Mainstream Solutions for the Problems of Disability

George Markowsky, Ph.D

Computer Science Department
University of Maine
Orono, ME 04469

Abstract: Mainstream Solutions for the Problems of Disability

It is the author's contention that solutions for problems of disability should be effective solutions to problems faced by the able. Such solutions are much more likely to be widely available at a reasonable cost, and allow the disabled to be more fully integrated into the work place because they can use the same equipment and procedures as everyone else. Such solutions can be discovered either by adapting existing mainstream products for use by the disabled, or by creating products for the disabled that have wide appeal to the able. Examples discussed from the first category are the programming language APL, and PC-Pedal™, a keyboard enhancing system developed by the author and Calvin Brown.

Examples discussed from the second category are wheelchair ramps and Super Terminal, which is a keyboard enhancing, voice-controlled add-on for the IBM 3270 terminal.

Finally, designers not expressly addressing the needs of the disabled are also urged to adapt their designs for disabilities of users. Often, very minor design changes can make a tremendous difference in the accessibility of a device to the disabled.

The Importance of Strategic Thinking

Sometimes, the obvious is hard to see. However, once seen, it is easy to appreciate. In this paper I want to present some observations which now seem obvious to me, but which took me a long time to see. Perhaps, I can convince you that they are important ideas for creating useful solutions to problems of disability.

It seems to me that any good solution to the problems of disability must have the properties listed in Figure 1 on page 22. It might appear that solutions to problems of disability are not likely to possess such properties. In fact, the properties listed there even seem contradictory. After all, how can a product for the disabled be affordable and yet be distributed and supported widely? Since the disabled population is sparse, widespread support and distribution would require a lot of money and necessitate a high cost.

The argument just given reflects the stage of my thinking several years ago. Eventually, I realized that it is based on the assumption that solutions for problems of disability can only exist on a small scale and are created in a vacuum independently of all other developments. In particular, it is assumed that solutions of common, everyday problems have little relevance to the problems of disability.

It was only after I had built and tested several devices for aiding the disabled that I was able to see the importance of thinking about mainstream problems when working on problems of disability. This paper will describe some of the devices I

1. It should be effective.

   It should solve a problem of interest to disabled people in an effective manner.

2. It should be available.

   Any disabled person needing it, should be able to find it easily.

3. It should be affordable.

   If a person can't afford it, it won't do that person any good.

4. It should be easy to learn.

   There should be ample materials available for learning how to use it, and it should be easy to find people who can answer questions about it.

5. It should be easy to support.

   Repairs and routine maintenance should be easy and inexpensive to obtain.

6. It should interface with other equipment.

   It should allow the disabled to participate in projects with other workers.

7. It should hide differences.

   Disabled people using it should not appear as radically different or weird to the other workers.

Figure 1. Desirable Properties. Any device or system designed for the disabled should have these properties.

have been involved in and some of the insights I have developed as a result. Since it took me quite a while to appreciate these points, I hope that this paper can speed the development of your own thinking about this topic.

The most important idea I would like to convey to you is to design devices and systems that can be used by the vast majority of people. Do not specifically target the disabled! Having made such an extreme statement, I will now have to qualify it somewhat. While there are undoubtedly disabilities so severe that only the most extraordinary solutions can be effective, many disabled people can be aided significantly by less extreme measures. If your device is perceived as a solution to a significant problem by many able users, then tremendous benefits will accrue to disabled users as well and your device will be more likely to possess the properties listed in Figure 1.

One tremendous advantage of tying your work to mainstream problems is to reduce differences among people. If the same device can be used by both the disabled and able, then as far as the particular task is concerned the disabled are no longer disabled. They can now enjoy all of the benefits, listed in Figure 2 on page 23, that come from using standard equipment. Not coincidentally, these benefits result from having the properties listed in Figure 1.

1. It is effective.

   Presumably, the standard equipment was chosen because it is effective for the task.

2. It is available.

   No special distribution channels are needed to find and purchase it.

3. It is affordable.

   No additional costs to purchase it since it is the standard equipment.

4. It is easy to learn.

   No special instruction or tutorials are needed.

5. It is easy to support.

   The standard support system is sufficient.

6. It interfaces with other equipment.

   It avoids isolation caused by different procedures or incompatible standards.

7. It hides differences.

   The disabled worker does not stand out because of unusual equipment.

Figure 2. Benefits of Using Standard Equipment

Two Useful Views of Disability

The following two views of disability make it plausible that designing with mainstream solutions in mind can succeed in providing successful solutions for problems of disability. These views of disability also make it easier to design devices with a broad appeal.

Disability is a Spectrum

People are not in two categories: able and disabled. Rather, ability and disability are simply two ends of a continuum as shown in Figure 3 on page 24. Even serious disabilities such as paralysis, blindness and deafness differ in degree and kind, and may be more or less limiting depending on circumstances. This variability, which was one of the first things I noticed when working with the disabled, is an important factor to consider.

Ability ________________________________ Disability

Figure 3. The Continuum Formed by Ability and Disability

It is important to notice that disability is sometimes a temporary state. Temporary disabilities can range from the relatively minor ones caused by colds and other nuisance diseases, through more serious disabilities such as broken limbs, to very severe disabilities such as strokes or serious injuries in automobile accidents. Thus, many able people are disabled temporarily during their life.

Aging often reduces a person's performance and increases the likelihood of temporary or permanent disability. Since the baby boom generation is now in middle age, the problems of disability will become even more important in the future as this large group of people ages.

Devices that can adjust to a wide range of user abilities are very helpful to people recovering from "temporary" disabilities. Many people with broken arms have discovered how unfriendly the equipment they previously used becomes when they no longer have two hands to work with.

Furthermore, disability in one area does not imply disability in another area. Similarly, ability in one area does not imply ability in another area. Thus, many apparently able people do not perform certain tasks well. For example, many able people are very poor typists since they use only one finger on each hand. This reduces their output tremendously, and may be considered a disability of sorts.

Deliberately trying to design mainstream solutions for problems of disability translates into designing for as large a market as possible. This is just good business. As we all know, good business has, or should have, very strong appeal. In fact, a strong business case will usually have more appeal to most companies than a strong ethical case.

Disability as an I/O Restriction

In many cases, the disabled can perform the tasks they are required to do, but can only do so very slowly compared to the average able person. For example, most quadraplegics type slower and have more difficulty handling computer printouts than the average able person, although many of them can perform these tasks at some level.

You can of course conclude that all difficulties are the result of these disabilities and nothing more. However, if you examine the situation from the perspective I have been recommending, you will immediately realize that this is not the correct view, since able people have the same problems.

It does not require elaborate surveys to conclude that many able people are poor typists. In some cases their lack of skills is severe enough to be classed as a disability in using computer equipment. Not only is typing a problem for many able people, but printouts, especially bulky ones, are a problem for all programmers.

Notice that both the able and the disabled have the same problems, and that the differences are of degree rather than kind.

Sometimes the disabled are stymied by relatively minor barriers: they may handle a device well except for one feature. A well-known example of this is the the three key reset on the IBM PC, which is very difficult for quadraplegics and one-armed people. This problem is similar to the one wheelchair users have in navigating streets that have no ramps on curbs. Such barriers are not deliberately erected, but result when disability is never considered as a factor in designing the device or system.

There are several ways to tackle I/O problems. First, reduce the need for I/O. Second, speed up I/O. Finally, do both.

Finding Mainstream Solutions

I hope that you are now completely convinced of the worth of looking for mainstream solutions to problems of disability. However, you may be puzzled about how to find such solutions. The next section describes several such solutions, which I hope will help your own search for acceptable solutions.

There are two main strategies for looking for such solutions: the accidental approach and the deliberate approach. The accidental approach consists of accidentally discovering that a device for the disabled is also valuable for the abled, or discovering that a device for the abled can easily be adapted for use by the disabled.

The deliberate approach is to force yourself to consider mainstream solutions in the early stages of your work: at all times in your work you ask yourself how your work can be made useful to as many people as possible. Keep in mind that disability is not an isolated state, but that ability and disability form a continuum. Try to make your work cover as much of this continuum as possible.

Unfortunately, I must confess that all of my true-life examples are examples of the first strategy: the accidental approach. It is only after stumbling over the same observations several times that I have decided to make them part of my conscious design philosophy in the future, regardless of whether I am explicitly working on devices for the disabled or other things. If all designers subscribe to the same design philosophy, we would not be stuck with such things as the three-key reset of the IBM PC.

The accidental approach can take two forms:

1. Discovering that something for the abled can be used to great advantage by the disabled.

2. Discovering that something for the disabled can be used to great advantage by the abled.

As examples of the first form, I will discuss APL and PC-Pedal™. As examples of the second form, I will discuss Wheelchair Ramps and Super Terminal.

It is important not to work in a vacuum if you want to discover solutions that have the desirable properties listed in Figure 1 on page 22. Too many efforts fail to take into account the total environment in which the proposed solution will have to operate. This results in one-of-a-kind solutions that have little chance of possessing the desirable properties just mentioned. To avoid working in a vacuum, investigate what others are doing in your area of interest and attempt to make your work relevant to the mainstream.

Examples

APL

One of APL's most attractive features for use by the disabled is its reduced need for I/O. In case you are not familiar with it, APL is a very compact language developed originally by IBM. It has very small input and output requirements.

Because APL is so compact, it reduces the need for printouts since many programs easily fit on a single screen. If printouts are necessary, they are often much smaller than they are for other languages. Furthermore, APL programs are often quickly written and debugged. Figure 4 shows a short BASIC program and a comparable APL program. Note how much shorter the APL program is.

BASIC

```
10 SUM = 0
20 INPUT X
30 SUM = SUM + X
40 IF X <> 0 THEN 20
50 PRINT SUM
```

APL

```
+/A <- []
```

Figure 4. Equivalent Programs in BASIC and APL. Both programs sum a sequence of numbers.

APL was the primary language in a voice-controlled programming system I built while working at the IBM Thomas J. Watson Research Center in Yorktown Heights. This system was tested briefly during a study at the NYU Institute of Rehabilitation Medicine. The system, see Figure 5 on page 27, was based on an IBM 5100. At that time, the 5100 was the only portable computer that ran APL.

```
 ______________     _____     ______
|              |   |     |   |      |
| Voice        |___| Aim |___| IBM  |
| Recognizer   |   | 65  |   | 5100 |
|______________|   |_____|   |______|
```

Figure 5. System Design of an APL Voice-Controlled Programming System

The primary aim of the project was to test the effectiveness of voice-control, and a secondary aim was to test the effectiveness of APL. The test did not resolve either issue conclusively, although it furnished many valuable ideas for future work. The test also uncovered some problem areas. The only prototype constructed cost about $30,000, so it was too expensive to replicate. Because of this it was only available for very brief periods to the subjects, none of whom had any serious interest in learning to program.

The results of the test were interesting enough so I wanted to try again with a new system based on an IBM PC that would cost about $5,000 (including the cost of the PC and the speech recognizer), and could be replicated much more easily. My experience with the 5100-based system showed that it is very important for any system to be affordable and easily replicated. Otherwise, too few prototypes will be built to learn much useful.

The results of the test (described in Markowsky[1981]) argued in favor of experimenting further with APL and voice-control for use by the disabled. Unfortunately, as far as APL was concerned, all of the other environments that I worked in either required another programming language, or were not involved with programming. Thus, I have not had any further chance to investigate the uses of APL in this context. However, I think that this is a promising area of research and should be pursued further.

Unfortunately, even though IBMers originated APL, IBM has not supported APL with as much gusto as it should. In particular, it took IBM quite a while to bring out the first version of APL for the IBM PC. After another long period, Version II was released. Unfortunately, Version II is sold only in Europe and cannot be purchased in the US.

To conclude, I would like to list some of the pros and cons for using APL in your own work.

Advantages of APL

- It is very compact.
- It is interpreted, which often helps with debugging.
- It is easy to write working code quickly.
- APL has a strong and dedicated (fanatical) following.

So as not to leave you with an incomplete picture, I would like to briefly indicate some of the disadvantages of using APL.

Disadvantages of APL

- Critics claim it is cryptic.
- It has an unusual character set which makes it incompatible with many other languages and terminals.
- It allows the programmer to program in a horrible style if desired.
- It is not available on many terminals and computers.
- It has a strong and dedicated (fanatical) group of detractors.

PC-Pedal™

PC-Pedal™ is a combination of a foot pedal and software which replaces any key on the IBM PC keyboard. PC-Pedal™ has been adapted to work with switches other than foot pedals and works with many PC Clones and related computers. Figure 6 on page 29 is a photograph of the PC-Pedal™. PC-Pedal™ connects to the parallel printer port of the PC or compatible so as not to interfere with the normal operation of the printer. Anytime the pedal is depressed it mimics the action of the appropriate key. In some cases PC-Pedal™ creates effects, such as erasing words backwards in WordStar™, that are not obtainable directly from the standard keyboard.

PC-Pedal™ was developed by me and Calvin Brown and is marketed by Brown & Co. (95 School St., S. Hamilton, MA 01982, 617-468-7236). It has been reviewed favorably in publications like *BYTE, PC Week, the New York Times*, and several specialized magazines, including some devoted to aids for the disabled. See the bibliography for specific references.

The original motivation for PC-Pedal™ was to speed typing by allowing the typist to keep his or her hands on the home row or numeric keypad, and control the Ctrl, Alt and Shift keys with a foot pedal. This device can significantly speed input in WordStar™ and Lotus 1-2-3™, and is primarily marketed for the average business user of the PC.

Working PC-Pedal™ made us realize that it had great potential for aiding various classes of disabled people, including people with just one arm. We had a very personal example of this when my partner Calvin Brown broke his arm in a fall and spent several weeks as a one-armed user of the IBM PC.

The fact that PC-Pedal™ has appeal to a wide range of people illustrates the main principle I have been stressing. In an office where it is a standard, a range of disabilities can be accommodated with no additional effort on the part of any other person.

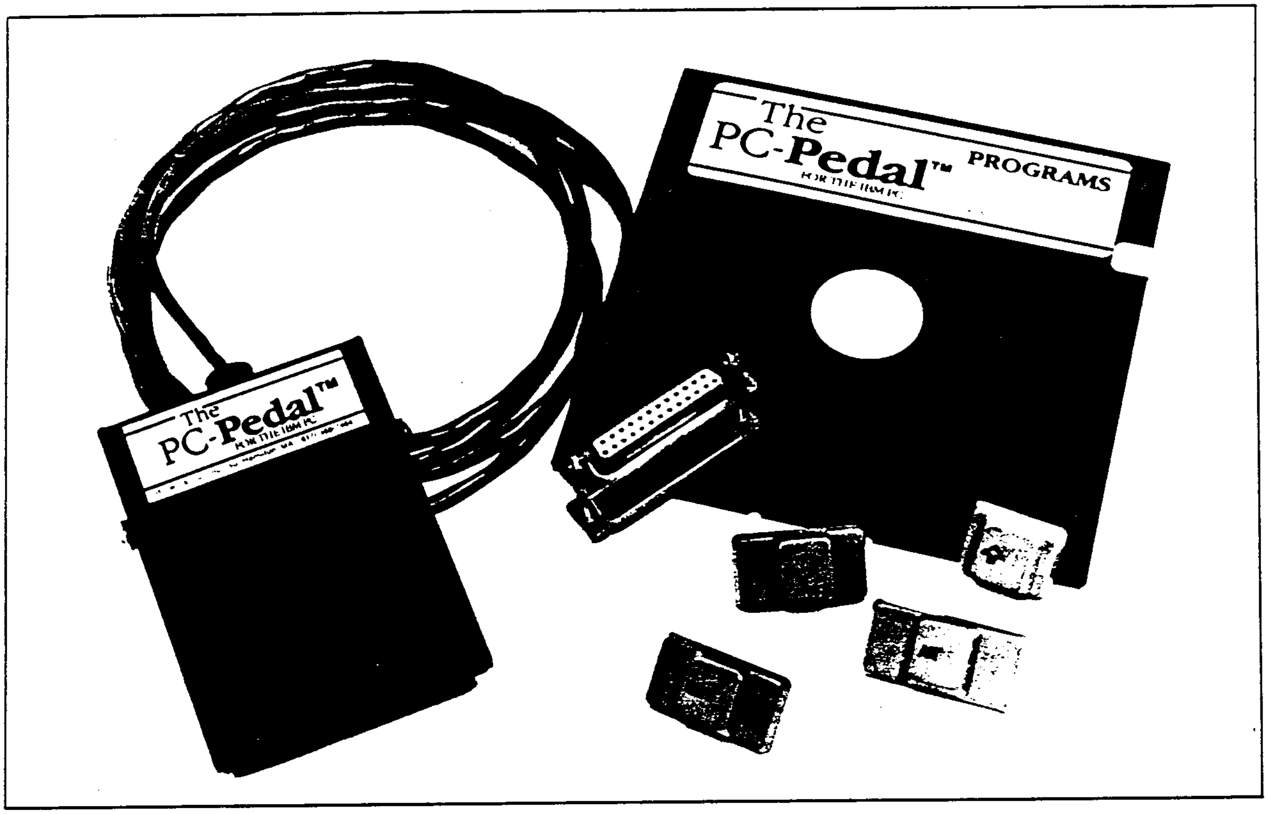

Figure 6. PC-Pedal™. This allows keys to be substituted for by a foot pedal or other switch.

Wheelchair Ramps

The wheelchair ramps that are found commonly on street corners and at entrances to buildings provide a very common example of a device installed for aiding the disabled that has turned out to be incredibly useful to the abled. Any parent who has pushed a stroller along a city street knows how helpful it is to have ramps cut into the curb. Bicycle riders and people moving heavy things have also found these curb ramps to be of great benefit. Taking note of deficiencies in our systems and a concern for rectifying them will go far toward ensuring that other, comparable improvements are made that will benefit everyone.

Super Terminal

This was a research project that grew out of my work with the IBM 5100 voice-controlled programming system. It was based on an IBM PC and was essentially a voice-controlled and keyboard enhanced version of the IBM 3270 terminal. Figure 7 on page 30 shows the system design of Super Terminal.

```
 ________      __________     ______________
|        |    |          |   |              |
| 3270   |____| IBM PC   |___| Voice        |
|        |    |          |   | Recognizer   |
|________|    |__________|   |______________|
                   |
 ____________      |
|            |     |
| PC         |_____|
| Keyboard   |
|____________|
```

Figure 7. System Design of Super Terminal.

A key difference between Super Terminal and the 5100 System is that the keyboard can duplicate all of the fancy features that are available for voice-control. Thus, it made available keyboard macros and a high level intelligent keyboard to users of the 3270 just as these ideas were beginning to appear in the personal computer world. Unfortunately, looking at IBM's recent terminal offerings shows that IBM has not yet been impressed with the need for keyboard assistance for all users.

Super Terminal's history is very interesting and illustrates the close relationship between the problems of designing for the disabled and designing for the standard market. After Super Terminal was completed, several units were installed throughout the company wherever it was desired. Permission was never given to install it outside the company, so all experience with it is IBM internal.

After Super Terminal was used in many situations, it became apparent that the souped-up keyboard was better than the voice-control. Furthermore, the keyboard-only version of Super Terminal costs about $2000 less, and only requires a PC and a 3270, both of which are standard pieces of equipment.

Sometime around 1982, I installed a keyboard-only unit at IBM Endicott to help Tom (not his real name), who became a quadraplegic because of an auto accident. An IBMer at Endicott, George Safranek, was assigned to help Tom learn to use Super Terminal I. Safranek did a splendid job and observed that Tom had gone from being the least productive to the most productive worker in the group. Furthermore, Tom was significantly more productive than any of the able people in the group. Not only was his rate of data entry much higher, but he made significantly fewer errors than any other worker.

Realizing this, Safranek pushed very hard for Super Terminal to be adopted by his own group and several other groups. He prepared training materials and organized several meetings so that other IBM groups could learn about Super Terminal and use it. Super Terminal has been and continues to be used for a large number of administrative tasks such as product documentation, purchasing and other administrative functions.

I estimate that at one time Super Terminal was used by more than 100 people located in at least six different sites within IBM. Since I am no longer with IBM, I have not been able to track the full extent of its use, although it was considerable.

This popularity is very encouraging since there was never any organized IBM effort to push it or to exploit its successes.

Unfortunately, Tom, who was so successful with Super Terminal, became a victim of government insanity. The government decided that he was earning "too much money," so he was given the "choice" of either quitting his job or losing his benefits. After he lost his fight to keep both his job and his benefits, he left IBM. I was told that leaving his job at which he had become very successful was very difficult for him psychologically.

As I have already stated, IBM never expressed any official interest in the fact that productivity had been so greatly improved, and did not move to exploit this development. It is interesting to note that the only person to receive an award in connection with this episode was Tom's manager.

Being invited to address this group today gave me the excuse to check up on how Super Terminal is doing nearly three years after I last had anything to do with it. I was very pleased to learn it is still used on a daily basis by the original group of people in Endicott, and has not been displaced by any newer systems. Interestingly enough, Super Terminal has not had any improvements or "maintenance" performed on it for more than three years.

Super Terminal is designed around a 3270, and will not work with newer terminals, because it needs to be updated. Nevertheless, the current users find that it increases their productivity significantly over what they can achieve on the newer terminals. Unfortunately, the current users are under pressure from their administrators to abandon their "obsolete" 3270s, but so far they are holding their own.

IBM's Role in Helping the Disabled

Because it dominates the computer field, IBM's attitudes about improving the user interface are very important. I sincerely hope that the company takes a much greater interest in improving the user interface, because it will benefit both the abled and the disabled. I hope that some of the indifference that the company has shown to this area in the past can be discarded, and a more positive attitude can be adopted. I am sure that this will result in improved business for IBM and a wider appeal of its products.

IBM must pay more attention to improving the user interface. New developments in the world of computing will force it to adopt a new philosophy, which I hope will be similar to the one I have been outlining.

I would like to mention a personal computer that has taken a very bold step in the proper direction: the Commodore Amiga. Every Amiga comes with a high-quality speech synthesizer built-in. Thus, of all the personal computers on the market, the Amiga is the easiest to adapt for the blind or those who cannot speak. I can only hope that the competing manufacturers will follow in Commodore's footsteps.

This paper is primarily addressed to people who work on devices for the disabled. Nevertheless, I would like to conclude by inviting people working in all fields to consider the points raised in this paper. It is extremely important that accessibility be considered in every device and system that is built.

Bibliography

Items two through six below are some of the reviews of PC-Pedal™.

1. Markowsky, Youdin and Reich *Vocational Rehabilitation of the Severely Disabled: Voice Controlled Computer Programming.* Proc. 4th Ann. Conf. on Rehab. Engineering, Washington, DC, August 30-September 3, 1981, pp. 135-6.

2. James Langdell, *Kick the Key Habit with a PC-Pedal.* PC Magazine, October 16, 1984, p. 64.

3. Editorial Staff, *Productivity.* Personal Computing, October 1985, p. 48.

4. Erik Sanberg-Diment, *Mouse Skates, Paper Planes and Other Yule Gifts.* New York Times, December 17, 1985, p. C-3.

5. Chris H. Pappas, *Non-Keyboard Input Devices.* BYTE, May 1986, pp. 285-288.

6. Editorial Staff, *New Foot Pedal to Run Computer.* Computer-Disability News, Summer 1986, Vol. 3, No. 2, p.5.

Author Profile

George Markowsky received his B.A. in Mathematics from Columbia University, M.A. and Ph.D in Mathematics from Harvard University and was a Postdoctoral Fellow in Mathematics at Harvard University. He joined IBM Research in 1974 and left in 1984 to become Chairman of the Computer Science Department at the University of Maine.

His research interests include algorithms, combinatorics, theoretical computer science, and devices for the handicapped. While at IBM, he worked in a variety of areas including hashing (Research Division Award 1979), reconstruction of geometrical objects (Outstanding Contribution Award 1981), and he created a voice-controlled 3270 terminal. He continues his work in many of these same areas at the University of Maine.

George Markowsky, University of Maine