Handling uncertainty in multimodal pervasive computing applications
Marie-Luce Bourguet *

Computer Science Department, Queen Mary, University of London, Mile End Road, London E1 4NS, UK

Abstract
Multimodal interaction can improve accessibility to pervasive computing applications. However, recognition-based interaction techniques used in multimodal interfaces (e.g. speech and gesture recognition) are still error prone. Recognition errors and misinterpretations can compromise the security, robustness, and efficiency of pervasive computing applications. In this paper we briefly review the various error handling strategies that can be found in the multimodal literature. We then discuss the new challenges arising from novel affective and context-aware applications for error correction. We show that traditional multimodal error handling strategies are ill adapted to pervasive computing applications, where the computing devices become invisible, and when users may not be aware of their own behaviour. Finally, we present an original experimental study into users’ synchronisation of speech and pen inputs in error correction. The results of the study suggest that users are likely to modify their synchronisation patterns in the belief that it can help error resolution. This study is a first step towards a better understanding of spontaneous user strategies for error correction in multimodal interfaces and pervasive environments.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction
Multimodal interfaces are a class of multimedia systems that implement multiple recognition-based modalities of interaction. Early multimodal systems were based on the recognition of active modes, such as speech and handwriting, for which there is now a large body of research work. However, the emergence of novel pervasive computing applications, which combine active interaction modes with passive modality channels based on perception, context, environment and ambience [1,9,22], raises new challenges. For example, context-aware systems can sense and incorporate data about lightning, noise level, location, time, people other than the user, as well as many other pieces of information to adjust their model of the user’s environment. More robust interaction is then obtained by fusing explicit user inputs (the active modes) and implicit contextual information (the passive modes). In affective computing, sensors that can capture data about the user’s physical state or behaviour, are used to gather cues which can help the system perceive users’ emotions [15,28].

Despite recent advances in computer vision techniques and multi-sensors systems, few successful multimodal interfaces have been implemented. This is partly due to the difficulty in designing and implementing such interfaces. Designing and implementing systems that take the best advantage of recognition-based modalities of interaction and multi-sensory observations is difficult. Our lack of understanding of how these technologies can be best used and combined in the user interface often leads to interface designs with poor usability and low robustness. Moreover, even in more traditional multimodal interfaces (e.g. speech and pen interfaces), technical issues remain. Natural modalities of interaction, such as speech and gestures, rely on recognition-based technologies, which are inherently error prone. Speech recognition systems, for example, are sensitive to vocabulary size, quality of audio signals and variability of voice parameters. Signal and noise separation also remains a major challenge in speech recognition technology, as current systems are extremely sensitive to background noise and to the presence of more than one speaker.

Recognition-based multimodal interaction is still error prone, but the possibilities of errors and misinterpretations in pervasive computing applications, where the capture and the analysis of passive modes are key, are even greater. Not only the computing devices have become invisible, but the users may not be aware of their behaviour that is captured by the system. They may also have a wrong understanding of what data is captured by the various devices, and how it is used. In most cases, they do not receive any feedback about the system’s status and beliefs. For example, in affective computing, if the system wrongly concludes from the analysis of a user’s facial expression, that he or she is sad or anxious, it may embark on trying to comfort him/her by accordingly changing its behaviour and response mode. In this situation, the user is faced with a number of challenges: understanding the computer’s change of behaviour (e.g. “The system is trying to comfort me”); analysing the cause of the system’s changed behaviour
(e.g. “The system believes I am in a sad mood”); and devising ways to correct the system’s wrong belief (e.g. “I should smile”). Traditional methods of multimodal error correction are ill adapted to pervasive computing applications and research is urgently needed to better understand user behaviour when faced with errors in this type of application.

In the next section of the paper we briefly review the various recognition error handling strategies that can be found in the multimodal interaction literature. In Section 3, we show that many multimodal error handling strategies, where active modes only are used, are ill adapted to pervasive computing, and we discuss the new challenges arising from the deployment of novel pervasive computing applications for error correction. Finally, in Section 4, we present an experimental study into users’ synchronisation of speech and pen inputs in error correction. The results of the study suggest that, given that the source of the error can be identified, users are likely to modify their synchronisation patterns in the belief that it can help error resolution. This study is a first step towards a better understanding of spontaneous user strategies for error correction in multimodal interfaces and pervasive environments.

2. Multimodal error handling strategies

Every study of recognition-based human–computer interfaces shows that recognition errors reduce the effectiveness of natural input modalities such as speech and handwriting [11, 16, 31]. It has also been observed that the impact of recognition errors depends upon a number of factors such as the amount of input required, the acceptability of uncorrected errors, the benefits of using recognition-based modalities as compared with other interaction means, and the availability of adequate error handling mechanisms [7]. Error handling strategies can be classified according to two variables: actor and purpose (Fig. 1). According to these two variables, six categories of error handling strategies have been defined: (1) error prevention by machine (which includes error reduction by design and error reduction by context), (2) error discovery by machines (i.e. automatic detection), (3) error correction by machine, (4) error prevention by users, (5) error discovery by users (i.e. machine-led discovery), and (6) error correction by users.

Most strategies for error prevention can be attributed to the machine. They work in two possible ways: either the interface is designed to influence or constrain user behaviour into less error-prone interaction (i.e. “error reduction by design”), or greater recognition accuracy is achieved through the use of additional or contextual information (i.e. “error reduction by context”). Error reduction by design techniques achieve error prevention by leading users towards the production of inputs that are easier to recognise. The different techniques differ in the level of constraints they impose on user behaviour and actions, and the degree of control the user has on the interaction. For example, Tap-to-speak interfaces are interfaces in which users must indicate to the system by a brief signal that they are going to talk before each utterance [23]. Another technique consists in implementing guided dialogues where users are prompted to say or do something from a limited set of possible responses. Another, less constraining technique, consists in controlling the system’s responses and discourse level throughout the dialogue to shape the users’ speech and actions to match that of the system’s [12]. Error reduction by context techniques achieve error reduction by augmenting user inputs with redundant or contextual information. Machine lip reading, for example, consists in combining acoustic information from the speech signal with visual information from the shapes of the speaker’s lips to achieve more robust speech recognition [20].

Error discovery by machine works in three possible ways: by using statistical data, by exploiting cross-modal information, or by applying knowledge-based rules [2]. Rules may be based on semantic, pragmatic or common sense knowledge [29]. With knowledge-based and cross-modal strategies, the automatic discovery of recognition errors can sometimes lead to automatic correction as well. This is generally true if the correct output figures in

![Fig. 1. Taxonomy of error-handling strategies based on actor and purpose (reproduced from [7]).](image-url)
the list of alternative hypotheses produced during the recognition process.

On the users side, user prevention strategies rely on users’ spontaneous change of behaviour to prevent errors. This is facilitated in natural multimodal interfaces by the availability of multiple modalities of interaction, which allows users to exercise their natural intelligence about when and how to deploy input modalities effectively [26]. When a recognition error occurs, users are normally in charge of notifying the machine. It is important, however, that the machine facilitates error discovery. Machine-led discovery techniques include implicit confirmation [21], explicit confirmation (e.g. when in safety critical systems users are asked to confirm that what has been recognised or understood is correct), visually displaying recognition results, and allowing the selection of the correct result from a list of alternative hypotheses. Once errors have been found, users can effectively help the machine resolve them, usually by producing additional inputs. Studies of speech interfaces have found that the most instinctive way for users to correct mistakes is to repeat [32]. However, although repeating might be the most obvious way to correct when the system mis-hears, it is often the worse for the system [10]. The main reason for this is that when repeating, users tend to adjust their way of speaking (e.g. by over-articulating) to what they believe is easier for the recogniser to interpret, which often has the opposite effect. In handwriting, a similar strategy to repeating is to overwrite a misrecognised word. Linguistic adaptation is another strategy that has been observed where users choose to rephrase their speech, in the belief that it can influence error resolution: a word may be substituted for another, or a simpler syntactic structure may be chosen [27]. In multimodal systems, it has been suggested that users are willing to repeat their input at least once, after which they will tend to switch to another modality [24]. For example, if speech input failed repeatedly when entering data in a form, users may switch to the keyboard in order to type their entry. Alternative strategies include locating a recognition error by touching a misrecognised word on a writing-sensitive screen where recognition output is displayed, then correcting the error by choosing from a list of alternative words, typing, handwriting, or editing using gestures drawn on the display [31].

As shown above, an extensive body of work exists in multimodal error handling and a large number of error handling strategies have been proposed and tried, however, most of these strategies assume the use of active modalities. In the next section of the paper, we show that many of these strategies are ill adapted to pervasive computing applications, where passive modalities play an important role.

3. Error handling in pervasive computing

It was observed in the previous section that there are two main actors in the process of handling recognition errors: the machine and the user. On the machine side, error reduction by design is a major error handling strategy in “traditional” multimodal interfaces. It aims at preventing interaction errors by influencing or guiding users’ behaviour. In pervasive computing, where the devices are invisible and must not interfere with social interaction and human behaviour, error reduction by design goes against the fundamental principles. Indeed, pervasive computing is about anticipating, not influencing users needs and actions. In effect, a whole class of error handling strategies becomes inapplicable. In [19] for example, the authors present an authentication model, which addresses the problem of authenticating mobile users without interfering with their mobile behaviour. They describe a “zero-stop” authentication model, which allows them to actively authenticate users in an environment populated with various mobile and embedded devices without disturbing users’ movements. The model they propose determines the timing constraint needed to realise “zero-stop” property from the speed of the users, size of the sensing area, and the overhead of sensing and authentication process. This work provides us with a good example of a pervasive computing application, which operates under the strict constraint of zero disturbance.

In contrast, error reduction by context strategies are likely to play an important role in the prevention of errors by machine in pervasive computing. The definitions of interaction contexts and the handling and integration of multi-sensors information are currently the subjects of intensive research. The use of context and multi-sensors information will not only render the interaction richer and more natural, it also has the potential to make it more robust and less uncertain. In [30], for example, is presented a prototype system that provides an infrastructure for leveraging the strengths of different sensors and processes used for the interpretation of their collective data. To be effective at disambiguating multimodal interaction, context and multi-sensors information must be complementary to users inputs, but not necessarily semantically significant. In [4], for example, timing information from hand gestures is used to locate in the speech signal the parts that are more semantically significant. Another good example of this consists in using information captured by multiple microphones or cameras about the position of a speaker in a room to help managing the speech recognition process (for example [18]).

In current multimodal interfaces, the automatic correction of recognition errors is achieved using semantic, pragmatic, and common sense knowledge. For example, the Open Mind Common Sense Project [29] has collected common sense statements from the public since the fall of 2000, resulting in a database that currently contains more than 700,000 facts. The common sense statements have been used to reorder the recognition hypotheses returned by a speech recogniser and filter out possibilities that “don’t make sense”. The researchers found that their common sense speech recognition technique was particularly efficient at disambiguating between words that are phonetically identical. In pervasive computing, the approach can be taken further by incorporating knowledge about human behavioural and social signaling. As the field matures, such knowledge will undoubtedly become invaluable to allow machines to automatically detect and correct recognition errors. For example, the understanding of users emotions through the analysis of facial expressions (affective computing) will allow machines to disambiguate between literal and ironic statements. In [13] a series of experiments are presented, which explore the extent to which iconic gestures convey information not found in speech. The results suggest that listeners can use gestural information to disambiguate speech. For example, an iconic gesture can facilitate the processing of a lesser frequent word meaning.

For error prevention, discovery, and correction by users, the invisibility of the devices raises one of the most important challenges. As the devices responsible for capturing and analysing interaction data become invisible, it will become increasingly difficult for users to identify the causes of recognition errors. Although users have a prime role in finding errors, the machine has traditionally been in charge of enabling error discovery by providing adequate feedback on its status and beliefs (machine-led discovery), and so in pervasive computing, the challenge is to devise ways of providing the necessary feedback while remaining invisible and unobtrusive. As the experimental study presented in the next section shows, the users’ ability to devise error handling strategies is dependent on the availability of system’s feedback about its current status and beliefs.

However, the invisibility of the devices is only one of the properties of pervasive computing applications that raise new chal-
Fig. 2. Device and data properties in pervasive computing.

<table>
<thead>
<tr>
<th>Devices</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invisibility</td>
<td>What is captured?</td>
</tr>
<tr>
<td>Multiplicity</td>
<td>What data</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>Use</td>
</tr>
<tr>
<td>Disparity</td>
<td>Trustworthiness</td>
</tr>
<tr>
<td></td>
<td>Accuracy</td>
</tr>
<tr>
<td></td>
<td>Combination</td>
</tr>
</tbody>
</table>

Fig. 2 summarises the characteristics of the devices and captured data in pervasive computing that are likely to make error handling by users difficult and render typical error handling strategies impractical.

In pervasive computing, not only the computing devices have become invisible ("invisibility"), but also users may not even be aware of their own actions, which have been captured and exploited by the system to enhance the interaction ("what data"). The use of passive modalities, for example through the capture of spontaneous gestures and facial expressions, is an important property of pervasive environments. However, the change from an environment where the user is always the conscious actor of every input received by the system, to an environment where the user is only one possible source of inputs among others, or where the inputs produced by the user are produced unconsciously, is a dramatic one. In [4] a driving assistant system is described that relies on passive modalities only (facial expression, head movement and eye tracking) to capture the driver's focus of attention and predict their fatigue state. The driver's face is monitored with a video camera and three signs of hypo-vigilance are tracked: yawning, blinking (or eyes closure) and head motion. Complex bio-inspired algorithms are then used in order to analyse the data and predict attention and fatigue. This system explicitly relies on the fact that users have no or little control on the data captured, in order to detect human behaviours, which may be dangerous in driving conditions. However, there is no consideration given to the possibilities that the system's predictions may sometime be wrong (system error) and that the user will have no means to correct the situation.

When data has been wrongly interpreted in a pervasive environment, it may be impossible to know which device is responsible ("multiplicity"), and what combination of data contributed to the wrong interpretation ("combination"). Even when users are aware of what data is captured, for example images of their face, it may not be clear how the data is used by the system ("use"), if it is used for the right purpose ("trustworthiness"), and how accurate is the data ("accuracy"). When it comes to try and influence system's behaviour and beliefs, it will be necessary to understand how sensitive the devices are ("sensitivity"), and how disparate or homogeneous they are in their properties and characteristics ("disparity").

A wrong knowledge or mental model of any of the device and data properties in a pervasive computing application will make error handling particularly difficult for users. Some recent work has started to highlight the importance of a user-centered approach to the design of pervasive computing applications. In [17] in particular, an approach of a usage model for specifying each context of services used for data capture, of the nature of the captured data, and of the use that is made of it. In the next section of the paper, we present an experimental study that aims to test users spontaneous change of behaviour in situation of error correction, when the cause and source of the error cannot easily be identified. The study was designed to verify if users are likely to modify some aspects of their input when repeating a complex multimodal command (e.g. a command that combines speech and gestures), in the belief that it can help error resolution. In particular, we are interested in comparing users modality synchronisation patterns in normal situations of interaction, and in situations of error correction.

4. Multimodal behaviour in error correction: experimental study

4.1. Input synchronisation patterns

Input synchronisation patterns are important to devise accurate and efficient modality integration techniques. For example, synchronisation patterns will allow a multimodal system to differentiate between independent inputs ("concurrent multimodality") and synergistic inputs ("compound multimodality"), and to set appropriate timeouts between subsequent synergistic inputs. Previous work in the area of multimodal user behaviour has uncovered typical patterns of natural integration and synchronisation of input modes [4,25]. In [25] for example, integration patterns for speech and pen inputs are described, in which pen onset usually precedes speech onset in sequential constructions (when one input is completed before the onset of the other) as well as in simultaneous constructions (when there is a temporal overlap between inputs in different modalities). In [4], it is shown that 3D hand pointing gestures are usually synchronised with either the nominal or deictic expression of a phrase. It is also shown that the timing of such gestures is predictable in the [–200 ms, 400 ms] interval around the beginning of
their related expressions. However, as far as we are aware, no previous work has been reported on modality integration patterns in situations of error correction. The aim of the experiment we present here is to study modality integration patterns when users are faced with system error, which they cannot easily analyse, and they are constrained to repeat their multimodal command. The results of this experimental study were originally presented at the International Workshop on Multimodal Corpora: From Multimodal Behaviour Theories to Usable Models, Genoa, in May 2006 [6].

4.2. Task

We implemented a multimodal speech and pen application on a graphic tablet for which the participants in the experiment were asked to place famous London landmarks on a digital map. They had to reproduce a map similar to a model provided on paper (see Fig. 3), where images of specific sizes had to be positioned in specific places. On the model, the names of the landmarks (e.g. "tower bridge") were written on each picture and the participants were instructed to use these names when speaking to the system. The task included: making the images appear on the empty map, resizing the images, and positioning the images as precisely as possible on the map. This task was chosen because it includes both verbal (naming the landmarks) and spatial (positioning the images) elements, for which multimodal interaction is highly appropriate.

To accomplish the task, four different interaction styles were implemented:

- speech–gesture commands (e.g. to make a picture appear on the map, the user draws a “P” and pronounces the name of the landmark);
- speech–point commands (e.g. to resize a picture, the user points at a picture and says “smaller”);
- speech–point–point commands (e.g. to swap two images, the user says “swap” and points at two different images);
- speech–draw commands (e.g. to move an image, the user says “move it here” and draws a line from the image to its desired location).

The vocabulary of the London map application is simple and does not allow linguistic adaptation (i.e. rephrasing). It comprises approximately 30 entries, including the names of the landmarks and command words or expressions such as “smaller”, “move it here”, etc.

Although the set of multimodal commands and vocabulary is fixed, inputs can be entered in any order (e.g. pen first, speech first or pen and speech overlapping).

The application was built using the multimodal toolkit described in [5]. Speech recognition was carried out using the ViaVoiceTM engine and gesture recognition performed via the Satin toolbox [14] on a Wacom LCD tablet.

4.3. Method

Eight paid subjects, with no experience with recognition-based systems, participated in the experiment. All of them received a brief demonstration of the application during which the experimenter was careful to show a range of different synchronisation patterns in order not to influence the execution of the multimodal commands. The participants also had the opportunity to undertake a short training session to familiarise themselves with the task and the different interaction styles.

No feedback was provided on the activity and performance of the speech and gesture recognisers. This means that, in case of interaction problems, users could only realise that an error had occurred when evaluating the response of the system. For example, if
nothing appears on the screen after a “draw image” command, users can only assume that either a speech or a gesture recognition error has occurred. Not being able to precisely determine where it went wrong was believed to be an incentive to repeat the entire command.

The experiment would only end when the entire task had been completed, i.e. when the produced map was similar to the model provided, with all the landmarks of the correct size and roughly positioned in the right place. This provided another necessary incentive to try and correct recognition errors.

This simple application does not make use of passive modalities and so, although it is multimodal, is not representative of a pervasive computing environment. However, it serves well our purpose of testing users spontaneous behaviour in error correction when the cause and source of the error cannot easily be identified, which is typical of a pervasive environment.

4.4. Data collection

During the experiments, automatic logs were set up to record various data: timing of every pen-down and pen-up event, speech onset and offset, and speech and gesture recognition results. The experiments were also videotaped. The log files were then compared with the video recordings in order to identify the situations of recognition error recovery. We were only interested in users’ strategies to cope with errors made by the recognition systems, so ungrammatical or out of vocabulary user inputs, i.e. inputs where the participants, as opposed to the recognition systems, had made an error were discarded.

Four types of recognition errors were observed: speech false rejection, speech misrecognition, gesture false rejection, and gesture misrecognition. For speech–gesture commands, combinations of speech and gesture recognition errors were also possible. To illustrate the different recognition errors, let us imagine that the user said “Tower Bridge” while drawing a “P” gesture on the screen. The possible recognition errors are:

- [-] User’s speech has not been recognised (speech false rejection): nothing appears on the screen.
- [-] User’s speech has been misrecognised (speech misrecognition). If the system recognised the name of another landmark, an unexpected image appears on the screen.
- [-] User’s gesture has not been recognised (gesture false rejection): nothing appears on the screen.
- [-] User’s gesture has been misrecognised (gesture misrecognition). If the system recognised a “delete” gesture and the gesture was drawn on an image, the image unexpectedly disappears. If the gesture was drawn on an empty space of the screen, nothing happens.

User inputs were then classified into one of the two following categories:

- [-] New commands: when a command is entered in normal situation of Interaction.
- [-] Recovery commands: when a command is repeated, in response to a recognition error. If the user corrects an unexpected result (such as deleting an unexpected image) before repeating the initial command, the repeated command is not considered a recovery command, but a new command.

4.5. Results

A total of 1073 multimodal commands were collected, of which 279 were entered in situations of error recovery. Fig. 4 summarises the most commonly observed synchronisation patterns for the four interaction styles. In each case, the following information is shown: (1) total number of commands observed; (2) average pattern (the top line represents speech and the bottom line pen; the lines are proportional to event durations); and (3) proportions of the two most frequently observed patterns. For example, 133 speech–gesture commands were collected in a normal situation (new commands). For these commands, the average pattern is characterised by pen onset first, followed by speech. Speech onset occurs approximately in the middle of the gesture execution and finishes after the gesture has been completed. Seventy nine percent of speech–gesture new commands conform to this typical pattern. Nineteen percent of speech–gesture new commands conform to a different pattern where speech onset precedes pen onset and where the gesture finishes before the end of the speech.

At first glance, it can be seen that across the data (independently of the interaction style and of the command category), pen onset tends to precede speech onset. This result corroborates
the main finding reported in [25] and is valid across the different interaction styles, in both normal and error recovery modes.

For the speech–gesture commands, users cannot easily determine which recogniser has made an error because the multimodal user interface does not provide any feedback on the activity of the recognisers and on the recognition results they return. Any of the two recognisers (speech and gesture) could be the source of the error (“multiplicity” problem), or the interpretation of the combined inputs could be wrong (“use” and “combination” problems). We observe in Fig. 4 that the integration patterns in normal and recovery modes for this type of commands are similar. In recovery mode, the participants tend to repeat their commands in exactly the same manner as in the normal mode of interaction. This can be explained by the fact that, when a command is unsuccessful, both modalities of interaction, speech and gesture, can be held responsible for the error. In these circumstances, users do not seem to be able to devise a strategy for error recovery.

However, in the speech–point, speech–point–point, and speech–draw cases, where speech is the only input that is subject to recognition errors, some differences can be observed between the new commands and the recovery commands.

For the speech–point and speech–draw commands, the data suggest that speech onset is shifted towards the beginning of the pen stroke in situations of error recovery. For speech–draw commands, a significant proportion of error recovery inputs (37% compared to 24% for normal inputs) shows in fact a precedence of speech over pen. In this case, it seems that users tend to deal first with the error–progn input (speech).

For the speech–point–point commands, the proportions of the two frequently observed patterns are reversed. In recovery mode, it seems that users are more likely to have completed their pen inputs before speech onset. This change of behaviour can be attributed to the complexity of the interaction style. In recovery mode, users tend to avoid multi-tasking by adopting sequential patterns of integration, where there is no temporal overlap between inputs in different modes.

4.6. Conclusion

Small and mobile computing devices, in the form of personal digital assistants, mobile phones, and wearable computers, are common components of pervasive computing applications; but compared with desktop computers, their screens are small or non-existent, and their small keyboards are hard to use when on the move. On these platforms, typical multimodal error handling strategies such as linguistic adaptation, modality switch, and lists of alternative words, may not be available, leaving repetition as the only viable strategy. The experimental study has shown that, when repeating a multimodal command, and given that the source of the error can be identified, different modality synchronisation patterns are likely to enter in users’ strategies for influencing the performance of recognition-based modalities. Synchronisation patterns that significantly depart from typical patterns should be interpreted with in view the possibility that the user is in error recovery mode, and modality integration techniques should be able to adapt to changing synchronisation patterns. However, users only seem to be able to adapt their behaviour when they can identify the source and nature of the system error.

5. Discussion

With the increasing diversity of devices, contexts of use, and users, the design of effective means of error prevention, detection, and correction will be a determinant factor of usability and users’ acceptance of pervasive computing applications. The experimental study presented above has highlighted the necessity of providing adequate support to allow the deployment of (often opportunistic) user strategies for handling errors. However, error handling in pervasive computing applications promises to be more complex than in current multimodal interfaces. In pervasive computing, it will be of paramount importance that users are supported in their forming of adequate mental models of the system. These mental models should provide users with the correct knowledge of what data is captured and recorded, and how it is used. Because of the invisibility of the devices and the necessity of being unobtrusive, supporting the development of adequate user mental models is more challenging than in traditional interfaces. Provided that the pervasive computing application successfully promotes adequate mental models, it can be anticipated that users will develop whole new strategies to cope with errors in pervasive computing applications. However, research to gain a better understanding of these strategies will be needed in order to devise appropriate interface designs and techniques to support them.

Acknowledgments

This research was supported in part by the Nuffield Foundation under grant NUF-NAL 00. Many thanks to Sarah Talbot for running the experimental study and to all the participants.

References


