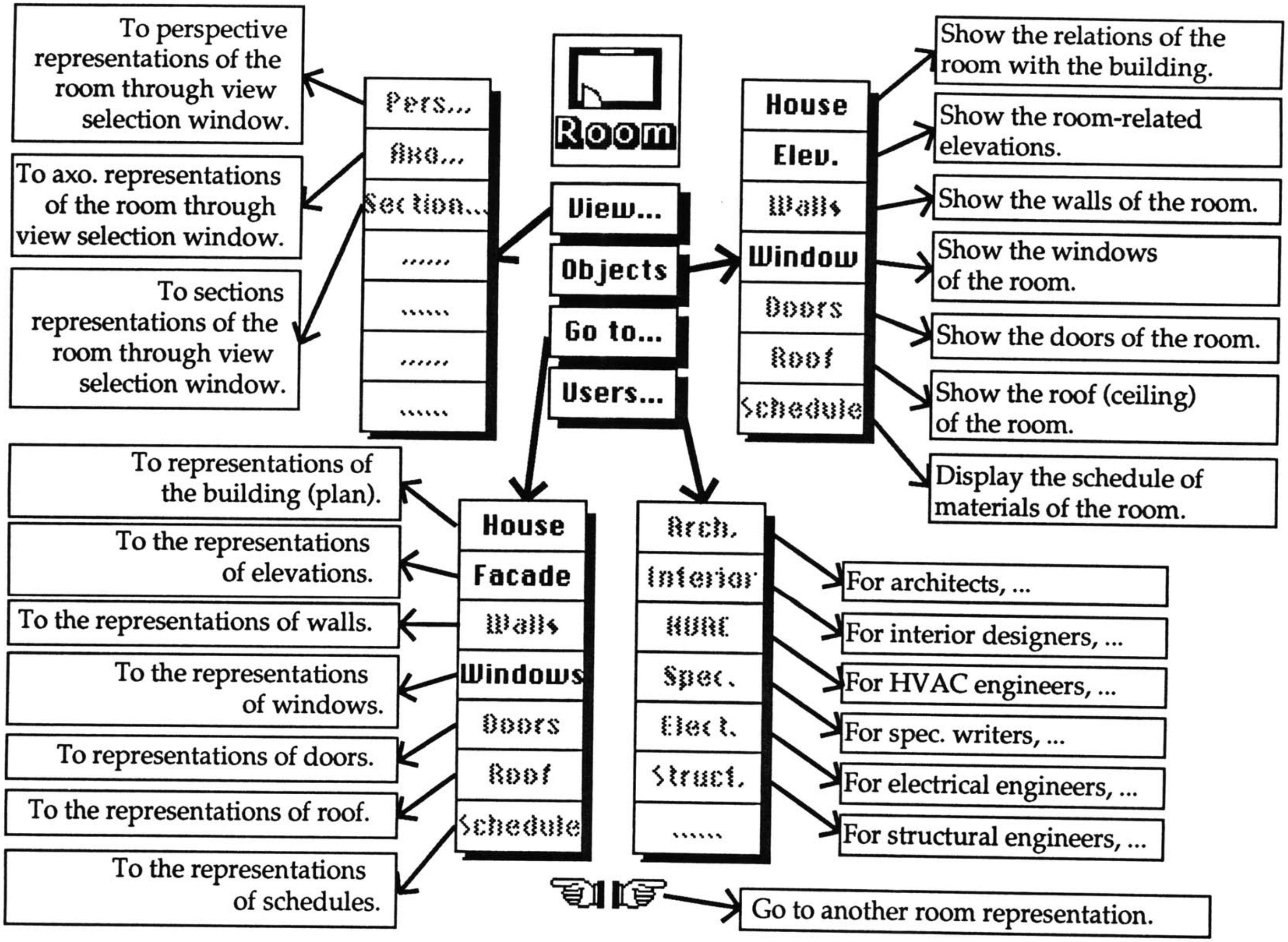
Figure 4.13: Inquiring information about the window objects of this room.

The data organizational structure for representing window information can be used in representing any domain of a building object. In this demonstration model, we can see the structure is also reflected in the domains of elevations, rooms, window type, and window schedule, although only window-related information is included. Conceptually, this structure can be expanded to represent the whole building model. For example, in a room representation, there are buttons containing links with all other representational nodes of the model. It is possible for us to ask for all kinds of relevant information about each particular room (Figure 4.14). Figure 4.15, 4.16 and 4.17 display the links built in the other three types of representational nodes in the model: elevations, window schedule and window types.

One thing should be mentioned about this demonstration model: there are many ways to browse for window information other than by following the examples discussed here. A floppy disk copy of the model is available upon request. It is supported by any Macintosh computer with a Macintosh II screen.

IV.2: Conclusion: problems and future development

With the development of building-modeling CAD systems, data organization is becoming a serious problem facing the CAD systems developers. The conventional methods utilized in the over-lay drafting systems fail to provide appropriate facilities to meet the practical demands of design practice. Under this circumstance, the thesis proposes a possible solution to this problem based on the concepts of hypermedia. The basic idea is to incorporate the organizational structure of nodes and links into a building model, which consequently provides the flexible and interactive browsing links of hyper-



To perspective representations of the room through view selection window.

To axo. representations of the room through view selection window.

To sections representations of the room through view selection window.

Show the relations of the room with the building.

Show the room-related elevations.

Show the walls of the room.

Show the windows of the room.

Show the doors of the room.

Show the roof (ceiling) of the room.

Display the schedule of materials of the room.

To representations of the building (plan).

To the representations of elevations.

To the representations of walls.

To the representations of windows.

To representations of doors.

To the representations of roof.

To the representations of schedules.

For architects, ...

For interior designers, ...

For HVAC engineers, ...

For spec. writers, ...

For electrical engineers, ...

For structural engineers, ...

Go to another room representation.

Figure 4.14: Button nodes for room representations.
(Dimmed buttons are not fully constructed in the demonstration model.)

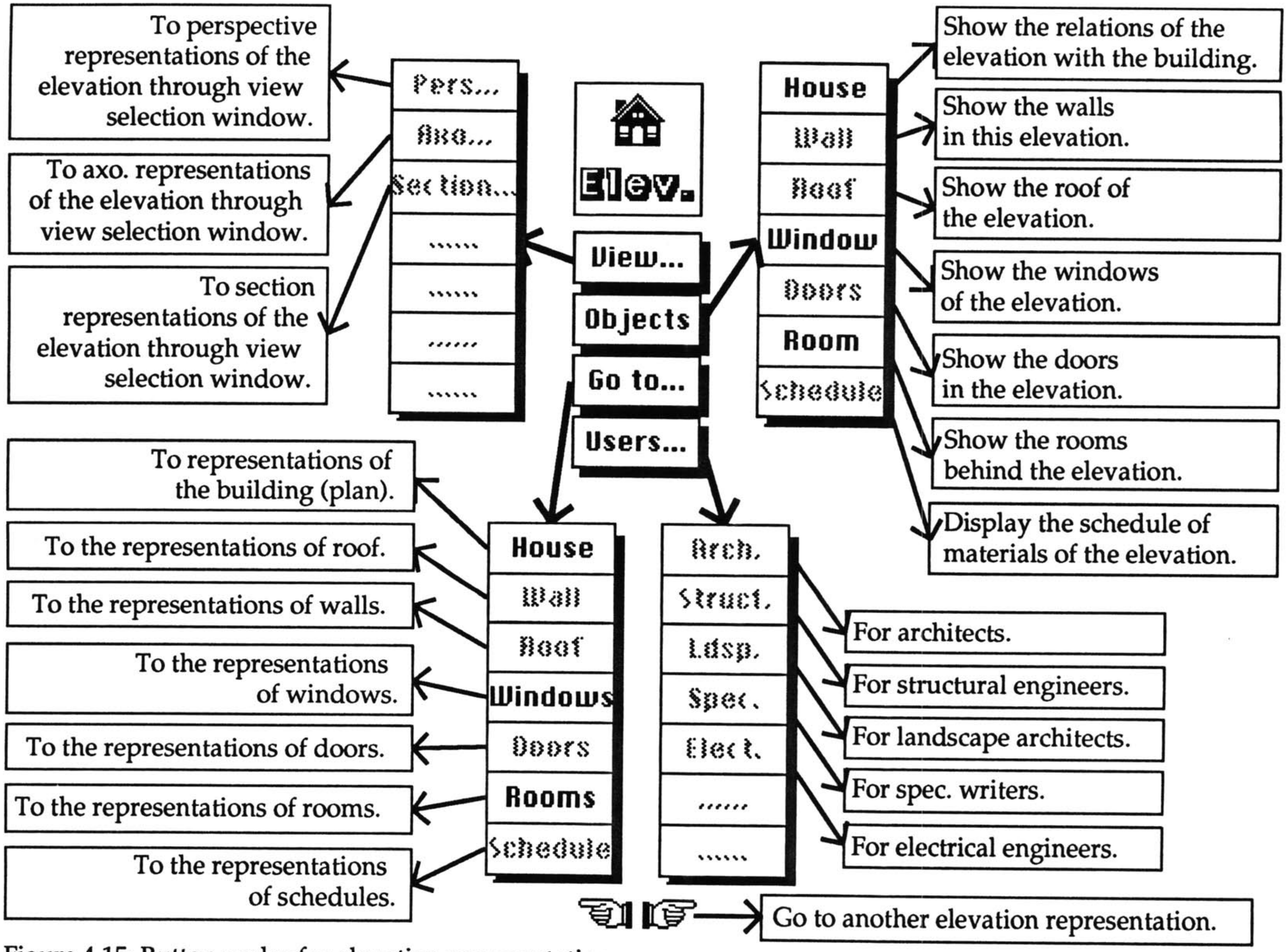

To perspective representations of the elevation through view selection window.
To axo. representations of the elevation through view selection window.
To section representations of the elevation through view selection window.
Pers...
Axo...
Section...
......
......
......
......
Elev.
View...
Objects
Go to...
Users...
House
Wall
Roof
Window
Doors
Room
Schedule
Show the relations of the elevation with the building.
Show the walls in this elevation.
Show the roof of the elevation.
Show the windows of the elevation.
Show the doors in the elevation.
Show the rooms behind the elevation.
Display the schedule of materials of the elevation.
To representations of the building (plan).
To the representations of roof.
To the representations of walls.
To the representations of windows.
To the representations of doors.
To the representations of rooms.
To the representations of schedules.
House
Wall
Roof
Windows
Doors
Rooms
Schedule
Arch.
Struct.
Ldsp.
Spec.
Elect.
......
......
For architects.
For structural engineers.
For landscape architects.
For spec. writers.
For electrical engineers.
Go to another elevation representation.

Figure 4.15: Button nodes for elevation representations.

(Dimmed buttons are not fully constructed in the demonstration model.)

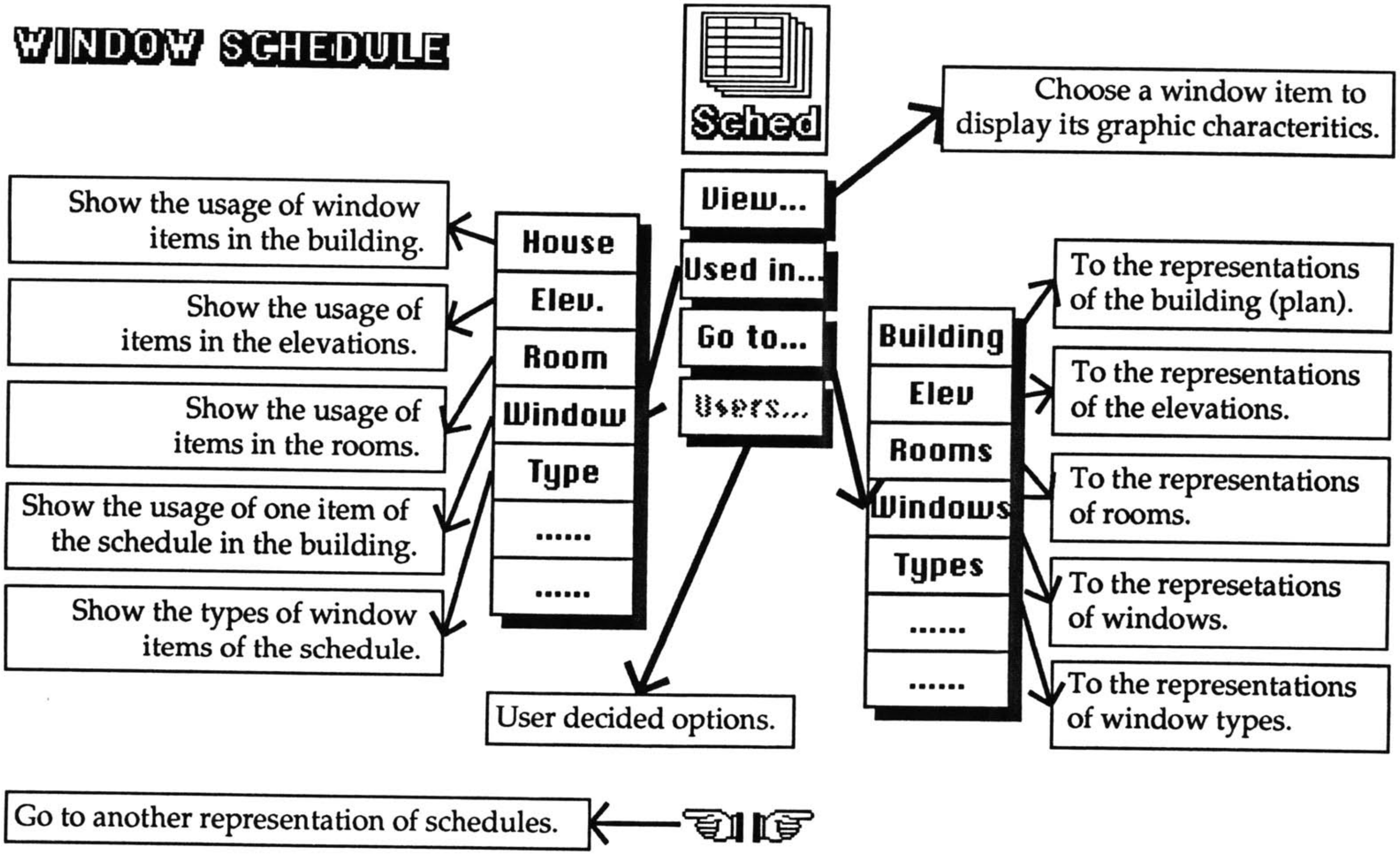

WINDOW SCHEDULE
Sched
View...
Used in...
Go to...
Users...
Choose a window item to display its graphic characteritics.
Show the usage of window items in the building.
Show the usage of items in the elevations.
Show the usage of items in the rooms.
Show the usage of one item of the schedule in the building.
Show the types of window items of the schedule.
House
Elev.
Room
Window
Type
......
......
User decided options.
Building
Elev
Rooms
Windows
Types
......
......
To the representations of the building (plan).
To the representations of the elevations.
To the representations of rooms.
To the represetations of windows.
To the representations of window types.
Go to another representation of schedules.

Figure 4.16: Button nodes for window schedule representations.
(Dimmed buttons are not fully constructed in the demonstration model.)

media to the users of a CAD building model. The possibilities and benefits of a hypermedia-based building model are further demonstrated with a simulated model representing the windows of a single family house.

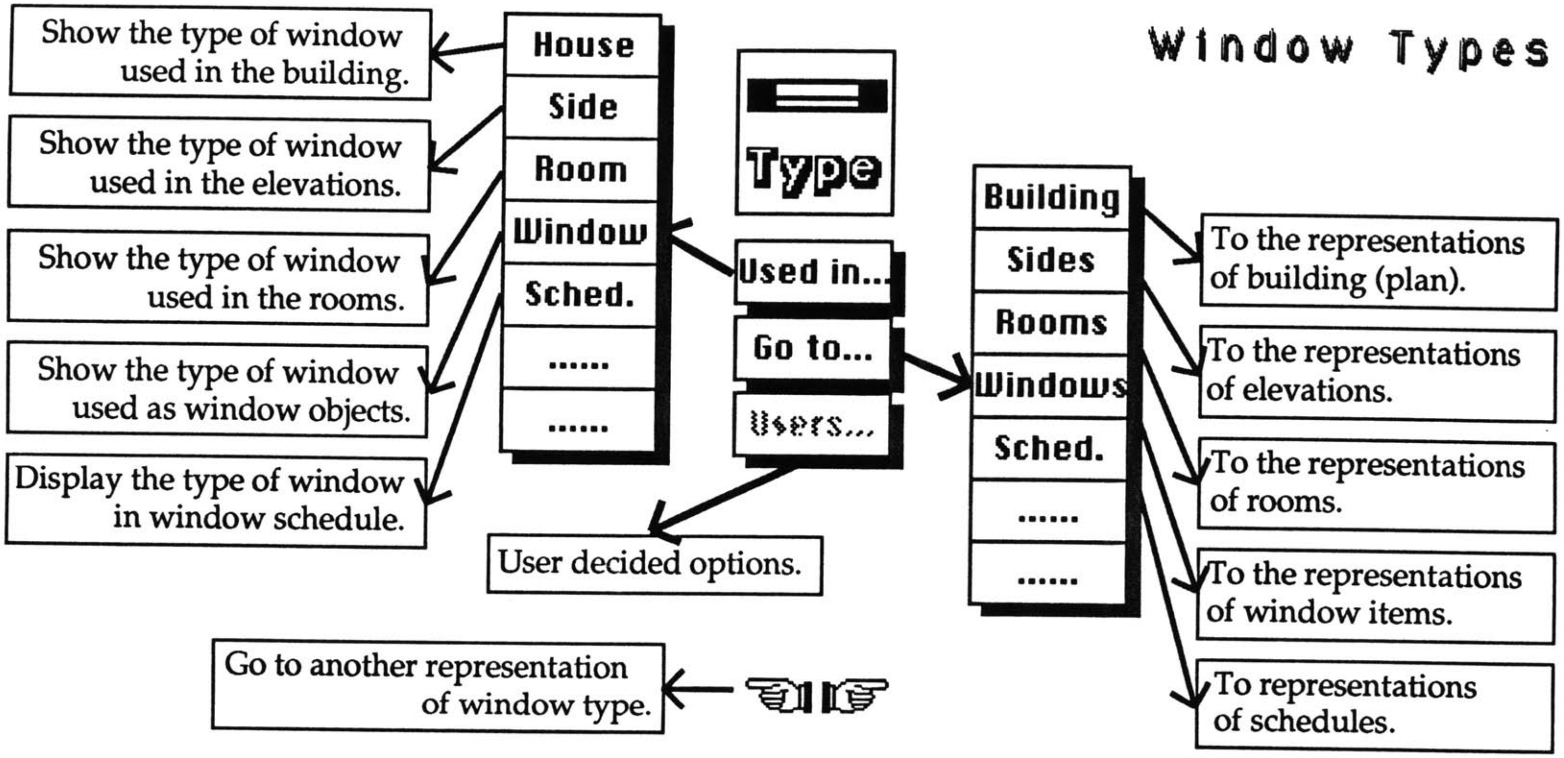

Window Types
Show the type of window used in the building.
Show the type of window used in the elevations.
Show the type of window used in the rooms.
Show the type of window used as window objects.
Display the type of window in window schedule.
House
Side
Room
Window
Sched.
......
......
Type
Used in...
Go to...
Users...
User decided options.
Building
Sides
Rooms
Windows
Sched.
......
......
To the representations of building (plan).
To the representations of elevations.
To the representations of rooms.
To the representations of window items.
To representations of schedules.
Go to another representation of window type.

Figure 4.17: Button nodes for window type representations.
(Dimmed buttons are not fully constructed in the demonstration model.)

Just like the problems of hypermedia, the problems of a hypermedia-based building model stem from the very flexibility which is its chief advantage and justification. From the viewpoint of system design, besides the space we need to store the building object data, we have to create a complicated organizing and controlling structure to represent the relationships among these data and to provide the mechanism for manipulation. More machine power has to be allocated to perform these kinds of "service" operations other than storing building data. As the project scale increases and the design is developed, the flexibility of the organizational structure is counteracted by the complexity, which may make the system less efficient and more difficult to maintain. In the simple window representational model, more than half of the storage space and almost all the manipulating power are used in constructing the manipulating structure. On the users' side, complexity may cause disorientation in browsing through a building model. It is difficult to maintain a sense of where things are in a flexible data structure. It is so easy to move from one point to another that we may forget which part we are working on.

In a hypermedia-based building model, the organizational structure constitutes a major part of the model. In creating this kind of model, we will be faced with the problem of how to allocate the efforts in inputting the structural data and object data. One possible solution is to build the organizational structure of a building model as a separate module of the CAD system. Different types of buildings can be accommodated by different this kind of modules. Each module is designed to meet the requirements of a specific type of building, and it can be added on to the CAD system when used in a specific project. So the work of data input is still mainly concerned with the building object data. Each piece of object data can be input into a cell specifically design for this type of data, and the cell is already connected to other cells according to inherent logic of building elements. In this way we do not have to create those links in terms of data input for every project. Most of the organizational work has been done before a project start. Once we finish a building model based on this kind of module, we may be able to take advantages of this organizational structure of hypermedia without spending more time on creating it.

But there may still be some organizational relations, especially those functional links, which cannot be included in a stereotype module. We may have to spend some time to create this kind of project-specific links. The tools we need may be included in the module. The time investment can be compensated by the time we save in using this kind of CAD systems. The building-modeling method allows us to start the creation of a building model from the early stage of design. Inputting building data becomes an integrated part of design. CAD systems are used just like the paper and physical models. This distributes the work load of CAD applications into every phase of design. The time we spend on producing construction documents can be greatly reduced. Since the model is the consistent information carrier, we do not have to reproduce several drawings with almost the same contents but different purposes. This may also save us a large amount of time.

Another problem is the standard we may use in creating a hypermedia-based building model. The conventions of architectural drawings and other representational documents may only be used as the references for this new structure. Unlike the drafting-oriented systems, which copied almost all the conventions of drawings, the object-oriented method prevents us from adapting these conventions directly. For example, how to represent a window in a building, what kinds of links should be created, and how these links are created are the problems we have not answered with the demonstration model. It displays the methods we can use to organize this kind of data, but the structure is only created on a tentative basis. This does not mean that the proposed hypermedia-based building model will have the same representational nodes or links. In the future, the standard of representing building data in a hypermedia structure may become a problem like the problems of standardization of drawing layers and file format we encounter today.

What the thesis proposes and examines is a CAD system which will stem from the building-modeling systems and be based on the concepts of hypermedia. But the discussion is mainly concentrated on the preliminary concept from the viewpoint of system analysis. It points out and demonstrates the ways to solve the data organization problems we have, but there are many other issues which need to be further examined and discussed on a deeper level. We hope that the discussion and experiment here will be of some value to those further explorations.

Bibliography

Apple Computer, Inc.: HyperCard User's Guide.

Bjork, Bo-Christer: Basic Structure of a Proposed Building Product Model.
Computer-Aided Design, March 1989

Bratko, Ivan: PROLOG Programming for Artificial Intelligence.
Addison-Wesley Publishing Company 1987

Conklin, E. Jeffrey: Hypertext: An Introduction and Survey.
IEEE Journal Computer, Sept.ember, 1987

Coyne, Richard D: Tools for Exploring Associative Reasoning in Design.
CADD Future 1989

Crosley, Mark Lauden: Architect's Guide to Computer Aided Architectural Design.
J. Wiley, New York 1988

Danner, William F.: A Global Model for Building-Project Information:
Analysis of Conceptual Structures.
Report NBSIR 88-3754 National Bureau of Standards, April 1988

Delisle, Norman: Neptune: a Hypertext System for CAD Applications.
Proceedings of the ACM SIGMOD'86

Dubeny, Fred: Perspective and other Drawing System.
The Herbert Press

Goodman, Danny: The Complete HyperCard Handbook.
2nd Edition, Bantam Books, 1988

Gookin, Dan: The Complete SuperCard Handbook.
Computei Books Radnor, Pennsylvania, 1989

Gutman, Robert:
Architectural Practice, A Critical View.
Princeton Architectural Press, 1988

Halasz, Frank G.:
Reflections on NoteCards:
Seven Issues for the Next Generation of Hypermedia Systems.
Communications of the ACM, Vol.31(7)

Kennedy, E. Lee:
CAD Drawing Design Data Management.
Whitney Library of Design 1986

Laseau, Paul:
Graphic Thinking for Architects and Designers.
Graphic Problem Solving for Architects and Designers.
Van Nostrand Reinhold 1989

Lockard, William Kirby:
Design Drawing.
Drawing as a Means to Architecture.
Pepper Publishing 1982

Malcolm, C. Harrison:
Data-Structures and Programming.
New York University 1972

Oles, Paul Stevenson:
Architecture Illustration, The Value Delineation Process.
Van Nostrand Reinhold

Parsaye, Kamran:
Intelligent Databases,
Object-Oriented Deductive Hypermedia Technologies.
John Wiley & Sons, Inc. 1989

Penttila, Hannu:
Computer-Aided Building Modeling.
ECAADE Conference 1989

Radford, Anton:
CADD Made Easy:
A Comprehensive Guide for Architects and Designers.
McGraw-HIll Book Company 1987