A Study of Input Device Manipulation Difculties

Shari Trewin
University of Edinburgh
Department of Articial Intelligence
80 South Bridge
Edinburgh
EH1 1HN
Scotland
shari@aisb.ed.ac.uk

ABSTRACT
People with a motor disability affecting their use of the keyboard and/or mouse often tend to make unintentional input errors. Little or no quantied data exists on physical errors in the use of standard computer input devices, particularly with respect to motor disabilities.

Such information, if available, could be used to develop techniques for automatic recognition of specic difculties. Once recognised, many can be reduced or eliminated by appropriate system and application conguration.

This paper describes the pilot study for an experiment intended to gather detailed information about input errors made with keyboards and mice. This work is a step towards provision of dynamic, automatic support for the conguration of systems and applications to suit individual users.

Some initial results from the pilot study are presented, including an assessment of the experiment design and a summary of some interesting characteristics of the data gathered so far.

KEYWORDS: keyboard, mouse, errors, physical disability, input devices, input logging.

INTRODUCTION

It is widely recognised that people with physical disabilities can have difculty in accurately manipulating the standard computer input devices: the QWERTY keyboard and the mouse. However, despite the fact that a vast number of access facilities and alternative input devices have been developed to get around these difculties (Brown, 1992) (CALL Centre, 1994), very little quantied data on their precise nature exists.

Physical errors in the manipulation of input devices are referred to here as performance errors, to distinguish them from cognitive and other errors. Examples of performance errors are missing keys, striking adjacent keys in addition to the one aimed for, pressing keys for too long producing repeated letters, moving the mouse while double clicking, and dropping the mouse button while dragging.

It is conjectured that particular performance errors caused by physical disabilities will exhibit patterns which can be recognised. For example, where a key adjacent to the intended key is additionally struck, the timings of the intended and accidental key presses may distinguish them from those of deliberate, separate key presses. If this conjecture is true for several different performance errors, then the information gleaned can be used to provide support for users experiencing difculties of this type.

This support could, for example, include the conguration of input devices to accommodate performance errors in the input stream. There are a number of conguration options available for keyboards and mice which can reduce or eliminate some varieties of performance error. If the need for one of these options could be dynamically recognised, then the option could be activated automatically, improving the usability of the computer system as a whole. Machine support for conguration is an attractive proposition, since an individual's needs may change dramatically over time, and human support is often in short supply.

Before an automatic conguration tool, or similar support can be developed, a good understanding of the identifying characteristics of performance errors in an input stream is required. An investigative study is proposed, the goals of which are: to examine performance errors from a number of different users with disabilities, producing data on the nature and frequencies of performance errors occurring; and to test the hypothesis that these performance errors are often characterised by recognisable patterns in the input stream.

This paper describes the pilot test for the study, and summarises some preliminary results obtained. The next section explains the lack of existing data of this kind, and the consequent need for this study. The study itself is described, followed by details of the technique used to log the input events generated by the subjects. Initial results obtained from the data gathered, and some preliminary conclusions from these observations are presented. The paper concludes with a summary of the implications for the design of the full study, and for the prospects of applying these results to improve the usability of standard input devices.

EXISTING DATA

Even for non-disabled people, data on physical input errors is difcult to nd. HCI research into human errors has generally focussed on cognitive errors and their causes, ignoring physical errors of the kind produced by erratic motor control. This despite evidence that 'keyboarding errors' are an important and signicant source of errors, particularly in large databases (Peterson, 1980).

Finding data produced by people with disabilities is even more difcult. Evaluation of keyboards, mice and applications is almost invariably carried out either with expert users, or with users who are typical of the intended user population, or of sub-classes of that population (Karat, 1988) (Vassiliou, 1984) (Monk, 1987). Disabled users cannot be viewed as a homogeneous class of users in this way. Proportionally, the expected number of physically disabled users of a general purpose system is small, and because their consideration in the evaluation process is difcult to justify economically, they are rarely considered.

One might expect that appropriate data is available from the evaluation of the computing needs of people with physical disabilities. However, when people are assessed in order to establish an appropriate input device and software, their keyboard and mouse skills are assessed by observation, and usage data is not recorded (Broadbent & Curran, 1992) (Lee & Thomas, 1990). In fact, the author is not aware of any input data recorded for physically disabled users of keyboards and mice.

Real input data is essential in order to be able to establish what errors do occur, and their relative frequencies. It will also allow examination of associations between specic errors and features of the stream of input events. The goal of the proposed study is to gather some accurate data from which this information can be extracted.

STUDY DESCRIPTION

Before launching into a major study of performance errors, it is important to ensure that the proposed experimental method and the data collection tools are adequate. To this end, a small pilot study was carried out. This test allowed the experiment design and data collection software to be assessed and rened prior to the major study. This section describes the methodology proposed for the study, and tested in the pilot study.

Subjects

The subjects chosen for the full study will form a set of case studies, rather than a representative cross-section of people with motor disabilities. The study will not focus on any particular disability, but aims to cover a broad range of people, experiencing a variety of keyboard and mouse difculties. It is anticipated that there will be overlap between the sets of difculties caused by different disabilities.

The intention of the study is to capture the normal performance of each user, and so it is essential to minimise the effects of unfamiliarity with computers and word processors. The subjects used in the pilot study were already familiar with using computers to do word processing. In the full study, subjects who have little or no previous experience will be given the opportunity to practice the required skills in advance of data recording. This will not eliminate all learning effects, but should go some way towards reducing their effect.

A related issue is that those subjects who do have computing and word processing experience may use a specialised conguration which affects the behaviour of the keyboard and/or the mouse. For the purposes of this study subjects may only use those options which do not affect the stream of input events reported. Disallowed options include sticky keys and implementation of a key acceptance delay. Subjects used to using these options will again need some time to familiarise themselves with the new behaviour of the computer, and the conguration they are used to will be taken into account when interpreting the data recorded.

Materials

A set of three different tasks was developed and tested in the pilot study. The tasks are based on Apple Macintosh 475 8/160 computers[1], and the ClarisWorks^{tm}[2] word processing package. The computers are equipped with a variety of accessibility hardware and software, including adjustable tables, wrist rests, a variety of different designs of mouse, and screen reading and magnication software.

The use of this simple equipment, particularly in nding the most appropriate positions for the monitor, keyboard and mouse, can prevent many performance errors occurring, by making the physical environment as comfortable as possible. It also minimises the effort required to operate the computer, which decreases fatigue.

The primary focus of this research is those performance errors that remain even after the physical environment has been optimised, and so providing a comfortable environment for the subjects is essential.

Subjects perform each task in their own time, and are allowed rests between tasks if desired. In the rst task, users type out a set passage twice. The rst time without any error correction, and the second time with error correction. The former provides easily analysable data, while the latter is a more realistic sample of typing, and introduces problems that occur when errors are made in corrections. It also gives some indication of the time spent correcting errors. The passage is constructed so as to test the user's ability to reach all parts of the keyboard, and to use the shift key in conjunction with keys in a variety of positions. It requires a minimum of 547 keypresses, including 25 uses of the Shift key.

The second task focuses on use of the mouse, and tests skills such as pointing, clicking, double and triple clicking, and dragging.

The nal task is an editing task, which requires use of both

[1] The computers are owned by Lothian Regional Council, and access was kindly provided by the Hands-On-Technology Group at the South Bridge Resource Centre.

[2] ClarisWorks is a registered trademark of Claris Corporation.

the keyboard and the mouse. A pre-typed passage is edited, the editing tasks covering the same set of basic skills as are examined in the typing and mouse tasks. This task is included to allow identication of higher-level patterns associated with typical editing tasks, particularly those involving combinations of keystrokes and mouse movements.

After subjects one and two had completed the study, the mouse and editing tasks were modied. The instructions were simplied, and passages altered to highlight the targets. All scrolling, and other actions that alter the positions of text items on the screen, were moved into the editing test. This means that the positions of the targets in the mouse test are constant, making analysis of pointing actions much easier. The scrolling in the editing test was designed so that all pointing targets would still remain in a constant position, as far as was possible. The tests still cover the same basic set of skills.

Procedure

For each subject, the environment was made as comfortable as possible. Subjects were then allowed time to become familiar with the environment to be used for the test. All terms used in the tasks, such as 'dragging' and 'scroll bar' were explained, and subjects had the opportunity to practice any skills with which they were unfamiliar. For a subject who is easily fatigued, this familiarisation would either be carried out at an earlier date, or a reasonable rest allowed before starting the study.

The tasks were administered by the same observer for each subject. She explained each task as it was presented, and provided verbal help where the subjects required it. The time taken to complete the tasks depends very much on the subject. In the pilot study, subjects took from 30 minutes to two hours to complete all the tasks.

To minimise fatigue, the tasks were presented in the order: typing 1, mouse 1, editing, mouse 2, then typing 2. The text window was of a xed size and was placed in a standard position on the screen, so that screen co-ordinates can later be related to the objects on the screen. Where required, rests between tasks were allowed. For each subject, the following data was recorded:

- An automatically generated log of input events, containing the times of every key press and release, and mouse button press and release. Mouse movements are also recorded. The logging software is described more fully in the next section.
- A video of the subject performing the tasks. This is useful in establishing the actual performance errors that occurred.
- Observations made during the word processing tasks. For each subject, the same observer recorded impressions and particular examples of the keyboard and mouse difculties experienced by the subject. These observations are later combined with the video evidence to establish what performance errors actually occurred.
- Background information about the subject. This includes the mouse design they were using, previous experience with computers and word processors, the set of conguration options they usually use (if known), and their reported levels of fatigue and ease of performance of the tasks.

AUTOMATIC LOGGING OF INPUT EVENTS

One of the conjectures to be tested by this study is that some performance errors can be recognised by characteristic patterns they produce in the input event stream. Patterns could appear at a number of levels in the input stream. For example, at a high level, difculties with tasks such as dragging could be indicated by repeated consecutive drags starting from approximately the same position. At the lowest level of detail, accidental additional key presses could be recognised by their timings relative to deliberate key presses. In order to allow investigation of performance errors at all levels, a detailed log of keyboard and mouse events is required.

To provide information about accidental key press timings, and the length of key presses, the log must record the time of every key down and key up event, and must distinguish between events generated by holding down a key, and those generated by repeatedly pressing a key. If difculties in pressing more than one key simultaneously are to be examined, the same information must also be available for control keys such as Shift and Option.

Similarly, timings and locations for the mouse down and mouse up events of mouse clicks are also important. The log should record the path taken by the mouse between clicks, so that higher level patterns in mouse movement can be identied. It is then possible to look for correlations between these patterns, and the tasks being performed by the user.

Because of the level of detail required, existing input recording packages were either unsuitable or too slow, and so a fast, unobtrusive logging mechanism `InputLogger` was purpose built for this project.

`InputLogger` is a Macintosh specic program which records log data in a le for later analysis. Each line in the log represents an input event, and includes the following information:

```
<time> <eventType> <eventData>
```

The time is measured in ticks: sixtieths of a second. The event type could be any one of: key down, key up, key repeat, mouse down, mouse up, or mouse move. For keyboard events, the event data recorded is the hexadecimal ASCII code for the relevant key. Unused ASCII codes are employed to specify keys such as Shift which have no ASCII code of their own. For mouse events, the event data is the mouse position on the screen, recorded as separate x and y co-ordinates. A mouse movement is recorded whenever the mouse changes direction by more than 5 degrees. This provides an accurate record of the path taken by the mouse, while reducing the number of mouse positions recorded.

The program is implemented as a system extension, and logging of input events is switched on and off via a control panel. Data is logged regardless of the application being used. The extension operates by trapping all events before they are reported to applications. Events of interest are copied to an internal store managed by Apple's `Audit` library. The control panel then independently reads events from this store and writes the log le.

On Macintosh machines, pressing control keys such as Shift or Option does not generate a keyboard event, but modies future keyboard events. Similarly, mouse movements also do not generate Macintosh input events. Instead, the mouse position and status of the control keys are checked by the control panel whenever other events occur, including `null` events. Changes in status are written directly into the log le.

PRELIMINARY RESULTS

The pilot study was performed with four volunteer adult subjects. Subjects one and two have motor disabilities caused by stroke, and type with the right hand only, although subject two can use his left hand to operate the Shift key. Subject three has neurological damage causing muscle wastage and spasticity in his hands, and types using several digits on each hand. Subject four has impaired dexterity due to incomplete tetraplegia. He types with his right hand, using his left to operate the Shift key.

All four subjects chose to use the standard design of mouse. All were familiar with the use of computers and word processing packages, although not necessarily with the Macintosh and ClarisWorks environments.

Data analysis is still in progress. The remainder of this section describes the analysis of the data and gives some interesting initial observations. A fuller description of the experimental methodology, analysis and the results can be found in (Trewin, 1996).

Analysis

The goal of the analysis is to nd performance errors, and input patterns which could indicate performance errors. Until patterns indicating performance errors are found, this task cannot be automated, and so the identication must be done by hand. Errors are found by examining places where the input differs from that dictated by the task. These errors are then categorised only some of them will be performance errors. The remainder are placed in a single error class. This class includes cognitive errors such as spelling errors, or errors caused by a misunderstanding of the task, and any other errors, including those caused by external events such as the subject being nudged, or distracted in some way.

All errors are found by combining the observations made with the video evidence and the recorded log le, and comparing these with the expected input, according to the task. Before the log le is examined the entries are sorted by their time stamps, and then ltered to transform them into a more readable format.

This ltered log le is then annotated, to mark out the performance errors and other important features. Log les are also annotated with indications of the tasks currently being performed, so that, for example, the log for the second typing task will include an indication of where error corrections start and nish, and mouse task logs will include an indication of the target of a positioning movement, or the type of task currently being performed. The annotation scheme is documented fully in (Trewin, 1996).

The annotated log le is automatically processed to extract summary statistics, and to transform the data into formats appropriate for visualisation and further statistical analysis.

The keyboard and mouse data are currently analysed separately. Analysis of patterns involving both the mouse and the keyboard has yet to be carried out. Such patterns, if they exist, should appear in the editing data. An example of such a pattern would be the use of the mouse to roughly position the cursor, combined with the arrow keys used for ne positioning.

Keyboarding

Table 1 summarises the typing tests for subjects one, three and four. Subject two had no keyboard difculties, and is excluded from this analysis. The table shows the total number of keypresses made, time taken, total number of performance errors (P) and other errors (O) made, and the time spent correcting errors of each kind. For all subjects, the majority of errors were performance errors, and where errors were corrected, more time was spent correcting performance errors than other errors.

Subject / Test	No. of keys	Time (secs)	Total Errors		Correcting Time (s)	
			P.	O.	P.	O.
1/1	588	929	39	18	0	0
1/2	1348	1844	36	24	334	260
3/1	550	228	49	3	0	0
3/2	637	308	48	7	56	17
4/1	565	276	16	2	0	0
4/2	559	271	7	2	0	0

Table 1: Summary of the typing tests

Non-performance errors included misreading the passage to be copied, forgetting to type some words, and using the Caps Lock key as a replacement for Shift when trying to type punctuation.

	S1		S3		S4		Total
	1	2	1	2	1	2	
ad	38	34	9	4	4	0	89
db	0	0	30	35	0	0	65
mi	11	6	10	3	2	1	33
dr	0	3	0	1	8	6	18
re	0	0	0	1	2	0	3
tr	0	0	0	2	0	0	2

Table 2: Observed performance errors

The performance errors observed for each subject's two tests, and total number of errors of each kind are summarised in Table 2. Performance errors are classied as follows:

1. Additional Key Errors (ad): A key adjacent to the intended key is activated.
2. Doubling Errors (db): An alphanumeric key is unintentionally pressed for longer than the default key repeat delay (16 ticks).
3. Missing Errors (mi): The intended key is missed entirely.
4. Dropping Errors (dr): The subject fails to press two keys simultaneously (e.g. use of the Shift key).
5. Remote Errors (re): A key not adjacent to any intended key is pressed (e.g. the subject accidentally leans on a key).
6. Transposition Errors (tr): Two keys are transposed.

As can be seen from Table 2, although each subject made performance errors of several different kinds, each individual tends to be prone to one or two specic types.

The most commonly occurring performance error was that of pressing two keys at once, categorised as type ‘ad’. Subject one was particularly prone to this, and an example of his typing is shown as the leftmost of the two graphs in Figure 1. For this subject, the timing patterns of accidental additional key presses proved to be highly distinctive, with the unintended key presses overlapping with the intended key press in all 59 instances where the intended key was actually pressed[3], while the deliberate key presses are well separated in time. Intentional overlapping key presses were conned to the use of the Shift key and did not occur in his normal typing. Furthermore, in 83% of these cases, the intended key was the last to be raised.

A similar pattern was exhibited by subject four, who also types with one hand and never overlaps keystrokes except when using modier keys or where ‘ad’ type errors occur.

These preliminary results are encouraging for the prospects for automatic recognition, and perhaps correction, of this type of keyboard difculty for those whose normal typing consists of non-overlapping keystrokes. Statistical knowledge-based methods, incorporating knowledge of English digram frequencies or a dictionary, could potentially be used to calculate the intended letter. The positioning of the errors on the keyboard, and direction in which the errors occur, may also provide further material on which to base decisions about these errors.

However, this pattern does not apply to subject three, who types using several digits on each hand. The right hand graph in Figure 1 shows an example of subject three’s typical keystroke timings. Many of the keystrokes overlap, and the example contains only one error of type ‘ad’. Since subject three made only 13 such errors, more data is required in

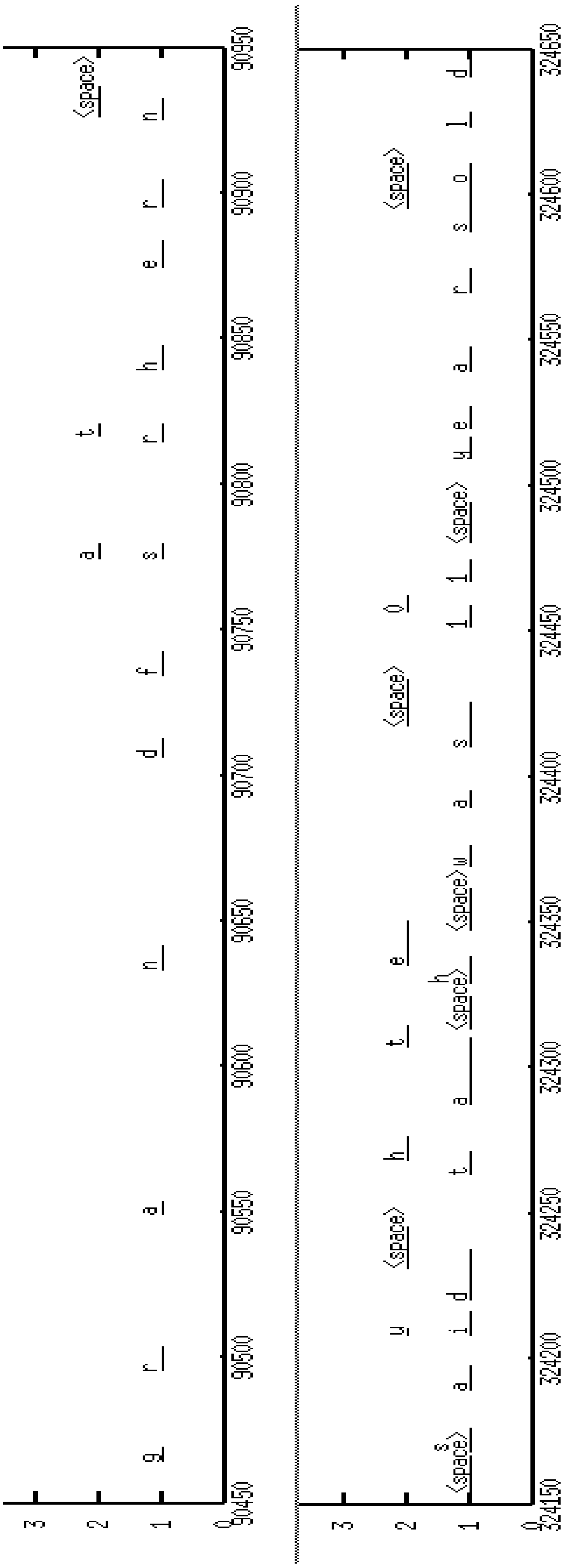

Figure 1: Example keystroke timings (the X-axis represents ticks, the Y-axis separates out simultaneous key presses for presentation purposes)

[3] In the other 13 examples of this error, the intended key was missed.

order to ascertain whether any pattern exists. No patterns based on the keystroke timings are immediately obvious. For those typists who, like subject three, naturally overlap their keypresses, it may be difcult to detect errors of this type.

Doubling errors, caused by pressing keys for too long, were the greatest difculty for subject 3. This error is extremely easy to detect, given the times of key down and key up events. Here, a doubling error was recorded when a key was pressed for 16 ticks or longer, which is the default key repeat delay on a Macintosh. Such errors can be eliminated by extending the key repeat delay, or deactivating the key repeat facility.

Dropping errors occur when the Shift key, or another modier key, is raised before the key to be modied has been pressed down. These errors were most noticeable in the typing of subject four. 14 dropping errors occurred, and in each of these, the Shift key was raised not more than 2 ticks before the modied key was pressed down. In the log les for this subject, wherever the `Shift` key was pressed down alone, this was a dropping error. An example of subject four's typing, showing two dropping errors, is shown as the left graph in Figure 2. For this subject, dropping errors are easily recognisable from the log le data.

Subject one also had difculty in pressing two key simultaneously, and the low number of dropping errors recorded is due to his use of the `Caps Lock` key as an alternative, even when typing punctuation. An example for subject one is shown as the right hand graph in Figure 2. First, he managed to hold down the key, but then struck the left key instead of '?'. After attempting to correct this mistake, he tried again, but dropped the Shift key, producing '/' instead of '?'. After correcting this mistake he succeeded in producing '?'.

However, because subject one is used to using the Sticky Keys facility, he often pressed the Shift key in isolation, forgetting that Sticky Keys were disabled for the purposes of the experiment. The pattern observed for subject four, therefore, does not apply to subject one. However, when subject one deliberately pressed the Shift key, it was an indication that the next letter was to be upper case, and so despite the fact that no dropping error had occurred, the conclusion that the subject may benet from using the Sticky Keys facility would still be valid.

Other, higher level patterns may also be useful in recognising dropping errors. For example, when typing text, a full stop followed by a space, a Shift key press and then a letter is likely to be a dropping error. Further investigation, and more data, is necessary in order to establish whether dropping errors produce reliable patterns in the input stream.

The remaining types of error remote keypresses and transposition errors were not observed in large enough numbers to allow examination of their properties at this stage. In the full study, more examples may be observed. Those errors that occur infrequently will not have a large effect on the usability of computer input devices, and so it is less important to nd methods of recognising them. It is hoped that the full study will give an impression of which errors occur most frequently, so that effort can be concentrated on nding ways

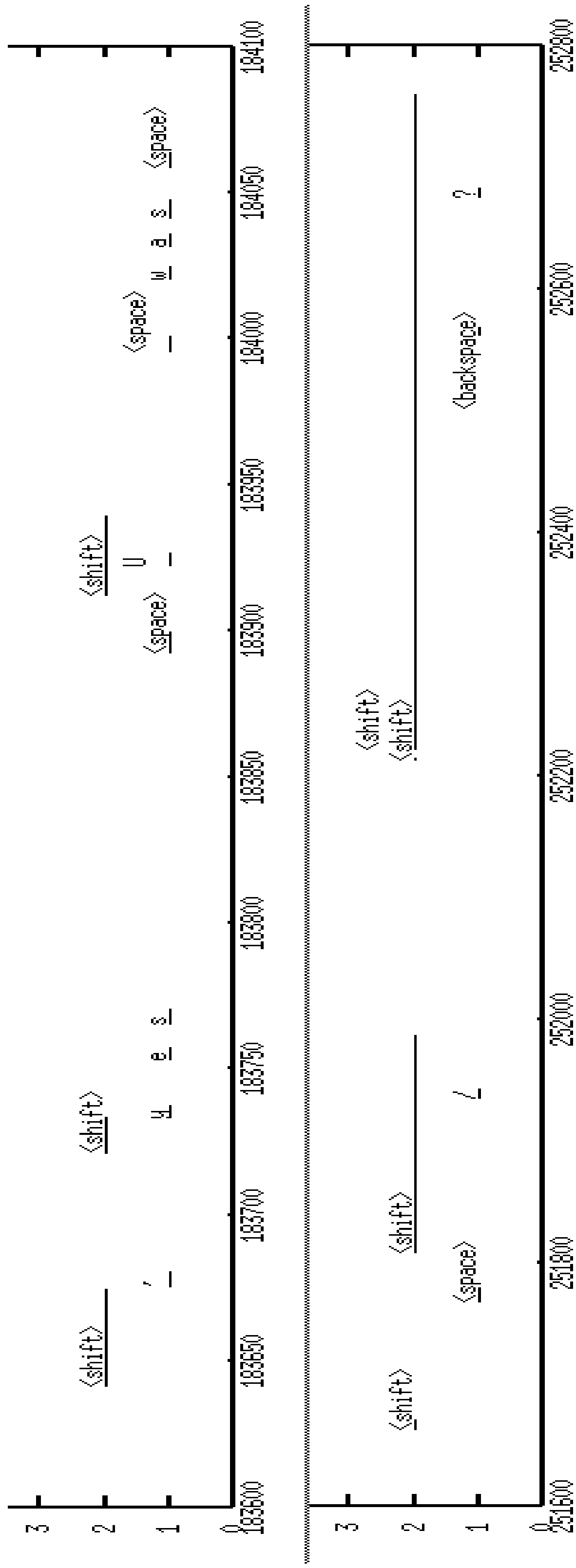

Figure 2: Dropping errors (the X-axis represents ticks, the Y-axis separates out simultaneous key presses for presentation purposes)

of eliminating them.

Mouse Usage
All subjects chose to use the standard mouse design, but found it difcult to use for certain tasks. Dragging and choosing items from hierarchical menus were reported as the most difcult mouse tasks to perform. This supports previously reported ndings that dragging is a more difcult task than pointing (MacKenzie et al., 1991). The performance of subject two improved noticeably the second time the mouse task was performed, despite reporting tiredness. This may be partly explained by some difculty in understanding the mouse tasks as they were specied, and partly by practice effects. The other subjects did not report tiredness, and their performance remained constant for the two mouse tasks.

The following mouse-based performance errors are recognised:

1. Click length: A click may be too long or too short. Click length is particularly important while using scroll bars.
2. Time between clicks: If this is too long, an intended double click is interpreted as two separate clicks. If it is too short, two clicks are interpreted as a double click.
3. Click movement: If the mouse is not held still within or between clicks, then a the action may be interpreted as a drag, rather than a click.
4. Positioning: The subject does not accurately point to their target.
5. Dragging errors: The subject has difculty in positioning the mouse with the mouse button held down.
6. Dropping errors: The mouse button is unintentionally released while dragging.

In addition, the path taken by the mouse from a source position to a target is also signicant, even though erratic or indirect paths are not explicitly labelled as performance errors by this classication.

One interesting feature to examine is the amount of movement between the mouse down and mouse up events while clicking the mouse. Most applications tolerate a small amount of movement within a click, but if the movement is too large, the click becomes a dragging operation. The Macintosh operating system records a click if the mouse down and mouse up events fall on the same object. Some standard Macintosh targets, such as the close box for a window, are as small as 10 pixels square. In the ClarisWorks word processor, the target area when clicking between the letters 'i' and 'l' is only 3 pixels wide when 18 point type is used. Both of these examples appear in the mouse and/or editing tests.

There are several different kinds of mouse click: single clicks, double clicks, triple and multiple clicks, presses and drags. A press is a click which species the length of time for which some action occurs. For example, clicking on one of the arrows of a scroll bar is classied as a press.

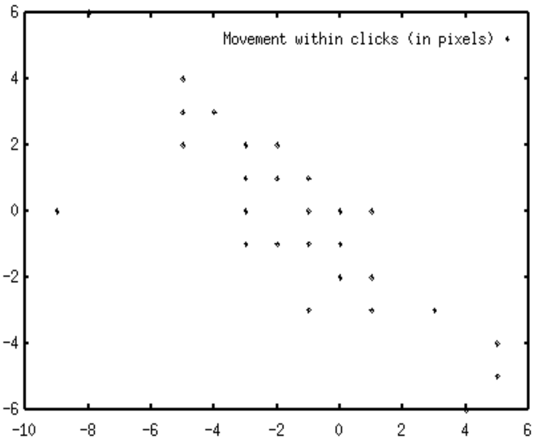

Figure 3: Cursor movement while clicking

Table 3 summarises the number of clicks of each type made by each subject, giving the number in which there was no movement, and the number in which the mouse moved during the click. Dragging operations are omitted, since the movement within them is intentional. Of the four, subjects one and two had the most difculty in holding the mouse still while clicking. Figure 3 graphs the differences in position between the mouse down and mouse up events for the 71 recorded mouse clicks by subject one. The mouse down position is (0,0) and the mouse up position is shown relative to that, in pixels. 22 of the mouse up positions were at least 3 pixels removed in one direction from the corresponding mouse down position.

Interestingly, the mouse movements made by this subject while clicking tend to follow one diagonal. This tendency could usefully be taken into account by an application accepting clicks from this subject, allowing some leeway along that diagonal.

Clicks	S1	S2	S3	S4
Single (no movement)	16	18	22	26
Single (movement)	39	8	4	5
Double (no movement)	0	2	12	8
Double (movement)	11	5	1	2
Multiple (no movement)	0	1	3	1
Multiple (movement)	4	3	1	3
Presses (no movement)	11	27	15	15
Presses (movement)	25	14	1	3
Total (movement)	79	30	7	13
Total	106	78	59	63

Table 3: Mouse click movement

Analysis of pointing and dragging actions, and click timings is still in progress.

CONCLUSIONS AND FUTURE WORK
The pilot study has provided much useful information on which the full study can build. As the study progressed, the experiment design was assessed, and some modications

were made to the mouse and editing tasks. Overall, the design proved successful, with the observations and video evidence combining to provide a good impression of what performance errors occurred.

Initially, the mouse and editing task instructions were unclear in some areas, and subjects sometimes had difculty in nding the targets they were to point to or click on. After modication of the mouse and editing tasks, the situation improved, and subjects three and four required noticeably less assistance in carrying out the tasks. The pilot study also highlighted the need to ensure that enough practice time is available, without fatiguing the subject. Potential fatigue has also been reduced by ordering the tasks so that the typing intensive and mouse intensive tasks are interleaved.

The `InputLogger` software proved effective. The timing accuracy was adequate (although it probably would not be for a fast touch typist), as was the accuracy of recorded control key presses. The logging software is unable to record the movements of the mouse while dragging. This is because the Macintosh does not generate events while the mouse button is held down. This restricts the information available about dragging and selection from menus. However, higher level patterns, such as repeated drags from similar positions, could be used to identify difculties with dragging.

The preliminary results obtained from this pilot study are encouraging. Although the passage to be typed was short, many examples of six different performance errors have been observed. It is also interesting to note that all of the subjects had very different error proles, and each was particularly prone to one or two of these error types.

When the full study is completed, the data gathered will provide information about the major difculties experienced with the manipulation of the keyboard and mouse, and their relative frequencies both within and between individuals. It may also be possible to identify correlations between different performance errors. This information will be useful to software designers and providers of access features for systems and applications.

Patterns in the timing of keystrokes, and in mouse movement while clicking, have already been observed. To test for the presence of characteristic input patterns indicating specic performance errors, more data is required. The data gathered to date indicates that some patterns will be specic to individuals, while others may be globally applicable.

Where such patterns exist, they could be used as a means of identifying difculties a user is having with the supplied input devices. This information can then be used to choose, and even implement, an appropriate machine/application conguration for a specic user. This would enable provision of automatic support for conguration, thereby improving the accessibility of standard computer input devices and applications.

ACKNOWLEDGEMENTS

The author gratefully acknowledges the support of the University of Edinburgh in funding this research, and the members of the H.O.T. Project and Lothian Regional Council, for providing access to appropriate machines. Thanks also to the volunteer subjects, and to Helen Pain and Mike Ramscar for their time and useful feedback.

References

Broadbent, Steven and Curran, Sandra. (1992). The assessment, disability and technology handbook. North West Regional ACCESS Centre and Oldham Education Department, Oldham.

Brown, Carl. (May 1992). Assistive technology computers and people with disabilities. Communications of the A.C.M., 35(5):3645.

CALL Centre. (July 1994). Alternative access to the Apple Macintosh: Using in-built features. University of Edinburgh, 4 Buccleuch Place, Edinburgh.

Karat, John. (1988). Software evaluation methodologies. In Helander, Martin, (ed.), Handbook of Human-Computer Interaction, chapter 41, pages 891903. Elsevier Science Publishers B.V.

Lee, K. S. and Thomas, D. J. (1990). Control of Computer-based technology for people with physical disabilities: an assessment manual. University of Toronto Press, Canada.

MacKenzie, Scott, Sellen, Abigail and Buxton, William. (1991). A comparison of input devices in elemental pointing and dragging tasks. In Proceedings of CHI, pages 161166, New York. ACM.

Monk, Andrew. (1987). How and when to collect behavioural data. In Baecker, Ronald M. and Buxton, William A. S., (eds.), Readings in human-computer interaction : a multidisciplinary approach, chapter 4, page 138. Morgan Kaufmann, Los Altos, Calif.

Peterson, J. (1980). Computer programs for detecting and correcting spelling errors. Communications of the ACM, 23(12):676687.

Trewin, Shari. (1996). Gathering and analysing keyboard and mouse data. DAI discussion paper, AI Dept, University of Edinburgh, In preparation.

Vassiliou, Yanis, (ed.). (1984). Human factors and interactive computer systems : proceedings of the NYU symposium ..., Norwood, N.J. Ablex Publishing Corporation.