SPEECH-DRIVEN COMPUTER-AIDED DESIGN
Innovation in human-computer interaction

H.NEUCKERMANS, T. DE RIDDER, B. GEEBELEN
KU Leuven - Department of architecture
Kasteel van Arenberg
B-3001 Heverlee
Belgium

Abstract The use of computers in architectural design has shifted from what initially was a very numerical approach into a graphical way and thus into a designerly way of working. Through the extensive use of WIMPs CAAD packages today are more or less user-friendly. In the meantime the ever increasing functionality of those packages inflate the number of commands, actions, options and choices which have to be activated with the mouse in order to generate a structured drawing. As a result packages become rather cumbersome to handle.
In this research we have explored a new medium, namely speech, to tell the machine what to do. With software for speech recognition and making use of a head-mounted microphone we have built a far more user-friendly way of handling a CAD package. Grids, snap, ortho, layers, settings and other commands that can be used in combination with mouse actions are activated and deactivated by voice command. We are convinced that this is a step further towards a more easy and natural way of using computers while designing and certainly a way for speeding up the modeling of real architecture. The experiment has been conducted for AutoCAD with the software for speech recognition by Lernout & Hauspie. This new way of working is not limited to architecture and the principle can be used right now for any other package, provided it has a programming interface.

1. Speech, images and design

More than 50 years ago the Von Neumann machine was conceived by engineers for engineers, that means for number crunching. Fortran could handle integers and floating-point arithmetic. As for character manipulation and graphics, everything had to be expressed in punch cards.
Together with the ambition of the computer industry to win the world, user-friendliness became the motto and the motor for developments into graphics and windows, icons, menus, and pointers (WIMPs). Today's CAD packages fully exploit these appealing features, while many engineering applications like CALM ¾ a Finite Element Modeler ¾ or like PHOENIX ¾ a Computational Fluid Dynamic FEM program ¾ still produce pages and pages of numbers. They are still victim of their past enclosed as they are in numeric thinking.
Graphicacy, a neologism introduced by N. Cross (1982), is what characterizes designers in general and architects in particular. The architect thinks and works with and within images, the drawing is his silent interlocutor. The

CAADRIA '98 : Proceedings of The Third Conference on Computer Aided Architectural Design Research in Asia.
eds. T. Sasada, S. Yamaguchi, M. Morozumi, A. Kaga, and R. Homma
April 22-24, 1998. Osaka University, Osaka, Japan. Pp. 309-315

drawing is the mirror of his mind. In his drawing the architect sees what he is thinking, sees what to keep, what to reject, what to change and how to change it.
Software developers and architects have met in graphics.
Today, CAD packages compete in graphic modeling capabilities and seem to produce a never-ending boom of alternative ways to have things done.
This results in an ever-increasing number of commands, in a multiplication of pop-up menus to walk through, in a cascade of nested windows to browse in order to activate this or that precise command. It forces the designer to think in terms of 'clicking the mouse'. What we propose is that the designer would think aloud making use of a head-mounted microphone. Speech is a very natural extension of his designerly way of thinking, because all or most of the drawings an architect makes are intentional, whether we think of designing real architecture or VR; intentional because design is 'reflection-in-action', according to D. Schon (1985); intentional because we model our buildings making use of primitives like vertices, lines, surfaces, solids and because they have a semantic value; they mean: wall, floor, window,...
Our proposal is to introduce voice commands, enabling the designer to express his intentions. It will definitely be helpful if we could shortcut the mouse clicks by increasing the width of the channel through which we can communicate with the CAD package: mouse plus voice. While the mouse has a great chance to lose its privileged role, because of the further development of other devices like (for example) the SMARTpen ™ ¾ a wireless force-sensor based pencil shown in figure 1 ¾ developed by IMEC, speech interfaces are bound to gain momentum because speech is an easy and natural way to communicate.
Our ambition in using speech is twofold:

- speed up the drawing activity and doing so, increase the efficiency;
- realize a more natural and versatile way of working for the architect.

Doing so the architect becomes visibly a part of the high-tech modern society.

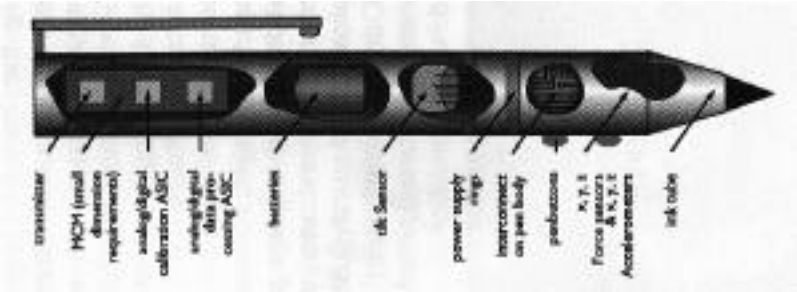

Figure 1. SMARTpen ™ (IMEC)

2. Implementation

2.1 HARDWARE

The hardware used is a Pentium 200 MHz MMX, 64 MB RAM, 4 GB hard disk; Soundblaster 16 (sound card); Shure WH10 (head-mounted microphone.)

2.2 SOFTWARE

We took AutoCAD R14 as an example for experimentation and have implemented, by means of a test, a few paradigmatic voice commands. The software used for realizing the speech interface was the Automatic Speech Recognition (ASR) Software Development Kit for Windows 95 version 3.0 by Lernout & Hauspie Speech Products (may 1996).

That speech software can operate in 2 modes. Firstly, the speaker independent mode, which makes use of a language upon choice and a dictionary, without requiring training of the system. Secondly, the speaker dependent mode, allowing the introduction of new words and requiring word training by repeating each word 3 times. Continuous or permanent speech recognition also requires the choice of a context; here L & H offers 3 alternatives: keyword spotting, connected digits, isolated words. 'Keyword spotting' is identifying a keyword in a sentence. 'Connected digits' allows communication of separate ciphers. 'Isolated words' expects separately pronounced words and this is what was used in our application.

ARX (AutoCAD's Advanced Runtime eXtension) can handle direct function calls to AutoCAD; it is a Dynamic Linked Library sharing AutoCAD's address space and is in fact the successor of ADS and AutoLISP after Release 12.
An object-oriented program *SPEAK*, written in Microsoft's Visual C++ 5.0, uses libraries from both AutoCAD and Lernout & Hauspie's ASR SDK. It runs in a permanent loop, listening for the occurrence of voice input. Whenever speech is detected, the utterance — containing the recognized data — is passed on to a callback function that sends a command request to AutoCAD. A separate module has been written as an object-oriented version of the subset of ASR types and functions used in *SPEAK*, preparing it for future object-oriented recognition engine updates and allowing a more elegant link between ASR and the already object-oriented *SPEAK* module.

3. Applications

In principle any command click can be replaced by voice input. Today CAD packages are completely oriented toward the use of the mouse, therefore we can distinguish two tracks in developing speech interface for design: firstly, facilitating the use of the existing commands; secondly rethinking a complete new way of commanding and thus other implementations of actions to be undertaken when building a structured drawing.

3.1 IMPROVING THE EXISTING COMMANDS

Let's look at some examples illustrating the first ambition of a *more user-friendly interface for the existing command structure*.
- Although 'on and off' toggles like < grid >, < snap >, < ortho > can easily be hit on the toggle bar, they all have that nasty inconvenience that they require you to leave the actual drawing position to reach the right button. So, in general all commands for which you have to leave your actual drawing position, are primary candidates for substitution by voice input. In a first attempt we focused on those commands that are used over and over again, and that somehow can be seen as a kind of interjection; think of < escape >, < cancel >, < undo >, < regen >, < oops >. Another problem is that the dialog box or the menu that pops up when clicking the right mouse button in combination with Ctrl, frequently hides the actual drawing position. If we allow the user to select an option aloud, the cursor can stay where it was and the user can continue drawing straight away.
Example: hitting < osnap > (objectsnap) on the toggle bar displays a large dialog box covering the actual drawing position, when using the pull down menu you have to leave the drawing area of the screen. Figure 2.

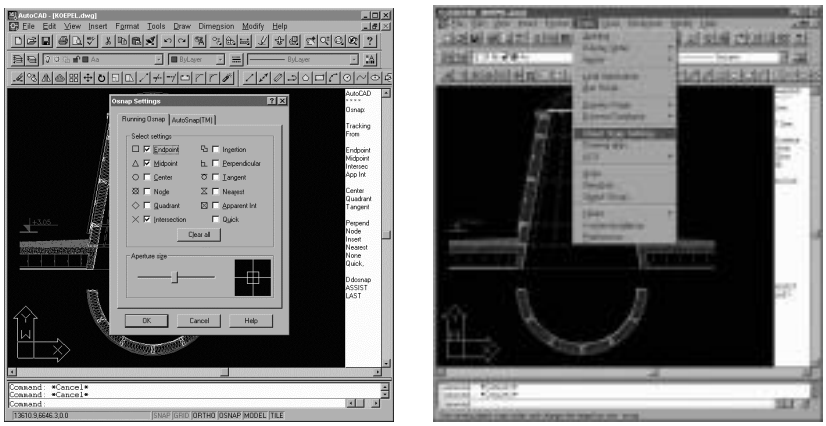

Figure 2. Leave position

- Voice input can eliminate or reduce the pollution of the drawing area with a multitude of pull-down and pop-up menus. The drawing area ¾ which was already small ¾ is indeed becoming an even smaller part of the screen area with every new release. Figure 3.

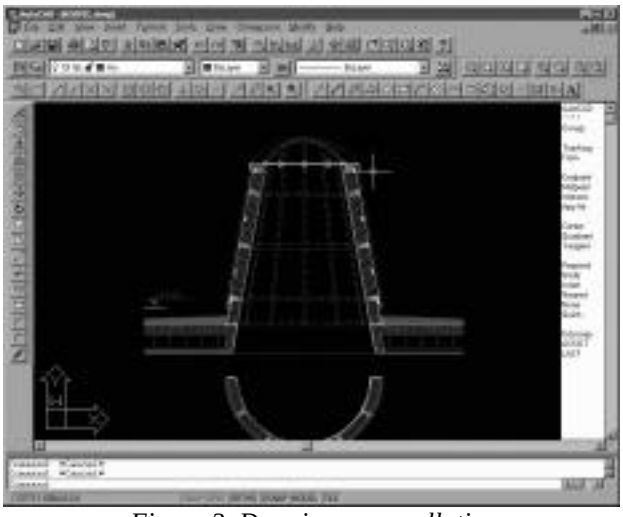

Figure 3. Drawing area pollution

- Some commands require 3 or more choices to be made; voice input can be used as a shortcut. Figure 4.

Example: spoken < planview world> instead of:

< view > → < 3D Viewpoint > → < Plan View > → < World UCS >

- Voice input with mnemonic keywords can help or is easier to remember

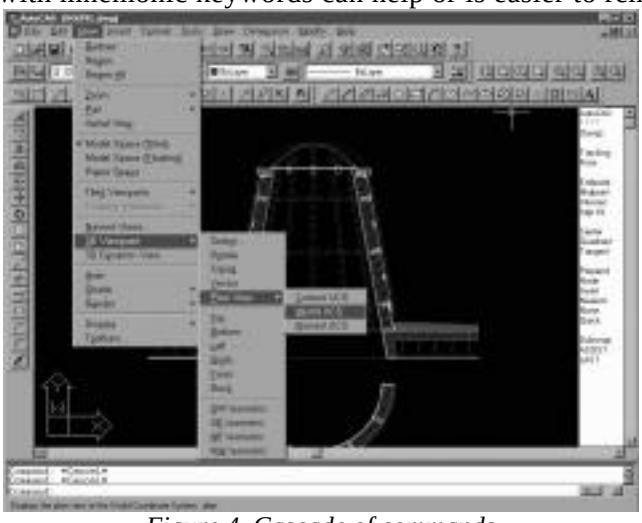

Figure 4. Cascade of commands

than the multitude of icons for the ever-growing number of options within some commands. A combination of speech and icons will be easier. Figure 5.

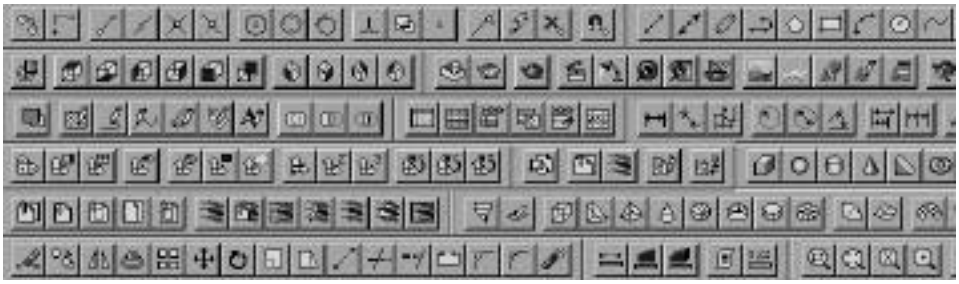

Figure 5. 'Unidentified' icons

3.2 NEW COMMANDS

An example of *a complete rethinking of a command* for speech would be a new < zoom > function. Release 14 has the < zoom realtime > option. It blows up (or reduces) the drawing gradually from the center of the screen. Doing so that particular part of the drawing you are looking for, swings out of the actual viewport. A rewritten zoom could react on spoken < zoom in> or < zoom out > centered around the cursor, and zooming as long as you press the space bar. The same would apply to the < pan > command. Another application could be the automatic adjustment of the viewpoint perpendicular to the user coordinate system (UCS).

If we move into design aids, rather than the drawing aids described so far, a new world of opportunities for speeding up the design process using speech emerges:
- custom designed building elements can be called: < outer wall >, ...
- the intelligent zoom (Neuckermans, 1992) could be implemented: a zoom that displays other things when jumping into another level of detail or design scale. Figure 6.
- implementation of some of the ideas proposed by Cicognani and Maher, as far as concerning us, those that can be used in designing real architecture.

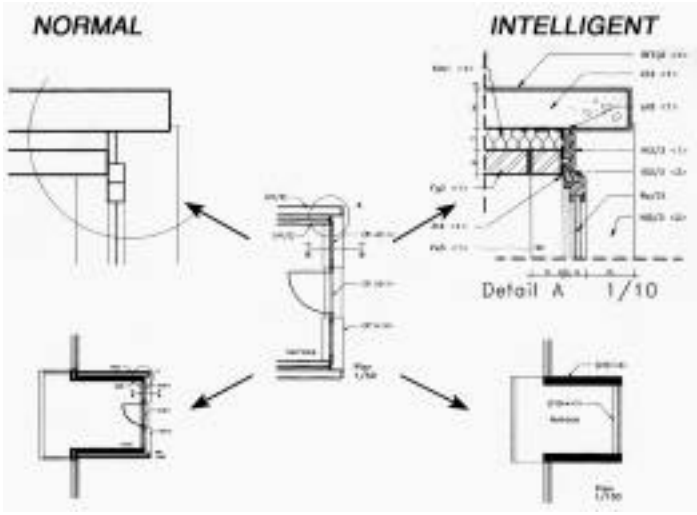

Figure 6. Intelligent zoom

4. Conclusions

We have implemented a speech interface for CA(A)D and explored how meaningful its use can be. The results are not so spectacular yet, mainly because today the execution time of the speech interface comes on top of the existing one and is not fully compensated the time benefit resulting from the speech shortcut. This is however only a temporary handicap until software will be (re)written for speech ab ovo. We are convinced that a trend has been set and that from now speech will be in the air forever.

References

Cicognani, A., and Maher, M.L., 1997, Design speech acts. “How to do Things with Words” in Virtual Communities, *in* Junge, R. (ed.), *CAAD Futures 1997*, pp. 707-717.

Cross, N., 1982, Designerly ways of knowing, *Design Studies* **3**(4), 221-227.

IMEC = Inter-University Micro-Electronics Center Leuven (Belgium)

Lernout & Hauspie Speech Products - Ieper (Belgium)

Neuckermans, H., 1992, A Conceptual model for CAAD, *Automation in Construction*, **1**(1), 1-6

Schon, D., 1985, *The Design Studio*, RIBA pub Ltd., London