A DATA-DRIVEN SPEECH SYSTEM

D Hawthorn, Department of Business Systems, Monash University
Dr C White, Department of Business Systems, Monash University

ABSTRACT: Communication is a process of meaningful message exchange that is fundamental to daily living. Any impairment of this process has a significant impact and greatly reduces the quality of life that many take for granted. Opening communication with people who have severe and multiple disabilities is a challenge. Furthermore, developing a system that can be readily customised and reused in a number of application areas poses a number of major challenges. This paper describes a system, currently being developed by the authors, that addresses the areas of communication, customisation and reuse. The system is a Dynamic Data-Driven (3D) system that runs on an IBM compatible PC incorporating speech output through a single point interface to SBTALKER using a Sound Blaster card (1).

INTRODUCTION

Communication is a process of meaningful message exchange that is fundamental to daily living. Any impairment of this process has a significant impact and severely reduces the quality of life that many take for granted. Opening communication with people who have severe and multiple disabilities is a challenge.

The communication hurdle for those with severe and multiple disabilities can seem insurmountable. History shows that this hurdle can be overcome and that it is worth overcoming. A well known example would be that of Helen Keller who was blind, deaf, and mute from infancy (Keller, 1967). She not only overcame her disabilities with regard to communication but was also a prominent writer.

There are many different ways of assisting non-speaking people and their families. These vary from low technology systems such as word boards and icon systems, e.g. Compic, through to very high technology systems such as specialised keyboards and specialised speech output devices, for example, the European Concept keyboard, the Canadian WiViK system (Windows Visual Interface Keyboard) or the Liberator (2). At the core of such systems is the objective of effective communication.

In this regard, a number of points need to be made. Firstly, the solution to many communication problems can be achieved if there is a sufficient amount of money available. In cases where the disabilities are multiple and severe typically resource demands are high for such things as wheelchairs and transport systems. Where these disabilities arise from congenital conditions, resources may be limited by family income and the additional expense needed to solve any communication problems may impose an excessive financial burden.

In addition, the use of available aids is often limited to those who have adequate fine or gross motor skills that allow the use of data entry keyboard or other pointing devices such as a mouse. Furthermore, the use of icon systems is limited in scope by the nature of the pictures displayed and often masks the realisation that the bulk of communication between people involves speech.

Finally, the development of an effective communication system for people with multiple and severe disabilities will certainly require the tailoring of a system to specifically match the capabilities of the person involved and the cost of producing one off solutions for such purposes is clearly prohibitive.

This paper describes a system, currently being developed by the authors, that addresses the areas of communication, customisation and reuse. The system is a Dynamic Data-Driven (3D) system (that runs on an IBM compatible PC incorporating speech output through a single point interface to SBTALKER using a Sound Blaster card.

THE PROBLEM

The development of the system has arisen from two sources, the need to address the communication problems of an eleven year old boy and the problem of developing large scale business systems software. While the two areas may appear to be radically different they share the need to provide an effective communication process with a user.

In the case of the eleven year old, he suffers from cerebral palsy as a result of exposure to cytomegalovirus (CMV). He is a microcephalic with spastic quadraplaegia and has never spoken. Communication is limited to a verbal 'yes' noise and a visual 'no'. In addition he has limited motor abilities and does not possess the fine motor control needed to use a keyboard, mouse or head pointer. It is this multiple handicap in both the speech and motor areas which makes it so difficult to communicate.

In considering solutions to the problem of communication, the available computer technology and its likely usefulness was considered. In general options available appeared to be costly and have a limited likelihood of meeting the specific needs of the person involved. Interestingly, workers in the UK reported mixed success with computer based systems and indicated that some success had been achieved in the communication area using physical menu systems based on business card holders. Initial tests based on such a system in which menu options were written on separate cards and inserted in the card holder, with a new page for each topic and an assistant to provide the scanning function, proved to be somewhat successful. The major problems with the system lay in the preparation of the cards, the bulkiness of the system, the need for an assistant to scan and the uniqueness of the solution.

As already noted, the authors also have research interests in developing large scale business systems software and that if the solution to communication problems is not to prove both unique and costly then there is a need to gain a high reuse rate of software across application areas. In this regard the structure of many business systems is similar to the menu driven system required to provide a communication system. The primary entry point into a business system is a menu navigation system in which the user moves to the subsystem of interest by making menu selections. Structurally, the two problem areas appear similar and it was decided to exploit this aspect.

Finally, in further tests with the person in the disability area a very strong reaction was forthcoming with software incorporating speech output. Perhaps somewhat fortunately, work commenced in this area as very low cost sound cards became available for the personal computer.

The aim of the work has thus been to pull together all the threads of communication, customisation and reuse with a view to providing a cheap and flexible communication aid for those with multiple severe handicaps. Given the problems associated with customisation our approach has been to use a dynamic data-driven system which also has application in the business systems area. The ability to reuse the program (i.e. the same executable code) in both manufacturing and communication aids is an indication of the flexibility of the approach.

THE 3D SYSTEM

As already indicated the aim was to produce a low cost configurable solution. The 3D meets this criterion, requiring only an inexpensive IBM compatible PC with 640KB of memory and a Sound Blaster card for the speech.

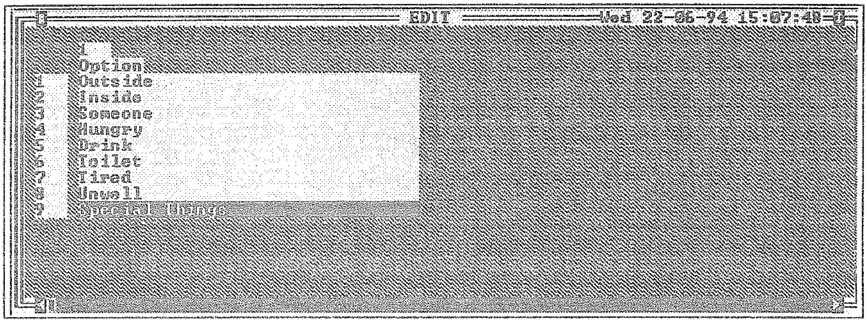
The system is fully configurable. Each item in every menu can be modified to suit individual users. There are effectively no restrictions on the number of items in a menu or on the number of menus. The total amount of data that can be accessed is only limited by the size of the disk. Currently 3D is limited to a menu system of using words or phrases, but it is intended to add a virtual keyboard capability when time permits.

Communication is achieved through software controlled automatic scanning, with 3D reading out the menu items. This exploits the positive response to speech mentioned earlier and is also required because it is unknown as to whether the person can recognise word symbols and read. Movement through the system is made by selection of an entry from a menu.

Clearly an important consideration is the method of selection and the input device to be used and as already mentioned there are a variety of input devices available with many electronic communication aids. The 3D is no exception and can accept input from the keyboard or mouse. In addition, to solve the problem of severe disabilities it has also been designed to accept input from any simple switch connected to the joystick port of a PC. In the case of the person mentioned a number of simple switches are used ranging from a modified Singer sewing machine foot pedal, providing a large surface area that can be hit by users with limited motor control, to an industrial limit switch that can be activated with a small elbow movement. As a single switch is being used for input, 3D scans the items displayed on the screen, and then allows the user to choose the current item by hitting the switch.

Figures 1 and 2 show sample screens from the prototype application. This is based on a business card holder system which the eleven year old currently uses.

Figure 1. Opening menu of prototype application.



THE METHOD

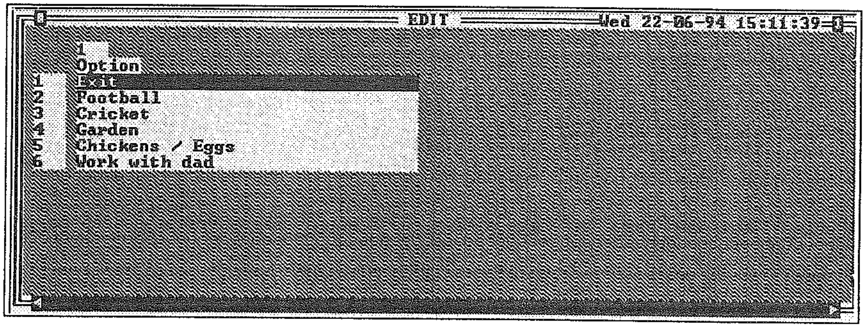
3D has been written in ANSI C and as previously mentioned a data-driven approach was used. The concept of data-driven programming is not new but little research has been done in the area. The term 'data-driven' has a variety of meanings. The meaning used here is that the functionality of a program is determined, not by code, but by data files. The purpose of the code is to interpret and use the data. O'Keefe & Haddock (1991) use the term in a similar manner in simulation programming to describe the relationship between components in a model using data rather than code. This is quite different to the use of the term in concurrent programming where it is used in the sense of processes being able to proceed when the necessary data becomes available (Lui & Cheung, 1990). Another use of the term is illustrated in Mazlack (1990) where it is used in the context of knowledge based systems to refer to the expert systems approach of "supplying each possible external knowledge source before coming to a decision".

3D is essentially a menu / data-entry screen system. The menu items, the action taken when an item is selected, and the structure of the data entry screens is determined by the data. The menus can also be changed into data-entry screens at the push of a button. This allows the development and editing of menus from within 3D. One initial menu must be designed and placed in data files before the program is run but after that point all menu development can take place within 3D. One way in which this development process is enhanced is by allowing the editing of a menu item and then saving it as a new item. As the items within a menu have a significant amount of common data this speeds up the process.

The ability to swap easily between menus and data entry is due to a rather blurred distinction between the two. Each field has a field type flag which specifies if it is a 'trigger' or a data item. This is the primary distinction between menu items and data items. A secondary distinction is a global flag that determines whether or not the edit mode is on. In order for a field to behave as a menu item the global edit flag must be off and the field type flag must specify that it is a trigger. Temporarily changing a menu item to a data item is achieved by switching the global edit flag to on. The field type flag can also be changed and this results in a more permanent swap between menu items and data-entry fields.

One important aspect of 3D is its modular nature. 3D is (as much as is possible) broken up into distinct units. Two of these are the sound and file system. The sound system used in 3D is Sound Blaster and it is implemented through a single-point interface. This allows for Sound Blaster to be easily replaced. The filing system currently used is Novell Btrieve, which opens the possibility of networked versions of the software, but its use is not strongly tied to the code and it can be readily replaced by a different file system with little difficulty.

Figure 2. Submenu for the 'Outside' option in the prototype application.



Advantages.

The principal advantage of the data-driven approach is fast development time. It should be noted that the fast development times only occur after the executable program has been written. After this point specific applications can be created quickly.

In this particular case it would have been much faster to use a traditional approach. Furthermore, the interface could be greatly improved without the restrictions imposed by the wide, but limited, flexibility. The reason for persisting with the data-driven approach in this case is due to the second advantage.

When using the data-driven paradigm to the extent that we have another advantage surfaces. The executable program can be used in widely different applications. 3D is also planned for use, in fact its primary use, in manufacturing systems. This is not possible with traditional programming and to our knowledge has not been explored to a great extent in the data-driven field.

One other advantage of this method is that it allows for incremental development of the application without repetitive compilation of the source code. This also allows for an application that changes with time.

The issue of incremental development raises another matter. If the application is not going to be finalised before the user starts to use it, can the end user do the further developments themselves? At the present time this is not possible as keyboard skills are necessary. The future addition of a virtual keyboard would make this possible. The largest problem would be in teaching how 3D works. 3D is somewhat complex and teaching it to anyone takes time. Difficulty in getting feedback from the learner would increase the time required.

Disadvantages.

The amount of time required to input data is one criticism of data-driven programs. This is a valid criticism. In this case, where we have a fairly simple menu system, the time required is approximately a linear relationship with size.

O'Keefe & Haddock state that incorrect data is a significant problem. In this situation it is not as significant. 3D is not a complex statistical package that can hide wrong results. Wrong data will clearly show up in the testing when incorrect menus are displayed on the screen.

Data-driven systems are very flexible but still have their limits. They are certainly limited in their flexibility when compared to traditional programming. The look and feel of 3D would be one example. The screen interface is nothing fancy. The priority of 3D was to get a working system rather than a system that looked good.

Differences.

The software presented here is different to other data driven paradigms in the extent to which the data/code balance is taken. The term 'dynamic' has been used in conjunction with 'data-driven' to distinguish this from other data-driven approaches. There are some advantages in a move in this direction. The greater the shift towards data the more flexible the system becomes (and admittedly the harder the data is to write). This flexibility allows the same code to be used in any application that mainly consists of menus and data-entry screens.

The way in which it is being used is also different. The closest use of the term 'data-driven' that we could find that is similar to ours was in the simulation of manufacturing systems (O'Keefe & Haddock). 3D is not being used to simulate manufacturing systems but in implementing them.

CONCLUSION

The ability to communicate with others greatly effects the quality of life that a person can enjoy. This important aspect to our quality of life can only be enjoyed with significant difficulty by those with severe and multiple handicaps. Aids designed to assist those with such difficulties must be configurable to the individual user. A data-driven, fully configurable, and affordable aid for non-speaking people is being developed by the authors that addresses this need.

The high reuse of code is a challenge in software development. The system presented here meets that need and provides a framework from which widely differing applications can be developed with little need for writing source code.

The sound system has been implemented through a single point interface which allows the associated hardware and software to be replaced with minimal effort.

NOTES

(1) "SBTALKER" and "Sound Blaster" are trademarks of Creative Labs Inc.

(2) "WiViK" is a trademark of The Hugh MacMillan Rehabilitation Centre, Toronto
"Liberator" is a trademark of Prentke Romich Company, Ohio

REFERENCES

Keller, H. (1967) Hellen Keller: Her Socialist Years, p. 7 (ed.) Foner, P.S. (International Publishers Co., Inc: New York).

Lui, S.P.H. & Cheung, Y.S. (1990) *Data Driven C - A Concurrent Programming Language.* IEEE Region 10 Conference on Computer and Communication Systems - IEEE TENCON '90, 165-169

O'Keefe, R.M. & Haddock, J. (1991*)* *Data-driven generic simulators for flexible manufacturing systems.* International Journal of Production Research. 29: 9, 1795-1810

Matzlack, L.J. (1990) *Saticficing in knowledge-based systems.* Data & Knowledge Engineering. 5: 2, 139-166