Natural language interface for CAAD system
R. Moshe*
E. Shaviv
ACA-Architecture and Computers Aids Ltd. (Israel)
ASU Tempe (U.S.A.)

ABSTRACT

This work explores issues involved in the development of a natural interface for man-machine dialogue in architectural design processes. A hand-touch on an interactive surface is suggested as the best natural-language interface for architectural CAD systems. To allow the development of a rich range of hand-touch natural-language for communicating information and commands to the computer, it is proposed to develop a new type of a touch-panel, for which a set of specifications is presented. A conceptual design of an architectural workstation, having the described touch-panel, is presented. This workstation is characterized by the integration of the entire range of control and communication facilities required for any architectural task into a single interactive unit. The conceptual model for this workstation is the standard size drawing board, on which the architect is accustomed to spread documents, drawings, books and tools, shuffle them around and interchange them freely by using the natural-language interface developed in this work. The potential of the suggested hand-touch natural-language and the proposed workstation are demonstrated by a case-study.

INTRODUCTION

The design of CAAD user-interface embraces four major components: the human factor, the task to be performed, the environment in which the work takes place and the machine.

The human factor may be analyzed according to a model proposed by Bailey, 1982, which comprises the senses, the conscious and automatic cognitive processes, the responders like mouth, hands and legs and the body in general. Each individual has capabilities and limitations with respect to each of these components. These limitations may change either temporarily following tension, fatigue and anger, or permanently as a result of age for example. The design of a user interface should consider these capabilities and the respective limitations.
To be more specific about the human factor we would like to focus our discussion on a specific user group, namely architects. This group is characterized by having academic knowledge, openness and capabilities to apply new methods. Architects tend to have individual approaches and design methods, and like to emphasize personal style, even in the architectural drawings. Architects are visualizers, who deal with shapes in three dimensions and use graphic and other visual aids for design, analysis and for communicating ideas.

Most architects today lack knowledge and experience in computer technology, though this is changing quite rapidly. Their approach to computers varies. Some feel threatened and alienated by the computer and describe it as unsuitable for creative tasks, while others are enthusiastic about the new possibilities. It seems to us that a natural language interface can remove a major stumbling block towards the integration of CAAD systems in the architectural practice.

The task is the architectural design. A design process may not be described as a simple step by step logical sequence, but rather complex trial and error and cause and effect, as well as iteration and convergence towards a solution. Also, architectural design methods are very individual and differ according to the project type. Moreover, large quantities of information are accumulated for any particular project, extracted from a variety of sources, and spanning a wide range of representation methods. This includes text and tables, mathematical models, graphics, 3-D models and pictorial methods. Some of this information can be qualitative and ill-defined. Hence, flexibility and freedom of choice are essential characteristics of any CAAD user interface.

The machine factor for a comprehensive CAAD system comprises of a wide range of graphical input and output components and devices. These components may include a locator, picker, digitizer, valuator and a scanner as input devices, and hardcopy, printer and plotter as output devices.

The environment factor consists of the immediate work environment. In our case this is the architectural office, including the furniture and the environmental conditions. Other aspects are the social and professional environment, and the wider cultural and economical contexts, which are beyond the scope of our discussion.

MAN-MACHINE DIALOGUE

To understand the man-machine dialogue we will use a language analogy. Accordingly four major levels can be defined: the conceptual, semantic, syntactic and lexical (Foley and Van-Dam 1982). The conceptual level describes the abstract model by which the user should view the application, especially the objects he deals with and the relationships between them. The semantic level describes each of the operations allowed by the system. At this level the following components are defined: the information required, the possible errors, and the results of each operation. The syntactic level deals with the sequence of input and output elements, which is the order of words in a legal command sentence. Finally the lexical level defines the actual
interaction techniques and the hardware primitives (called lexemes) to be used, and thus represents the binding of the hardware-independent tokens of the input and output languages, to the actual hardware capabilities.

There are many differences between man-man and man-machine dialogues. The first one is rich and highly dependent on the context. Man-machine dialogue is, on the other hand artificial and covers only a restricted subset of concepts and operations that are pre-programmed in the machine. However, the human who lives in a much wider world, may attach to the messages received from the computer semantics related to his world. For example, a set of lines displayed on the screen is interpreted by a skilled architect as a representation of a building, while the computer "perceives" these same data as a set of geometric entities. Furthermore man-machine communication is asymmetric, since man and machine differ in their characteristics, capabilities and limitations. Hence, we have to deal with two different languages; that of the human and that of the computer. This work will focus on the human language in CAAD systems.

One of the most important types of information expressed by the user is the command expression. Using the natural language analogy, each command expression may be thought of as a "sentence". Thus, a command sentence in CAAD systems is typically composed of two types of phrases which are verb phrases and noun phrases. A verb phrase is composed of a verb and may contain also two types of complements, the object and the descriptive complements. The object is in itself a noun phrase. The descriptive complements are various characteristics of the action noted by the verb, such as the location in the three-dimensional space, the time and quantity, or the conditions for performing the operation. The noun phrase refers to an entity, like a file, a geometric entity or a menu item.

The "terminal elements", namely the most elementary units comprising the command sentence, are the "words", which can be a verb, like INSERT, DELETE, or a complement, like the identity of a geometric entity, a text string or a file name. It is important to understand that these "words" in the specific command of the human language (in CAAD systems) need not necessarily be expressed verbally (like orally or by typing them). Rather, they may be conveyed lexically, using a variety of media and techniques that the CAAD system supplies as a user interface. For example, a geometric entity displayed on the graphics screen may be referred to by a picking device, by typing its name on the keyboard or by selecting its name from a menu.

The picking and the selecting can be performed in different ways, out of which the most natural interface are spoken language or direct hand-touch of the object displayed on an interactive screen. The hand-touch operations tend to be in CAAD systems more efficient by conveying several components of the command sentence in one operation. For illustration, dragging an object conveys simultaneously the operation type, identity of the movement and final location. This means that the verb and the compliment of the sentence are presented in one operation. Additional natural operations can be carried out by natural hand-touch language, as will be presented in chapter 4.
WHAT IS A NATURAL LANGUAGE INTERFACE FOR CAAD SYSTEMS?

A major concern of current designers of user interfaces is to achieve more efficient, consistent, comfortable, transparent and natural interface between the people and their computer. Furthermore, we would like to achieve all of the above and at the same time to ease the learning process. Let us elaborate in some details the meaning of these terminology. Considering the four major components of CAAD user interface which are; the human factor, the task, the machine and the environment (as was explained in the introduction), we shall define the above terms as follows:

"Efficient" means that there is no need for a plenty of computer resources, i.e. memory and time, to parse the user sentences and convert them into the proper sequences of computer instructions. Moreover, the performance should be fast to allow short response time.

"Consistent" denotes that the user interface for CAAD systems should be designed according to a predefined coherent plan, capable of creating a clear conceptual model in the user's mind. There should be an invariability in sequence and structure of the commands patterns, terminology, syntax and logic.

"Comfortable" implies the suitability of means and methods to the characteristics of the various architectural tasks and to the user skills and level of knowledge. It also indicates the need for a completeness of the user set of commands, freedom of choice and flexibility, with the ability to use a variety of responders.

"Transparent" means that one can use a tool or a medium without conscious effort in controlling it, as if it were an organic extension of his body. In such a state the human can allocate most of his cognitive resources to the task at hand. Transparency is not a feature of the medium itself, but is rather a description of the way the human relates to it, after reaching a certain level of skill. This is the way one is driving a car, speaking his mother language or sketching with a pencil. All these skills are not easy to acquire, and are certainly not transparent in the early learning stages. Obviously consistency in terminology, methodology, syntax and logic help to achieve transparency. We might say that the goal of skill acquisition is to bring the trained to this target state. The issues of training time and effort needed to achieve this state are also important, though they must be weighted against other performance criteria.

"Natural" implies familiarity, similarity to previous habits, relying on concepts already known, and also capability of becoming transparent. A medium which is natural to one person is not necessarily natural to others. For instance: mathematic formulas are natural to mathematicians, flowcharts and mnemonic codes are natural to programmers, while architectural drawings are natural to architects. Our goal is to design the user interface for CAAD systems to be natural and consistent with the professional world of the architect. The obvious solution is to try to imitate in the user interface the same tools and
methods used in the conventional design process. However, although a simplistic simulation of familiar methods is easy to learn, it may also prevent us from fully utilizing the potentials inherent in the modern technology. This may prove useful as a short term strategy, but in the long run new methods and tools must be developed, and will probably become natural to the architect not less than the current tools.

Natural language interfaces, like English, French or Hebrew, are frequently proposed as a solution to the problem of user friendliness. We would like to raise few questions; is a natural language a viable means for a user interface in the context of CAAD? Is a natural language really "natural", transparent and consistent? Observing any natural language one can easily find out plenty of deficiencies regarding those criteria. To name just few: inconsistency, rules and exceptions, changes of meaning according to the context, complexity, huge vocabulary, vague meaning, influences from various languages, synonyms and related meanings, as can be demonstrated by the Thesaurus.

A language evolves through generations by a large number of people without any predefined plan and is continuously changing. Therefore, there is no rigid format, or syntax that can make it easier for the compiler to process a natural language to a proper sequences of computer instructions. This means that efficiency is not easy to be achieved and large databases of explicit world knowledge should be provided to assist in parsing the natural language sentence as well as in interpreting it. Moreover, if the system should recognize spoken words, than on top of all the above difficulties presented in understanding natural languages, exist problems that microphone and the background noise introduce interferences into the recording of the spoken utterance. If the system must recognize words spoken by more than one user, than the task is even more difficult (Barr and Feigenbaum, 1981).

We believe therefore, that (at least today) "speaking" to the computer in one of the "natural" languages cannot serve as the best user interface for CAAD systems. This is not only because of all the deficiencies discussed above and the considerable computational resources needed to process a natural language input, but also because it is much easier to use hand-touch as a natural language interface for CAAD systems than talking to the computer. Take for example the command sentence: "Take the cube that is above the yellow pyramid that is to the left of the red cube and put it below the green prism and to its right". It is much shorter to say: "Take this and put it here". The words this and here can be understood if you touch the object you want to move and than slide it to the place you want to locate it, and not if you pronounce it verbally.

To summarize, it seems to us that a hand-touch language for CAAD systems can be more natural than talking a "natural" language like, English, with the computer. This is because picking, replacing, stretching a window and so on, are performed faster and more naturally by indicating with the hand the object, or defining the exact size of window that is needed than describing these operations verbally. The following chapter will discuss the potential in the development of a rich hand-touch natural-language, and the requirements for achieving such a language.
INTERACTION BY HAND-TOUCH LANGUAGE.

To allow the development of a wide range of hand-touch natural-language for information communication and computer commands, it is proposed in this work to develop a new type of a touch-panel, which is able to utilize characteristics of human hand-touch, not available commercially today. These additional characteristics can be either static or dynamic. The following is a set of specifications for such a touch-panel.

The static touch sensitivities should include: - Accurate identification of the point touched, - Pressure intensity differentiation, - Duration of pressing, - Multi-touch sensing (simultaneous touch of several points), and - Shape recognition of the area touched (defined by a set of points).

The dynamic touch sensitivities should include, in addition to the static capabilities demands, the following: - The course of movement, - The speed and the intervals separating discrete sequential touches, and - Dynamic activation of various zones of the work surface.

Several hand-touch operations were developed in this work to illustrate the features and potentials of the proposed touch surface. These hand-touch commands were designed to imitate natural operations by an architect organizing his drawing table while working on his design.

The hand-touch commands are based on four types of touch which are: - "Touch" (T) means a short and soft touch, - "Press" (P) means a touch with press on the surface, - "Dynamic" (D) stands for dynamic touch i.e. sliding the fingers on the surface, - "Sequence" (S) implies sequence of touches.

Shapes of contact areas were also distinguished and include: - one finger (.), - several fingers in a row (elongated shape) (...), and - a touch with a palm (.PL).

Following are some typical examples of general purpose hand-touch operations. Additional examples are demonstrated in Ben-Moshe, 1986.

"Dragging a document" is achieved by pressing the document with several fingers and then sliding the fingers in the desired direction until the required location is reached.

"Exposing a document or a drawing covered by another" is performed by pressing a finger within the limits of the currently exposed part of the document. This operation results in "jumping" this document to the upper level without affecting its position on the surface.
"Turning a page" is done by sliding a finger softly on the lower corner of the displayed page (the right corner for next page, the left corner for the previous one). A fast sequence of such touches at the same corner will initiate a continuous sequence of page turning, stopped by one press of a finger at the same corner.

"Scrolling" can be done by sliding a finger along one of the vertical margins of the document, simultaneously echoed by scrolling of the document, forward or backward accordingly. In this operation, a sequence of short slides results in a continuous scrolling, stopped by a short finger press.

"Changing the vertical size of the exposed part of the scroll" (namely the number of lines displayed simultaneously in the view-port) is performed by pressing one finger at the center of the upper margin and one finger at the lower margin, and then sliding the fingers in the required directions to stretch or to shrink the window.

"Changing of the size of a view-port together with the scale of the displayed picture" is performed in a similar way to the previous operation, except that two opposite corners should be touched.

"Canceling the display of a document or a drawing" namely, taking it off the table, is performed by sliding a finger in an X-shape upon the entire view-port of the document.

Hand-touch operations of this type are especially appropriate for manipulations of large objects and for performing operations that do not require accuracy. However, the same medium can be used to express approximate quantitative information like much/little, fast/slow, more/less, by the pressure, speed or rate of discrete touches used. More accurate quantitative information may be expressed by a circular finger movement on a displayed dial or on a linear ruler.

Development of such a language will probably require sophisticated techniques of artificial intelligence and pattern recognition. It is difficult to characterize, identify and unambiguously decode attributes like the shape of area touched, and the dynamic properties of touches. It is also difficult to distinguish between meaningful gestures and random touches (background noise). The last problem mentioned may be simplified if certain constraints are applied like dynamic activation of limited zones of the surface. Namely, allocating only certain active windows to be touch-sensitive at any point of time, while touch at the rest of the surface is ignored by the system. Another possibility is to treat differently simple and complex operations. Simple operations can be performed immediately, while more complex operations should require reconfirmation by the user like touching a PERFORM button.

In addition to the general purpose hand-touch operations, one can develop any other hand-touch operations that are specific to a particular architectural task. Using general and specific hand-touch operations, a test-case was
developed to demonstrate the natural and easy way of communicating with the computer provided by this approach. This case-study include the creation and retrieval of architectural programming data-base, viewed as a module of an integrated CAAD system. The architectural program is conceived as a tree structure of sections in several levels. Each section is referred to by its title, and these titles are displayed on the screen and manipulated like magnetic strips on a metal board (see Figure 1). The architect is able to rearrange the order and the hierarchy of the sections by touching their titles and dragging them across the window of the table of contents. The system responds by automatic hierarchic numbering of sections and subsections, and this table of contents actually controls and reflects the current structure of the document. Accessing a specific section may be done by touching its title in the table of contents, and the user can either look at a full table or only at a certain level of hierarchy, displaying for example only the titles of the top-level sections. The table of contents may be related to both as a sequence of discrete pages (turned over by touching a corner of the page), or may be scrolled by sliding the finger along one of its vertical margins as was shown above. A full description of this case-study can be found in Ben-Moshe, 1986.

Figure 1 Structuring process by manipulation of titles in the table of contents
CONCEPTUAL DESIGN OF A WORKSTATION FOR ARCHITECTS

A conceptual design of an architectural workstation, having the described touch-panel, is suggested. This workstation is characterized by the integration of the entire range of control and communication facilities required for any architectural task into a single interactive unit. This unit includes the computer equipment and furniture dedicated to a single person at the architectural office. We allowed ourselves to designed this workstation with some degrees of freedom, that are not possible in a commercial environment, regarding available hardware technology and price constraints.

Different considerations in the design of the workstation were taken into account. The workstation is to be used interactively for a variety of work modes and information types, which require specific devices and communication means simultaneously. The workstation should be able to support an integrated modular and open software system, and to enable the architect to access a wide variety of applications, sharing a consistent user-interface. Multi-tasking using different windows should be catered for. The workstation is normally used by one person at a time, but a team work and group discussions must be created for. Typical work-sessions are long and require a considerable creative and intellectual effort, hence the equipment and furniture should be integrated and spatially concentrated, to reduce the need to run between devices, and enable the user to focus his attention to a limited space. Operation should be functional and aesthetic and quit the atmosphere of the architectural office. Expensive input and output devices, which are less frequently used, may be shared by several users or offices.

Taking into account all the above considerations, the conceptual model for the proposed design of architect’s workstation is based on the standard size drawing board on which the architect is accustomed to spread documents, drawings, books and tools, shuffle them around, and interchange them freely by using the natural-language interface developed in this work. (see Figure 2). This standard size drawing board is a multi-purpose interactive touch-panel surface as was described in the former chapter.

The functional components of the workstation include the information display, the operational controls, the input and output components and the machine resources. The first three components are implemented by the touch-panel drawing board, which is the information display, as well as the operational control buttons that are dynamically displayed on it. It also serve as the input and output devices. These three components will be described shortly in this work. A full description of the workstation suggested can be found in Ben-Moshe and Shaviv, 1987.

The information display that is designed as a touch-panel drawing board with dimensions of 70 X 120 cm, allows a simultaneous display of the reference documents like architectural brief and bill of quantities. It also allows the display of graphics like drawings, maps and 3-d geometric model of the design. This work surface should be thin for ease of manipulation and slope adjustment, up to vertical positioning for group presentations. Other features required are: Real time dynamic display for purposes like simulating movement
Figure 2  A conceptual design for a CAAD workstation

Figure 3  A typical layout of the main work surface. A is the main control board. B are the local control buttons. C is a standard alphanumeric keyboard. D is 2-D graphics output, and E some documents.
through building spaces, scrolling and dragging. It also required high resolution and large variety of line fonts, line thickness, color and letter fonts, as well as colored and toned polygon representations and both interactive touch sensitivity for hand-touch comands and stylus-like input device for accurate input.

In addition to this large work-surface, a standard alphanumeric display with a keyboard is supplied for tasks requiring large amounts of data entries. Transfer of information between these two media must be immediate.

The operational controls are the set of tools available for command expression and for the display of the operational feedback, like error messages and prompts. The main considerations for the design of these controls were efficiency and effectiveness in command input, flexibility for change and development and suitability for use in an integrated environment allowing for multi-tasking in different windows. At each stage only the relevant controls and options should be displayed, close to the focus of interest. Two major controls were designed: the main control board and the local control buttons. Both are programmable graphic and alphanumeric objects displayed on the work surface, and operated by touch (see Figure 3).

The main control board serves for command interaction, especially for the general commands not covered by the local control buttons, like system commands. This board may be freely dragged on the work surface enabling the user to bring it close to the objects he deals with. A typical format includes global command buttons which are frequently used by most applications, like HARDCOPY, REGRET, HELP, and SAVE. It also contains temporary special-purpose buttons, windows for the alphanumeric command input and for system messages (see Figure 3).

Local control buttons are specific to the current task performed at a given window and are adjacent to that window, or sometimes "popped-up at the current operation zone, as was seen in the case-study presented in the former chapter (see Figure 1). Each button is labeled by a word or an icon associated with a command expression component, such as a command name, a parameter or a menu item.

The multi-purpose interactive touch-panel surface can serve as the Input devices as it can replace a digitizer, a picker and also a valuator. The valuator is obtained by touching a programmable dials and linear scales, while a digital representation is echoed on the screen. It also designed to serve as the output device by defining a window on the hand-touch drawing table and getting hardcopy of all the information included in the defined window. The user can also indicates the scaling factor for the hardcopy to be produced.

The design proposed here is at a conceptual level and focuses on the functional and general shape specification, and planned to provide the performance of all the specific tasks of the architectural design processes. We believe that the suggested architectural workstation along with the hand-touch natural language interface have the potential to naturalize the CAAD systems in any architectural firm and turn them to an integrated tool in the design process.
REFERENCES


