INFORMATION VISUALIZATION TECHNIQUES FOR MOTION IMPAIRED PEOPLE

Emilio Di Giacomo[1], Giuseppe Liotta[1] and Stefano Federici[2]
[1]Department of Electrical and Computer Engineering, University of Perugia, Via G. Duranti 93, Perugia, Italy
[2]Department of Human and Education Science, University of Perugia, Piazza Ermini 1, Perugia, Italy

Keywords: Human-Machine Interfaces for Disabled Persons.

Abstract: Several alternative input techniques have been proposed in order to make computers more accessible to motion impaired people. They include brain computer interaction (BCI), eye movement detection, and speech/sound interaction techniques. Even if these alternative techniques partially compensate the reduced capabilities of the end-users, the overall interaction can become slow and convoluted since the number of commands required to complete a single task can increase significantly, even for the most common computer applications. In this position paper we describe a novel Human-Computer Interaction paradigm for motion impaired people based on sophisticated diagrammatic interfaces. The main idea is to use Information Visualization approaches to overcome the limited interaction capabilities of the alternative input devices typically used by individuals with motor disabilities. Our idea is that the limited information bandwidth of the input devices can be compensated by the broad bandwidth of the adopted diagrammatic interfaces, capable of conveying large amounts of information at once.

1 INTRODUCTION

Computers have become an essential and increasingly pervasive tool in everyday's life, in both the professional and the personal entertainment spheres. Unfortunately, people with motion impairments, especially those who do not have complete functionalities of their upper limbs, face serious difficulties in using traditional input devices such as keyboards and mice because such devices require sequences of small and precise movements to accomplish even the simplest tasks: Think for example of the continuous movement of a mouse when browsing information in a computer.

In order to overcome the barriers for the motion impaired to use computers, several alternative techniques have been studied including, e.g., brain computer interaction (BCI) (Chin, Barreto & Alonso 2006), eye movement detection (Fejtová, Fejt & Štepánková 2006), tongue control (Struijk 2006), and speech/sound interaction (Manaris, McGivers & Lagoudakis 2002), (Sporka, Kurniawan & and Slavík 2006). These techniques make it possible for the end-user to interact with the computer by performing a sequence of commands, where each command corresponds to selecting one from a very limited number of options. Conceptually, this is equivalent to having a keyboard with very few keys, possibly a single button. In other words, these input devices can be modelled as systems with a reduced number of statuses (only two for the single button case).

Even if these alternative devices partially compensate the reduced capabilities of the end-users for a single command, the overall interaction can become slow and convoluted because the number of commands required to complete a single task increases with the loss of expressiveness of the input device. For example, searching a document in the Web with an interface operated by BCI technologies requires translating each keyboard and mouse command into a sequence of binary choices imposed by the two control statuses of a typical BCI.

In this position paper we propose a new interaction paradigm between computers and motion impaired people. The leading idea is that the use of sophisticated Information Visualization technologies can significantly reduce the number of commands needed to complete a task, thus overcoming the aforementioned discomfort in the interaction between computers and motion impaired people. Information Visualization conveys abstract

Di Giacomo E., Liotta G. and Federici S. (2010).
INFORMATION VISUALIZATION TECHNIQUES FOR MOTION IMPAIRED PEOPLE.
In *Proceedings of the Third International Conference on Health Informatics*, pages 361-366
DOI: 10.5220/0002758403610366

information in intuitive ways. Visual representations and interaction techniques take advantage of the human eye's broad bandwidth pathway into the mind to allow users to see, explore, and understand large amounts of information at once.

In the rest of the paper we shall justify our vision by recalling existing approaches and their pitfalls (Section 2), proposing a new approach (Section 3), and describing a reference software architecture that can be used to implement our approach (Section 4).

2 EXISTING APPROACHES

Different approaches have been proposed in the literature to overcome the digital divide for motion impaired people. We classify existing approaches with respect to the hierarchical model of Figure 1. This figure describes the interaction between a user and a computer as a traversal of a stack of four different layers: The Task Layer, the Operation Layer, the Command Layer, and the Action Layer.

When a user interacts with a computer, his/her goal is to perform some tasks that are specified in the Task Layer. For example the user may want to search a file in the file system, or send an e-mail, or search a page in the Web. The choice of a task triggers activities at the lower levels of the hierarchy.

At the Operation Layer, the user has to execute a number of operations. For example, if the task selected by the user is that of searching the Web, the user's operation consist of writing the query, submitting the query, browsing the results, and eventually accessing those Web pages he/she is looking for.

Each operation then translates into a set of commands at the Command Layer. A command corresponds to an event detected by the program, for example the insertion of a character, a click of the mouse, or the selection of an icon on the screen.

The user sends commands to the computer by performing a suitable number of actions at the Action Layer. An action is performed by interacting with the input device and it corresponds to simple acts such as pressing a key on the keyboard to write a character, or executing a "double click" by pressing twice the left button of the mouse.

Notice that, for able-bodied people, there is typically a one-to-one correspondence between the actions and the commands, i.e., each action executed results in a command sent to the computer. This is not necessarily true for disabled people who use alternative input devices. For example, typing a single character (the command) using a binary switch can require to operate the switch (the action) several times.

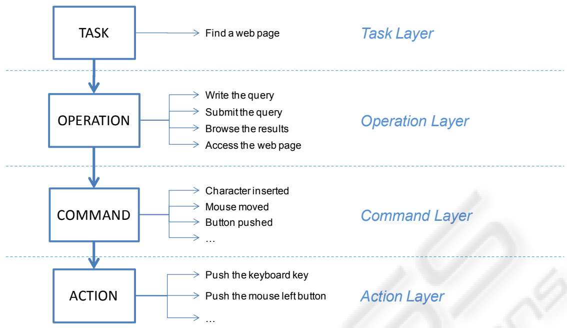

Figure 1: A hierarchical model for human-computer interaction.

We now use the model of interaction described above to classify existing approaches that have been proposed in the literature to overcome the digital divide for motion impaired people. Depending on the layer of the stack to which these approaches refer, we distinguish between Action Layer Approaches and Command Layer Approaches.

Action Layer Approaches focus on the design and realization of alternative input devices that allow disabled people to interact with standard software applications. In terms of the interaction model of Figure 1, these approaches allow motion impaired people to perform the same set of commands as the able-bodied ones by means of a different set of actions. For example, an impaired user might move the mouse cursor by means of the voice or by moving the eyes. Depending on the user's disability and on the actions he/she can perform, different devices have been considered in this context. Examples include speech/sound based interfaces (Manaris, McGivers & Lagoudakis 2002), (Sporka, Kurniawan & and Slavík 2006), tongue control (Struijk 2006), eye movement detection (Fejtová, Fejt & Štepánková 2006), EMG interfaces (Chin, Barreto & Alonso 2006), and light-spot operated mouse (Itoh 2006).

The advantage of Action Layer Approaches is that they do not require the software to be modified, and therefore the impaired people can potentially use any computer application. Unfortunately, there are some drawbacks to take into account within these approaches. One disadvantage is that a long training is required to reach a good level of usability. Also, obtaining an alternative input device that can completely replace keyboard and mouse can be a difficult result to achieve, especially for individuals with particularly severe disabilities. Therefore the execution of the commands remains a major bottleneck for an efficient interaction.

Command Layer Approaches address the above mentioned problem by using alternative input devices, the effectiveness of which is enhanced by means of software adaptation layers. These software layers act as a bridge between the standard applications and the input devices. An example of this approach is the use of scanners (Ntoa, Savidis & Stephanidis 2004). Scanners highlight software controls (for example software buttons or menu items) in a predefined order. The user may choose one of the highlighted controls by using an input device with just two statuses, such as a BCI or a single button. Another example is the use of Force-feedback gravity wells, i.e., attractive basins that pull the cursor to the centre of an on-screen target (Hwang et al. 2003). These techniques are designed to help users who have tremor, spasm, and co-ordination difficulties to perform "point and click" tasks more quickly and accurately.

Referring to Figure 1, Command Layer Approaches allow the user to execute the same operations as in the standard interaction, but they require him/her to perform a different set of commands. As an example, consider the operation of sending a query to a search engine. With standard input devices, the following commands must be executed: "Move the mouse to the search button" and "Press the button". The same operation performed with a scanner consists of the scanner highlighting the search button and the user executing a single command, namely "Activate the highlighted button".

The major drawback of Command Layer Approaches is that although the actions to be performed by the user on the input devices are in general reduced or simplified, the time needed to execute a single command typically increases. For example, pressing a button using a scanner requires significantly more time than pressing the same button with a mouse, due to the time needed to scan the whole set of command options. Also, to offer a seamless integration between the adaptation layer and any application software, the latter should adhere to precise software design rules that in most cases have not been taken into account in the design of the application software.

3 THE PROPOSED APPROACH

The approach that we propose aims at overcoming the main disadvantages of the Action Layer Approaches and of the Command Layer Approaches that have been described in the previous section. The main characteristic of this approach is to act at the Operation Layer of the hierarchical model of Figure 1. The idea is to change the set of operations associated with the execution of a task in such a way that the total number of corresponding commands is reduced. Reducing the number of commands aims at compensating for the loss of efficiency that a motion impaired person must pay in executing them because of the limited number of statuses available in his/her alternative input device.

To achieve this goal, we plan to exploit enhanced Information Visualization technologies. Information Visualization aims at conveying abstract information through visual representations of data. Visual representations, obtained by using geometric primitives and transformation, colours, and other visual objects, translate data into a visible form that highlights important features that would be otherwise hardly identifiable or even hidden.

It follows that, when compared with different possible representations of the information space associated with a task, visual representations are more efficient in conveying information. This is due to two main reasons. On one hand, visual representations take advantage of the human eye's broad bandwidth connection into the brain to allow users to see, explore, and understand large amounts of information at once. On the other hand, the use of visual objects makes the acquisition of information more intuitive and immediate, and therefore the cognitive elaboration is reduced.

Thus, the main novelty of our approach is to make the end-user interact with a computer in which data are presented in a non-traditional way by means of sophisticated diagrammatic interfaces. All the previous approaches that we are aware of, aim at reducing the discomfort of motion impaired within the classical iconic representation of the data offered by traditional operating systems. They do not try to compensate the reduced expressiveness of the input devices by enlarging the amount of information that can be visually processed by the end user in the same time frame.

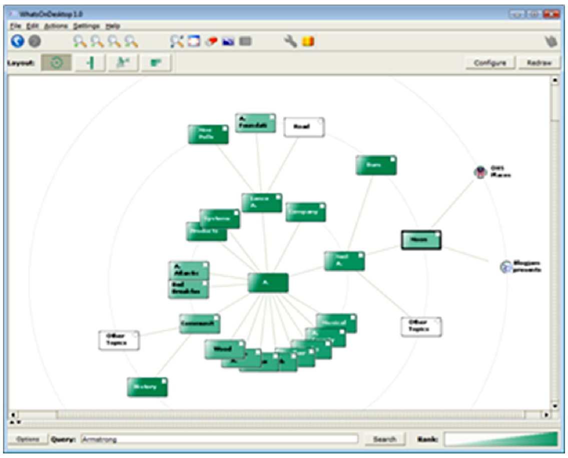

Figure 2: Example of diagrammatic interface that reduces the search commands to be executed when accessing a page on the Web.

Consider for example the task of searching a page on the Web. A possible set of operations is {*write the query, submit the query, scan the list of results, access the Web page*}. One of the efficiency bottlenecks for the motion impaired would be scanning the list of results which can be very long. Our approach is to change this critical operation. Traditionally, search engines results are presented as a list of pages that are sequentially scanned. An alternative presentation could be the following: pages are grouped into different categories, where each category contains pages that are semantically coherent; furthermore, possible relations between different categories are explicitly showed. In this scenario, the number of commands associated with the browsing can be significantly reduced because the information space to be searched by the end user looking for a page is naturally narrowed by selecting categories or sub-categories and by discarding large quantities of uninteresting pages with a single command. A snapshot of a possible diagrammatic interface for this specific operation is given in Figure 2. The figure represents the output of a visual Web search engine called WhatsOnWeb (Di Giacomo et al. 2007), developed by two of the authors in a previous research project. Even if WhatsOnWeb was designed, implemented, and tested considering only able-bodied users, we think that the system highlights some aspects that can be reused and adapted to support our approach. It is worth mentioning that WhatsOnWeb has already been adapted to be used by visual impaired (Rugo et al. 2009).

4 REFERENCE ARCHITECTURE

In this section we describe a reference architecture that we envision it can be used to implement the principles of the proposed approach. The reference architecture is given in Figure 3.

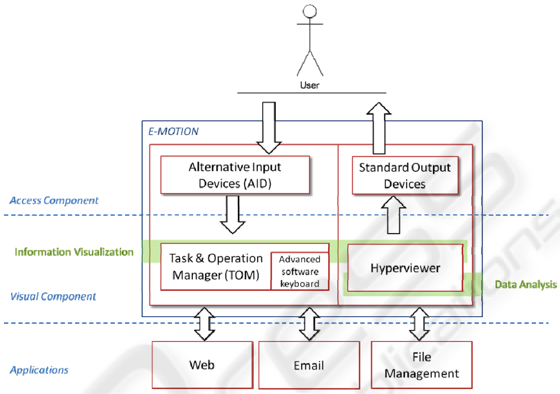

Figure 3: A reference architecture.

As shown in the figure, the interaction between the end-user and the standard applications is performed through an Access Component and through a Visual Component.

The Access Component consists of the input and output devices. For the input, Alternative Input Devices (AID) are used while no alternative devices are foreseen for the output because we assume that standard monitor and audio devices are equally suitable for motion impaired people.

The Visual component consists of adaptation software that is placed on top of standard applications by using light software interfaces. It offers a novel interaction paradigm between users and computer applications based on expressive and highly informative diagrammatic interfaces that will be easily accessible by the alternative input devices.

As illustrated in Figure 1, the Visual Component consists of two modules. The *Task & Operation Manager (TOM)* visually supports the user with a diagrammatic interface that suitably represents the set of operations associated with the task he/she wants to perform. The *Hyperviewer* represents the output of the standard application (e-mail, Web browsing, file handling, and so on) in a non-standard way. This module returns multiple interactive representations that allow the user to significantly enlarge the amount of information that he/she can process in the time unit.

4.1 The Task & Operation Manager

The Task & Operation Manager (TOM) is depicted in Figure 4.

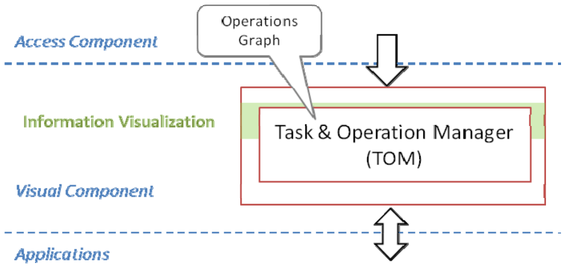

Figure 4: The Task & Operation Manager.

Once a task to be performed has been selected, TOM requires the user to select the options related to the task he/she wants to perform. Then it automatically computes the minimum number of Operations necessary to complete the Task with all the selected options, and it guides the user in the execution of such operations by means of a diagrammatic interface. More precisely:

1. TOM presents the computer functionalities as a set of possible tasks. For example, this could be a control panel where the tasks of sending e-mail, searching the Web, finding a file in the computer, and so on, are displayed. The user can select one of these tasks, e.g., via an alternative input device and a scanner.
2. Once a specific task has been chosen, TOM presents the set of operations associated with the task as a network of interconnected operations. This network is displayed as a directed acyclic graph called *Operation Graph*. The vertices of the Operation Graph are the operations associated with the task. The edges describe the logic or temporal dependencies between operations.

The user interacts with the Operation Graph by selecting one or more operations associated with the selected task. In response to this selection, TOM automatically triggers and activates the remaining operations needed to complete the task. The user will then be guided through this sequence of operations. TOM presents easy alternatives when necessary but, whenever possible, it automatically sends execution commands for those operations which do not require a new interaction with the user.

As an example, consider the task of sending an e-mail message. TOM interacts with the user by means of a diagrammatic interface that contains an Operation Graph of the type shown in Figure 5.

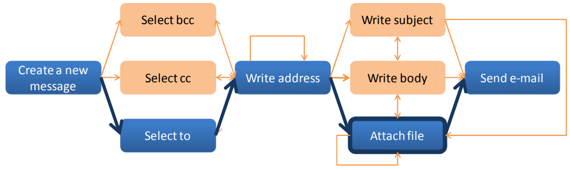

Figure 5: Example of the Operation Graph associated with the task of sending an e-mail.

The user can select one or more nodes of the Operation Graph. For example he/she can select the node "Attach file" (having thicker borders in Figure 5). TOM would then automatically compute the most convenient execution path from the source node "Create new message" to the sink node "Send e-mail" in such a way that this path contains all the required nodes. The most convenient execution path selected by TOM is highlighted with thicker arrows and darker nodes in Figure 5.

By means of such a path, TOM is capable of: (1) reducing the effort in specifying the input by requiring the user to enter as few information as possible and (2) reducing the number of operations, by guiding the user through just those operations which are strictly necessary to complete the task. We also believe that the Visualization of the Operation Graph can help the user to quickly select the desired operations. Indeed, the Operation Graph can give an overview of the operations involved in the selected task in a more concise manner, as opposite to providing the user with a series of successive selections of alternative options.

4.2 The Hyperviewer

The Hyperviewer is depicted in Figure 6. The Hyperviewer makes use of sophisticated algorithms of Information Visualization to present the

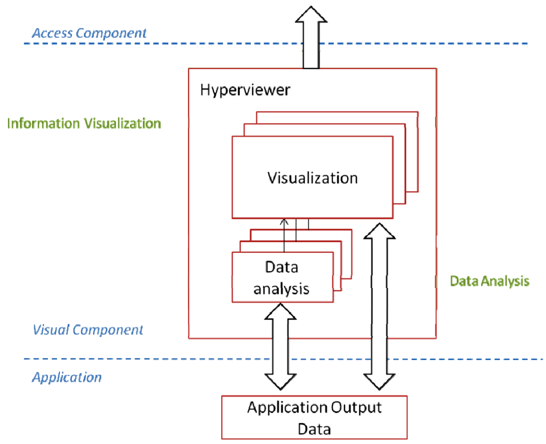

Figure 6. The Hyperviewer.

application output data in a non-traditional, compact, and easy-to-browse manner. As a result, the number of operations that the user must execute to elaborate these data and find the wanted information is reduced. To this aim, it is important that the Hyperviewer contains not only libraries of Information Visualization algorithms of but also implements enhanced primitives of data analysis.

An example of a non-traditional visualization of the results of a Web search engine has been given at the end of Section 3 . By interacting with the diagram of Figure 2, the number of commands that the user must execute in order to retrieve a page on the Web is reduced, because the semantic clustering makes it possible to navigate the data in a non-sequential manner. At each step of the navigation, the user selects a category of his/her interest and discards all the (possibly numerous) pages that are not semantically related with this category. This mechanism is made possible by strongly integrating on-line technologies of semantic clustering with sophisticated algorithms of network visualization.

As another example of a non-traditional interaction paradigm offered by the Hyperviewer, let us consider the task of finding a file in the computer. The Hyperviewer can show the file system according to multiple visualizations. For example, one visualization represents the hierarchical organization of the file in the file system while other visualizations categorize files based on the creation time or on the file types (images, documents, etc.). Files can also be semantically clustered on the basis of their content. Each visualization can be browsed by drilling up and down the categories. The user can choose whether to use just one visualization or to switch among different visualization. Clearly, the Hyperviewer maintains automatic consistency and synchronization of these different representations, which means that selection/filters applied to one diagram affect also the others. These multiple synchronized views, together with efficient selection algorithms coupled with the available alternative input device, provide a comprehensive and intuitive view of the data, and hence reduce the required human computer interaction.

5 CONCLUSIONS

We described a novel approach to overcome some of the interaction difficulties between motion impaired people and computers. Our idea is to use enhanced Information Visualization techniques to easily convey large amounts of information within an interaction time frame.

Our plan is to implement the proposed reference architecture and to carry out an extensive experimental validation of the approach. This will also be possible with the collaboration of associations of disabled that already expressed their interest in our ideas.

REFERENCES

Chin, C, Barreto, A & Alonso, M 2006, 'Electromyogram-Based Cursor Control System for Users with Motor Disabilities', *Computers Helping People with Special Needs*.

Di Giacomo, E, Didimo, W, Grilli, L & Liotta, G 2007, 'Graph Visualization Techniques for Web Clustering Engines', *IEEE Trans. Vis. Comput. Graph.*, vol 13, no. 2, pp. 294-304.

Fejtová, M, Fejt, J & Štepánková, O 2006, 'Eye as an Actuator', *Computers Helping People with Special Needs*.

Hwang, F, Keates, S, Langdon, P & Clarkson, PJ 2003, 'Multiple Haptic Targets for Motion-Impaired Users', *CHI 2003*.

Itoh, K 2006, 'Light Spot Operated Mouse Emulator for Cervical Spinal-Cord Injured PC Users', *Computers Helping People with Special Needs (Lecture Notes in Computer Science)*.

Manaris, B, McGivers, M & Lagoudakis, M 2002, 'A Listening Keyboard for Users with Motor Impairments — A Usability Study.', *International Journal of Speech Technology*, pp. 371-388.

Ntoa, S, Savidis, A & Stephanidis, C 2004, 'FastScanner: An Accessibility Tool for Motor Impaired Users', *Computers Helping People with Special Needs*.

Rugo, A, Mele, ML, Liotta, G, Trotta, F, Di Giacomo, E, Borsci, S & Federici, S 2009, 'A Visual Sonificated Web Search Clustering Engine', *Cogn. Proc.*, pp. 286-289.

Sporka, J, Kurniawan, H & and Slavík, P 2006, 'Acoustic control of mouse pointer', *Univers. Access Inf. Soc.*, vol 4, pp. 237-245.

Struijk, LNSA 2006, 'A Tongue Based Control for Disabled People', *Computers Helping People with Special Needs*.