Towards a Taxonomy of Error-Handling Strategies in Recognition-Based Multimodal Human-Computer Interfaces

Dr. Marie-Luce Bourguet
Computer Science Department, Queen Mary, University of London
Mile End Road, London, E1 4NS, U.K.
mlb@dcs.qmul.ac.uk
Tel: +44 (0)20 7882 5200
Fax: +44 (0)20 8980 6533

Abstract

In this paper, we survey the different types of error-handling strategies that have been described in the literature on recognition-based human-computer interfaces. A wide range of strategies can be found in spoken human-machine dialogues, handwriting systems, and multimodal natural interfaces. We then propose a taxonomy for classifying error-handling strategies that has the following three dimensions: the main actor in the error-handling process (machine versus user), the purpose of the strategy (error prevention, discovery, or correction), and the use of different modalities of interaction. The requirements that different error-handling strategies have on different sets of interaction modalities are also discussed. The main aim of this work is to establish a classification that can serve as a tool for understanding how to develop more efficient and more robust multimodal human-machine interfaces.

Keywords: recognition-based technology, multimodal interfaces, error-handling, taxonomy, interaction design, interaction robustness

1. Recognition-based technology

Multimodal interaction refers to interaction with the virtual and physical environment through natural modes of communication such as speech, body gestures, handwriting, graphics, or gaze. Unlike keyboards and mice inputs, natural modes of communication usually are non-deterministic, and have to be "recognised" by a recognition system, before they can be passed on to an application. Recent developments in recognition-based technology (e.g. speech and gesture recognition) have opened a myriad of new possibilities for the design and implementation of multimodal applications. Handwriting recognisers, for example, are being used in personal digital assistants (e.g. Paragon's multilingual PenReader software for Pocket PC devices), and speech recognition has made its way into desktop machines (e.g. IBM's ViaVoice™ speech recognition engines). However, designing and implementing systems that take the best advantage of these new interaction techniques is difficult. Our lack of understanding of how recognition-based technologies can be best used and combined in the user interface often leads to interface designs with poor usability. In particular, users' expectations of a system's capabilities and users' mental models of how a multimodal interface works are often inadequate, and these trigger interaction problems. Another source of errors comes from the ambiguous nature of many recognition-based users' inputs (inputs that can be understood in two or more possible ways). This is especially true of speech, as the expressive power of spoken language is often obtained at the price of some ambiguity and imprecision in the messages. Moreover, recognition-based technologies are still error-prone. Speech recognisers, for example, are not perfect listeners. They make mistakes.

Speech recognition accuracy is usually calculated as the number of words correctly recognized, as a percentage of the number of words spoken. The technology has made much progress in the last several decades and recognition accuracy rates as high as 98% have been reported [1][2]. However, the performance of today's speech recognition systems is still dependent on a number of factors, such as vocabulary size, quality of audio signal, and variability of voice parameters. Background noise and small changes in voice characteristics, due, for example, to the speaker having a cold, can significantly affect the performance of a recogniser even after the user has trained it. Besides, a linear relationship has been identified between a user's level of physical exertion and recognition accuracy; the higher the level of exertion, the lower the accuracy rate [3]. Finally, current speech recognition systems have been designed to process read text or speech that does not exhibit disfluencies, such as pauses, hesitations, repetitions and repairs. For spontaneous and disfluent speech, recognition accuracy decreases drastically [4].

Handwriting recognition systems are not perfect either. The causes of the recognition errors that are likely to occur in handwriting interfaces include: discrete noises, badly formed shapes, incorrectly spelt words, cancelled material, and device-generated errors [5]. In 1999, MacKenzie and Chang [6] compared two handwriting recognisers by asking users to copy words that were presented to them and by logging entry speed and accuracy. Accuracies of between 87% and 93% were reported. In 2001, a study by Read et al [7] measured accuracy, speed, and user satisfaction with handwriting recognition, using children as subjects. Recognition rates averaging 86% were reported for unconstrained text entry. Higher performance numbers have been achieved in recent years [8], however, all recognition performance numbers are dependent on the particular test set and should be taken with a pinch of salt [9]. This is because, in order to correctly interpret these statistics, one need to know the experimental conditions that prevailed in the studies.

Every study of recognition-based human-computer interfaces shows that recognition errors reduce the effectiveness of natural input modalities such as speech and handwriting [10][11][12]. Typical text entry rates of 25 to 30 words per minutes (wpm) via speech recognition systems with recognition accuracy as high as 94%, are much slower than voice dictation speeds, which are generally around 150 wpm [13] [14]. The primary reason for this slower speed is the need to correct recognition errors. But recognition accuracy and speed are not the only determinant of user acceptance of recognition-based technology [15]. In speech systems, error rates have been found to correlate only loosely with satisfaction [16]. According to [17], higher speech recognition accuracy tends to be associated with better general performance (e.g. higher text entry rate), but the correlation, at 0.62, is not as strong as might be expected. In fact, users who have the best performance tend to be those who employ the best correction strategies. Similarly, Frankish et al [18] have studied the relationship between recognition accuracy and user acceptance of pen interfaces, and have shown that it is highly task dependent. For example, it was found that users were less frustrated by handwriting recognition errors when the task was to enter a command in a form than when they were writing journal entries. In general, the impact of recognition errors depends upon a number of factors such as the amount of input required, the acceptability of uncorrected recognition errors, the benefits of using speech or handwriting as compared with other interaction means, and the availability of adequate error recovery mechanisms.

A wide range of strategies for handling recognition errors can be found in the literature on spoken human-machine dialogues, handwriting systems, and multimodal natural interfaces. In the next section of the paper, we survey the different types of error-handling strategies that have been proposed. We then present in section 3 a taxonomy for classifying error-handling strategies, which is based on three dimensions: the main actor in the error-handling process (machine versus user), the purpose of the strategy (error prevention, discovery, or correction), and the use of different modalities of interaction. A discussion on the requirements different error-handling strategies have on different sets of interaction modalities follows in section 4 of the paper. The main aim of this work is to establish a classification of error-handling strategies that can serve as a tool for understanding how to develop more efficient and natural multimodal human-machine interfaces.

2. Error-handling strategies in recognition-based interfaces

Mankoff et al [19] suggest four key research areas for error-handling in recognition-based interfaces: error reduction, error discovery, error correction techniques and toolkit-level support. According to Mankoff, error reduction involves research into improving recognition technology in order to eliminate or reduce errors. We have found in the literature a variety of other techniques to achieve error reduction, such as better interface design and the use of contextual information.

In the area of error discovery, Mankoff mentions two categories of strategies: explicit users' input to tell the system that an error exists, and the system's output of uncertain interpretations of recognised input to make the user aware of potential errors. A number of other strategies have in fact been proposed, for example, the mutual disambiguation of input modes in natural multimodal interfaces.

In this section, we present error-handling strategies using the following six categories: *error reduction by design* (constraining or influencing users' actions, expectations, and mental models to prevent errors), *error reduction by context* (augmenting input signals with contextual information to improve recognition), *user prevention* (users' spontaneous behavioural changes to prevent errors), *automatic detection* (techniques to automatically detect potential recognition errors), *machine-led discovery* (techniques to facilitate error discovery by users), and *user correction* (strategies to allow users to resolve recognition errors).

2.1. Error reduction by design

A number of different techniques have been proposed whose main goal is to prevent or reduce recognition errors. "Error reduction by design" describes techniques that achieve error reduction by leading users towards the production of inputs that are easier to recognise by machines. 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. We describe them here, starting from the most constraining methods.

In recognition-based technology, there are two basic problems that recognisers must solve: segmentation and recognition. In handwriting, the input to a recogniser is a series of ink strokes stored as sets of digitised points. Segmentation is the process of determining which segments are in which characters [8]. One way to reduce the complexity of segmentation is to constrain input, for example, by supporting block printed characters only, or non-connected cursive handwriting only. On-screen boxes are sometimes used to force users to write **isolated characters** of uniform size, and discourage connected cursive handwriting.
In a 1993 paper, at a time when handwriting recognition technology was considered a failure, Goldberg and Richardson at the Xerox Palo Alto Research Center [20] introduced the Unistrokes alphabet. Unistrokes is a stylised **single-stroke alphabet** that is both easier for software to recognise (because there is no possibility of segmentation errors), and quicker for users to write. The public never adopted Unistrokes, probably because it is difficult to learn, but in 1996, Palm Inc. introduced another single-stroke system called *Graffiti* that proved much more successful. *Graffiti* has been credited as a significant reason for the commercial success of the *Palm* [8]. The great advantage that *Graffiti* has over *Unistrokes* is its similarity to normal hand-printed characters (see figure 1). Recognition accuracy rates as high as 97% have been reported in [21] for inputs with Graffiti.

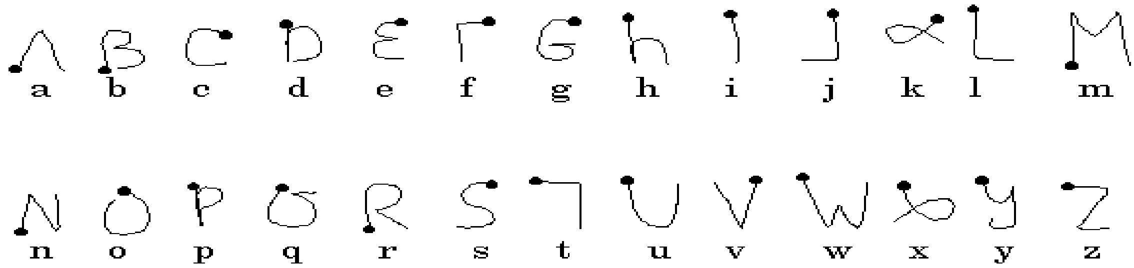

Figure 1. The graffiti alphabet (reproduced from [8]).

Currently, all commercial products that support pen-based text input work with constraints or stylised alphabets. But for speech, isolated word recognisers (recognisers that can only take one word at a time) are no longer deemed acceptable, and the general expectation is for continuous speech recognition. Nevertheless, other types of constraints have been tried, in attempts to simplify the task of speech recognisers. **Tap-to-speak interfaces** are interfaces in which users must indicate to the system by a brief pen gesture or a mouse click that they are going to talk before each utterance [22]. It is reported in [23] that in such interfaces, user speech is significantly more intelligible than during open-microphone interaction, resulting in better recognition accuracy. No study could be found in the literature on user acceptance of tap-to-speak interaction, and on the effect it has on the naturalness of user multimodal behaviour patterns. Although potentially efficient, this method seems to be rarely used.

By contrast, the use of language grammars has generalised to most continuous speech recognition systems. Rule-based grammars are used to limit the range of possible inputs that a recogniser will find acceptable and hence limit its research space and the risk of recognition errors. Many factors must be considered when designing a grammar, but the most important is assessing the trade-off between flexibility and performance. Obviously, the more constraints are imposed on what users can say to an application in a **restricted grammar**, the less likely they are to encounter recognition errors, but this should not be obtained at the price of forcing users to memorise commands. A badly designed grammar or a grammar that imposes too many constraints on user speech can be the trigger for more errors, as user speech, especially beginners' speech, will often not conform to the pre-defined grammar (this is dubbed "the vocabulary problem" in [24]). The Wizard of Oz technique presents a useful methodology to elicit user vocabulary and natural expressions for building appropriate grammars and testing dialogue models [25]. It consists in making users believe that they are speaking with a fully implemented spoken dialogue system, whereas a "human wizard" responds to their spoken inputs, and while their speech is being logged in order to build a corpus of valid user expressions.

The structure of the dialogue normally determines who has the control of the interaction between the user and the computer and determines how information is presented and received. A dialogue may vary in this degree of control from user initiated (where users are in command), to computer initiated (i.e. **guided dialogues** where users simply respond to a request of the system) [26]. An advantage of guided dialogues is that they can prompt the user to say something from a limited set of possible responses [27]. When a graphical display is available, 'Form Filling' dialogues can be implemented, where data is entered onto screens that resemble a form. It is reported in [22] that a structured form-based interaction can reduce speech disfluencies to just 30-40% of those that would have occurred in an unconstrained interface, i.e. a free form interface that requires more self-structure and planning from users.

In [28] a speech-enabled VRML browser has been developed for the study of different speech errors that occur in 3D worlds. It introduces **context-sensitive cues** as a solution to support a system's metaphor. Context-sensitive cues can be implemented in a variety of ways, the simplest being to express the names of the commands as labels and put them into the scene around the user's current position. In this study, context-sensitive cues are designed as hints to express what actions the possible commands could produce, in order to prevent user semantic errors. However, the technique can also be used for explicitly (i.e. visually) presenting valid spoken commands to the user, incurring less out-of-vocabulary input and speech recognition errors. It presents the added benefit of enabling the safe use of "speech modes". Speech modes refer to specific dialogue states, for which only sub-sets of the recogniser's grammar are active, effectively reducing the recogniser's search space and improving recognition accuracy. Speech modes are inherently dangerous to use (suddenly, words that seemed to work don't work anymore), unless users are well aware of them and are always able to recognise in which mode the dialogue is operating. This can be facilitated with the use of context-sensitive cues placed in the user's visual field.

Another approach consists in using techniques that facilitate human adaptation to the language of the system [29]. The goal is to "get people to say and type what computers can understand" [30]. For example, controlling the system vocabulary and discourse level throughout the dialogue will shape a user's speech to match that of the system's grammar. Well-designed and consistent prompts are keys to achieve the desired effect [16] [31]. Two fundamental guiding principles have been identified in [32]: **consistency and symmetry**. *Presentation consistency* refers to the similar structure and format of prompts presented to the user. The principle of *symmetry* suggests that users respond to prompts in the same style and wording used by the prompt. The symmetry principle constitutes one of the most powerful and least intrusive means to achieve the goal of getting people to say what the system can understand. Recently, research has started on two-way adaptation frameworks whereby both users and systems dynamically adapt to each other's capability and needs during the course of interaction. In addition to helping users to dynamically learn the system's capabilities in context, the approach aims to enhance the overall interpretation capability of a system by learning new user expressions on the fly [33].

In multimodal graphic interfaces, one effective interface technique for reducing spoken disfluencies, and as a consequence reducing recognition errors, is to implement highly **structured graphics** [34]**.** It has been demonstrated in [35] that increased utterance length is a powerful predictor of spoken disfluencies and that structured graphic interfaces can successfully elicit brief spoken utterances, reducing difficult sources of linguistic variability in human speech and writing by a factor of 2 to 8 fold. For complex spatial domains, such as map-based interaction where disfluencies are concentrated more heavily on spatial location constituents, a highly structured map has been shown to significantly reduce disfluencies [34]. In this context, a structured map may be a detailed reference map that displays the full network of roadways, buildings, and labels that are conventionally found on hard-copy reference maps; whereas, a less structured map would be a minimalist version of the reference map that displays only a streamlined network of roads.
Finally, spoken disfluencies can also be reduced by designing multimodal interfaces that include another **adapted modality**, which can be used complementarily or redundantly with speech [36]. According to [34], for map-based interactions, pen input can be used more effectively than speech to indicate points, lines, and abstract areas, and contribute to reduce spoken disfluencies to just 50% of the rate that would have occurred for the same speaker completing the same task in a speech-only interface. In particular, the speech modality is well suited for specifying commands (“calculate the distance”) and degrees (“at least a mile away”), but less so for articulating spatially-oriented descriptions (e.g. “ the handicapped center that is east of Deer Creek School”) [34]. In this case, a pen is more adequate as the lengthy spoken input can be easily replaced by a simple manual selection. More generally, an adapted modality is a modality that is best suited for a particular type of content (e.g. verbal versus spatial content), for a specific category of users, or for a particular context of interaction. “Adapted modalities” and “structured graphics” have been shown to combine their effects to create the best conditions for robust multimodal interaction [34].

To summarise, “error reduction by design” techniques exercise various levels of control and influence on user actions and behaviour. Error prevention is achieved by leading users towards the production of inputs that are easier to recognise by machines. Ultimately, these techniques contribute to reducing possible mismatch between a user’s mental model of how a recognition-based system works and the system’s model of the users. They also contribute to adjusting user expectations to the real capabilities of a system and establishing *grounding* between the interaction participants.
These techniques include: (1) isolated characters, (2) single-stroke alphabets, (3) tap-to-speak interfaces, (4) restricted grammars, (5) guided dialogues, (6) context-sensitive cues, (7) consistency and symmetry, (8) structured graphics, and (9) adapted modalities.

2.2. Error reduction by context

"Error reduction by context" achieves error reduction by augmenting user inputs with contextual information. Some multimodal systems, called **feature level multimodal systems**, are capable of integrating multiple tightly coupled inputs at very low levels of processing, so that the recognition process in one modality influences the course of recognition in the other modality. For example, "Machine lip reading" achieves speech recognition by combining acoustic information from the speech signal with visual information from the shapes of the speaker's lips. Computer vision techniques are used to extract information about the speaker's lip shapes, and this information is interpreted in terms of "visemes" (by analogy with the phonemes in the acoustic signal). Comparisons are then made between the visemes and phonemes that have been recognized to determine the most probable speech recognition output [37]. Lip reading has been shown to significantly improve automatic speech recognition accuracy, especially when the auditory speech is degraded [38]. However, feature level multimodal integration is only applicable to closely coupled and synchronised modalities, such as speech and lip movements for which both input channels provide corresponding information about the same content. Moreover, modelling complexity, computational intensity and training difficulty are typical problems associated with this approach. For example, a large amount of training data is required to build this type of systems, and multimodal training corpora are not readily available [39].

A more promising approach has recently emerged from two areas of computer science: ubiquitous computing and **context-aware systems**. Ubiquitous computing (or "pervasive computing") integrates computation into the environment. One of its goals is to enable devices to sense changes in their environment, which opens the access to a plethora of contextual information [40]. Systems that exploit such contextual information are said to be "context-aware". Context-aware systems can incorporate knowledge 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 and implicit contextual information. For example, in a video conferencing setting, combining carefully placed multiple distributed microphone pairs with calibrated cameras to identify the current speaker and their location, allows a finer control of the speech recognition process [41]. In a speaker localisation task, [42] shows that exploiting contextual audio information gives 65% accuracy versus 50% for a video-only baseline technique. Context is also becoming increasingly important in mobile computing (e.g. mobile guides) where the user's context often changes quickly [43], and has also been applied to "smart home" environments [44]. However, the concept of context in computing is still largely under-specified and the process of capturing and modelling contextual information for use in multimodal applications is currently the object of intensive research.

“Error reduction by context” techniques encompass two classes of systems: (10) feature level multimodal systems that can integrate tightly coupled interaction modalities to enhance signal recognition, and (11) context-aware systems, capable of sensing a user’s environment and exploiting multiple contextual information to achieve greater interaction robustness.

2.3. User prevention

In natural multimodal interfaces, the availability of multiple “adapted modalities” allows users to exercise their natural intelligence about when and how to deploy input modalities effectively. This has been called “agent assignment” in the CARE (Complementarity, Assignment, Redundancy, and Equivalence) usability framework [36]. CARE describes two types of assignment: strict assignment and agent assignment. Strict assignment expresses the absence of choice, and is when some modalities have been dedicated to specific tasks by system design. Conversely, agent assignment corresponds to the situation whereby the user has a choice, but always opts to use the same modality for executing the same task. We prefer to describe agent assignment as the opportunistic decision made by a user to use a modality rather than another for the execution of a specific task, in a particular interaction context. Agent assignment can take three different forms and serve three different user strategies for error prevention.

When users spontaneously select the interaction modality they believe is the most robust for a certain type of content, and avoid using the modalities that they believe are more error-prone, they can produce more robust inputs. In this case, **modality selection** is determined by recognition accuracy. For example, even if users initially prefer speech, they can learn with experience to avoid using speech when they are tired (speech recognition accuracy decreases with user physical exertion [3]), and decide to use keyboard entries instead for lengthy inputs. In the absence of a keyboard, when there is the choice of inputting data by voice or by handwriting, user preference will be influenced by how they perceive the reliability of the recognition result they are likely to obtain in each of the modalities. When the environment is noisy, handwriting will probably be preferred to speech.

Alternatively, users can learn that the confusion matrices for different lexical content differ across modalities, i.e. that the words that are easily confusable tend to be different for different recognisers. They can then exploit the asymmetric nature of different modalities by producing **complementary inputs** [45]. Complementarity describes the conjoint use of two modalities, whereby each modality conveys only partial information, but the integration of the two results in a complete and comprehensible message [36]. To be complementary, both modalities must be semantically rich, i.e. they must both carry significant information. With complementary inputs, inputs that are hard to recognise in one modality can be disambiguated based on inputs in the other modality. This has been called “mutual disambiguation of input modes” in [23] and is explained in more details in the next section.

Redundancy also describes the conjoint use of two modalities, but in this case the two modalities convey the same information [36]. For example, users produce **redundant inputs** when they choose to both speak and write the words of a message. As for complementary inputs, ambiguities in one modality can be resolved based on the second representation of the same information in the other modality. Redundancy implies that the two modalities are equivalent [36], which means that they are able to convey similar semantic content and can be used interchangeably. According to [45], equivalence fails to acknowledge the fact that different modalities are likely to differ in the type of information they transmit, their functionality during communication, and the way they are integrated with other modalities. Some modalities, for example speech and writing, may be relatively comparable, but most other modalities are not. Indeed, experimental evidence suggests that the dominant theme in users’ natural use of multimodal input is complementarity of content, not redundancy [45]. For example, in speech and pen interaction, it has been shown that speech and pen input consistently contribute different and complementary semantic information [34]. Even during multimodal correction of system errors, when users are highly motivated to clarify and reinforce their information delivery, speech and pen input rarely express redundant information [45].

To summarise, “user prevention” strategies describe strategies spontaneously adopted by users to prevent recognition errors. They include: (12) modality selection, (13) complementary inputs, and (14) redundant inputs.

2.4. Automatic detection

Many recognition systems return with each recognition hypothesis a confidence score that is a measure of the probability that a user’s input was correctly recognised. If this confidence measure is below some pre-defined **threshold**, the system will normally assume that the hypothesis is incorrect. When a recognition system does not directly return confidence measures, they can be calculated a priori from a confusion matrix, built while the recognition system (speech or handwriting) is being trained [46]. However, sensitivity to the environment (e.g. background noise) that is characteristic of recognition technologies makes these confusion matrices highly unstable and hence unreliable. Deciding on a threshold is in many cases an ad hoc process, and unfortunately, thresholding sometimes provokes false rejection errors, and does not contribute to the automatic correction of the rejected input.

A more sophisticated method has been suggested in [47], which consists in using a rule base to determine when errors may have occurred. Rules may be based on **semantic, pragmatic or common sense knowledge**. For example, a scheduling application might assume that an error has occurred if the user appears to want to schedule a meeting for 3 a.m. The Open Mind Common Sense Project [48] 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" [49]. The researchers found that their common sense speech recognition technique prevented 17 percent of the errors and reduced dictation time by 7.5 percent. It was particular efficient at disambiguating between words that are phonetically identical (e.g. "break" and "brake").

In section 2.2, it was explained that feature level multimodal systems were capable of integrating multiple inputs at very low levels of processing, but were only suitable for tightly coupled modalities such as speech and lip movements. The other fundamental type of multimodal architecture is called "semantic level architecture", and is more appropriate for modalities less tightly coupled such as speech and pen [39]. In this type of architecture, information is typically integrated at the semantic or pragmatic level [50]. Data structures called frames [51] are used to represent meaning and to merge information that result from different modality streams in a process called "semantic fusion" [52]. The automatic recovery from recognition errors and false interpretations can be achieved during this process, when complementary or redundant inputs (for example a speech utterance and a hand gesture) are combined. During semantic fusion, lists of possible interpretations provided by the speech and the gesture recognition system are compared, and interpretations that cannot be combined are filtered out [52]. The phenomenon in which an input signal in one modality allows the automatic discovery of impossible interpretations in a second signal in a different modality is called **mutual disambiguation** of input modes. It has been shown that the rate of mutual disambiguation achievable in a multimodal interface is particularly high for non-native speaker users of speech recognition systems [23].

In a similar but slightly different approach, synchronisation relationships between sets of modalities have been used to automatically detect and correct recognition errors. For example, timing information from 3D hand pointing gestures has been used to automatically detect recognition errors in speech [53][54]. Experimental studies have shown that, during speech and gesture multimodal interaction, 3D hand pointing gestures tend to be synchronised with either the nominal or the deictic ("this", "that", "here", etc.) expression of a phrase [53]. It has also been shown that the timing of the gesture is predictable in a [-200 ms, 400 ms] time interval around the beginning of the nominal or deictic expression. This quantitative **synchronisation model** of speech and gesture integration has then been used to predict the position of the nominal and deictic expressions in the user's speech. The speech interpretation that best verifies the speech-gesture synchronisation requirements described in the model, would have its rank in the list of possible interpretations and its confidence score increased. In [54], it is shown that the use of a speech and hand gestures synchronisation model can result in the recovery of up to a third of speech recognition errors.

In summary, techniques for the automatic detection of errors can be further classified into *statistical* ((15) thresholding), *rule-based* ((16) semantic, pragmatic and common sense knowledge), and *multimodal* ((17) mutual disambiguation and (18) synchronisation models) methods. Rule-based and multimodal methods in particular allow for not only the automatic detection of errors, but also, in some circumstances (i.e. when the correct output is present in one of the alternative hypotheses on the recogniser's list) for their automatic correction.

2.5. Machine-led discovery

When the machine leaves the error-handling process in the hands of the user, the best way it can facilitate error discovery is by providing adequate feedback on the recognition process and on its beliefs about what the user said or did.

In speech-only interfaces, one way of providing feedback is by repeating the user's input. This technique is called **implicit confirmation**. For example, if the user asks for a weather forecast for Boston for Tuesday, the system might respond "Tomorrow's weather for Boston is..." A flexible correction mechanism could then allow the user to contradict the system by saying "No, I said Tuesday". However, when compared with other strategies, implicit confirmation has been shown to result both in the longest time for the dialog to get back on track and the lowest rate of getting back on track [55].

In situations where the cost of recognition errors is particularly high (e.g. safety critical tasks), the use of **explicit confirmation** will allow a user to detect an error before the machine commits any action based on the misrecognised input. When using explicit confirmation, the system asks the user to confirm that what has been recognized or understood matches the user speech (e.g. "Do you really want to delete the file?"). Confirmations will help prevent the consequences of recognition errors from occurring. Explicit confirmation is particularly important for safety critical tasks (e.g. the piloting of an airplane) and in any situation where the risk of recognition errors cannot be afforded (e.g. a "delete all" command). However, the systematic use of confirmations can rapidly become very cumbersome and considerably slow down the interaction. In addition, if the confirmation is given by speech, it can be the carrier of further recognition errors.

Just as errors can be prevented by designing interfaces capable of influencing human actions and behaviour, systems can also employ a number of **mediation strategies** to lead users towards error discovery and correction [29][56]. Yankelovich *et al* [16], for example, introduce the idea of *progressive assistance* as a way of addressing repeated problems, such as repeated rejection errors, using increasingly targeted feedback. [29] lists a rich set of mediation strategies for resolving complications and errors: *freshness* (avoiding repeated utterances), *method shifts* (when one form of

instruction fails, trying another), *modality shifts* (switching or augmenting the modalities of communication, *e.g.* using visual rather than auditory cues), *level of discourse* (simplifying the vocabulary and language structure), *backtracking* (backtracking to the last state of mutual understanding), and *graceful failure* (offering natural exits for the user, such as time outs).
Multimodal speech systems that can **visually display a recognition result** make it easier for users to detect errors. If a rejection error occurs, no text will appear in the area where recognition results are displayed. If the recogniser makes a misrecognition or misfire error, the user can see what the recogniser thinks was said and correct any errors. With visual feedback, errors can also be identified by the system by highlighting words with low confidence scores or by the user by pointing at erroneous words.
Taking advantage of the fact that most recognition systems can provide a ranked list of several input interpretations, some multimodal systems give to users the opportunity to select the correct recognition result from a **list of alternative hypotheses** [57]. In this strategy, rather than providing the single most likely result, a voice recogniser provides to the graphical user interface (GUI) a list of multiple, likely possibilities, known as an *n-best list*. According to [58], however, the problem with allowing users to correct a word by choosing from a list of the computer's top ten or so possibilities is that the correct word or phrase is often not listed, and when this happens, it slows correction down considerably. Similarly, [59] proposes for graphics a drawing understanding system that displays for each input stroke, a bold line that represents the system's top guess, and dotted lines representing potential alternatives. If the user continues input as normal, the system implicitly accepts the default choice. For handwriting, [60] proposed an interface where the most likely choice of character is displayed in large font, with the two most likely alternatives displayed in smaller font below. However, this technique actually failed to be of any use in user tests due to the high cognitive overheads.

All of the strategies listed in this section are for the purpose of helping users to become aware of recognition errors and to facilitate their corrections. They include: (19) implicit confirmation, (20) explicit confirmation, (21) mediation strategies, (22) visual display of recognition results and (23) list of alternative hypotheses.

2.6. User correction

Most studies into human strategies for error correction in spoken interfaces have found that the most instinctive way for humans to correct mistakes using speech is to **repeat**. In handwriting, a similar strategy is to overwrite a misrecognised word. However, although repeating might be the most obvious way to correct when the system mishears, it is often the worse for the system [58][61]. The main reason for this is that a misheard word is likely to be a difficult one to understand in the first place. Moreover, continuous speech recognition systems are designed to understand a word in context, when a word is said in isolation it sounds very different. Finally, 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. However, [62] suggests that the situation can be improved by correlating repair input with the context of repair, i.e. instead of interpreting repair input as an independent event, other words that were correctly recognised in the vicinity of the error might be used in the process.

Another strategy for correcting errors in speech interfaces is to **spell out** the misrecognised words. But there is also a catch to using speech to spell a word in order to correct it. The main problem is that systems are not very good at recognising the switch to spelling [58]. Users must explicitly tell the system, for example through a dedicated command, that they are going to spell. This has the effect of slowing down the correction process. Moreover, in the context of a multimodal interface, spelling is not a natural method of error resolution, as users usually prefer to change modality if the repeat strategy has failed to work [63].

In [64] the author reports cases of linguistic adaptation, 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. In response to an implicit confirmation, users may also **contradict** the system. In order to support linguistic adaptation and contradiction, systems must implement large vocabularies and flexible grammars.

Experimental research has found that the two most common correction strategies when the system is cognisant of an error are rephrasing and repeating, and that they contribute to 82% of all user responses under error [55]. When the system is not necessarily cognisant of an error (such as when using an implicit confirmation strategy), users are most likely to contradict the erroneous system behaviour.

In multimodal systems, when recognition errors occur, [63] suggests that users are willing to repeat their input at least once, after which they will tend to **switch to another modality**. 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. This strategy implies that users renounce their initial modality selection choice. In 1999, Halverson et al investigated three commercially available speech recognition systems and compared error-correction strategies of first time users versus experienced users [12]. They found that, with experience, users learn to switch modalities. Inexperienced users, meanwhile, tend to stay with the speech modality and re-dictate the same word several times before switching techniques, incurring "re-dictation spirals" that considerably slow down text entry speed. In the study of an in-car system interface for controlling infotainment and communication services, which can be operated using speech, head, and hand gestures, as well as a touch panel and a key pad, it was found that when the system works properly, most users prefer tactile or speech interaction to head and hand gestures [65]. In case of hidden system errors (i.e. when users are unable to identify the cause of an interaction problem) changes from speech to tactile interaction and vice versa were observed, but none of the subjects selected head or hand gesture input as the leading fallback modality.

Modality switch is to be distinguished from what we will call **cross-modal correction**, which has often been referred to as "multimodal correction". Cross-modal correction involves the switch to another modality, not to re-enter a misrecognised input, but to select and/or correct the wrong output. For example, Suhm describes in [66] an experimental "multimodal listening typewriter", which implements four modalities: continuous speech, spelling, handwriting, and pen gestures. The user locates recognition errors by touching misrecognised words on a writing-sensitive screen where recognition output is displayed. Correcting errors can then be done by re-speaking, spelling, choosing from alternative words, typing, handwriting, and editing using gestures drawn on the display. Partial word repair is also supported, which allows a user to correct part of a word when it is almost correct, by replacing, inserting, or deleting letters with a pen or with spoken input. A user study on the use of the prototype showed that cross-modal correction makes it possible to input text at a faster speed than unskilled typing, including the time necessary to correct errors.

A particular case of cross-modal correction is when users select the correct output from a list of alternative hypotheses displayed on a GUI. The preferred method of error correction is normally to repeat the misrecognised input, but Larson and Mowatt found that when the n-best list option is improved by making it easier to access and dismiss, and by increasing its accuracy, a large increase in its frequency of use can be observed [67]. Eventually, the most popular repair strategy is to try the n-best list first and then switch to repeating if the correct alternative is not on the list.

Huerst et al. [68] have experimented with a handwriting pre-processor which looks for user **correction marks** on their own handwriting, such as crossing out a letter, and applies these corrections *before* sending handwriting to a recogniser. As speech recognition systems become more sophisticated and capable of recognising spontaneous speech, they will also eventually become capable of finding in user speech natural marks of self-correction such as pauses and word repetition.

Strategies for error correction include: (24) repeat, (25) spell out, (26) rephrase, (27) contradict, (28) modality switch, and (29) cross-modal correction. Finally, as recognition systems become more sophisticated, they may be able to interpret user natural (30) correction marks.

3. Taxonomy

In this section, we further classify the error-handling strategies into a taxonomy based on the following three dimensions: the main *actor* in the error-handling process (machine versus human), the *purpose* of the strategy (error prevention, discovery, or correction), and the use of different *modalities* of interaction.

3.1. Classification based on actor and purpose

Figure 2 shows all the strategies mentioned previously according to the two following variables: *actor* and *purpose*. According to these two variables, we define six categories of error-handling strategies: error prevention by machine, error prevention by users, error discovery by machine, error discovery by users, error correction by machine, and error correction by users.

Purpose / Actor	Error Prevention	Error Discovery	Error Correction
Machine	- *Error reduction by design* (1) isolated characters (2) single-stroke alphabets (3) tap-to speak interfaces (4) restricted grammars (5) guided dialogues (6) context-sensitive cues (7) consistency and symmetry (8) structured graphics (9) adapted modalities - *Reduction by context* (10) feature level multimodal systems (11) context-aware systems	- *Automatic detection* (15) thresholding (16) semantic, pragmatic and common sense knowledge (17) mutual disambiguation (18) synchronisation models	- *Automatic detection* (16) semantic, pragmatic and common sense knowledge (17) mutual disambiguation (18) synchronisation models
User	- *User prevention* (12) modality selection (13) complementary inputs (14) redundant inputs	- *Machine led discovery* (19) implicit confirmation (20) explicit confirmation (21) mediation strategies (22) visual display of result (23) list of alternative hypotheses	- *User correction* (24) repeat (25) spell out (26) rephrase (27) contradict (28) modality switch (29) cross-modal correction (30) correction marks

Figure 2. Taxonomy of error-handling strategies based on actor and purpose.

3.1.1. Error prevention by machine or 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, or greater recognition accuracy is achieved through the use of additional or contextual information. In the first case (reduction by design), error prevention is the result of improved user behaviour, such as shorter or less disfluent speech. This improved user behaviour is obtained thanks to interface design techniques, which is why the error-handling process is attributed to the machine. In contrast, the category “error prevention by users” comprises three strategies that rely on users spontaneously adapting their behaviour towards less error-prone interaction. User prevention strategies may sound like strategies on which the system has no control. In fact, nothing could be less true than that. These user strategies will only be applicable if they are adequately supported by system design. For example, we mentioned earlier (section 2.3) that adapted modalities had to be made available in the user interface, so users could exercise their natural intelligence on when and how to deploy input modalities effectively [45], through agent assignment. This has strong implications on multimodal interaction design and is discussed further in section 4.

3.1.2. Error discovery and error correction by machine

Error discovery by machine works in three possible ways: by using statistical data, by applying knowledge-based rules, or by exploiting cross-modal information.

There is an overlap between strategies by machine for error discovery and for error correction. With knowledge-based and multimodal 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 the list of alternative hypotheses produced during the recognition process. For example, if a domain knowledge-based rule tells the machine that a meeting is unlikely to be scheduled at 3 a.m., but likely to be scheduled at 3 p.m., and that both times appear in the recogniser's list of possible outputs, 3 p.m. will be chosen even if its accompanying confidence score is less than for the 3 a.m. hypothesis. Similarly, the *mutual disambiguation* and *synchronisation models* strategies can help recover from recognition errors when confidence scores are re-calculated based on the confidence scores of matching hypotheses in other recognisers' lists of possible outputs. Error correction by machine is thus dependent on the capacity of recognition systems to provide a ranked list of alternative hypotheses.

3.1.3. Error discovery and error correction by users

When a recognition error occurs, users are normally in charge of notifying the machine. It is crucial, however, that the machine facilitates error discovery through techniques such as the ones presented in section 2.5. (machine-led discovery). Although users are primarily responsible for finding errors, the machine holds a primary responsibility in enabling error discovery. Once errors have been found, users can effectively help the machine resolve them, usually by producing additional inputs.

It cannot be stressed enough that the quality (readily availability, ease of use, robustness and speed) of error correction mechanisms is of paramount importance for error correction success and for improving user experience with recognition-based systems. No perfect error correction mechanism has yet been proposed and error correction success and speed are very dependent on user ability to select and apply the most appropriate mechanism for the current type of error or context of interaction. There is a learning curve and experienced users have been shown to develop the appropriate automatisms or knowledge that considerably improve error correction success rate and speed [12]. The question on how to best support the development of these automatisms remains an important challenge in research on error correction.

3.2. Multimodal properties

It is possible to further classify the error-handling strategies by considering their multimodal properties. A modality of interaction can be described in terms of two sub-components: the mode and the media. The mode of interaction refers to how information is encoded, for example, natural language, graphics, and pointing are all different modes of interaction. The media describes the physical support or sometimes the sensory channel onto which the mode is expressed, for example, speech, pen, and keyboard are three different media that can convey the same mode (natural language). Multimodal error-handling strategies, indicated in underlined text in figure 2, are strategies that involve different modalities of interaction, i.e. either different modes (*multi-mode strategies*), different media (*multi-media strategies*), or different modes and media (*multi-modal strategies*). When there is a unique mode and media involved, the strategy is said to be *mono-modal*. The terms multimodal, multi-mode, multi-media, multi-modal, and mono-modal are only used here for explanation purposes, and do not, in any way, belong to a shared terminology. Figure 3 shows another classification of error-handling strategies based on their multimodal properties. Strategies that appear in *italic* in the figure are strategies that have been simultaneously classified in more than one category.

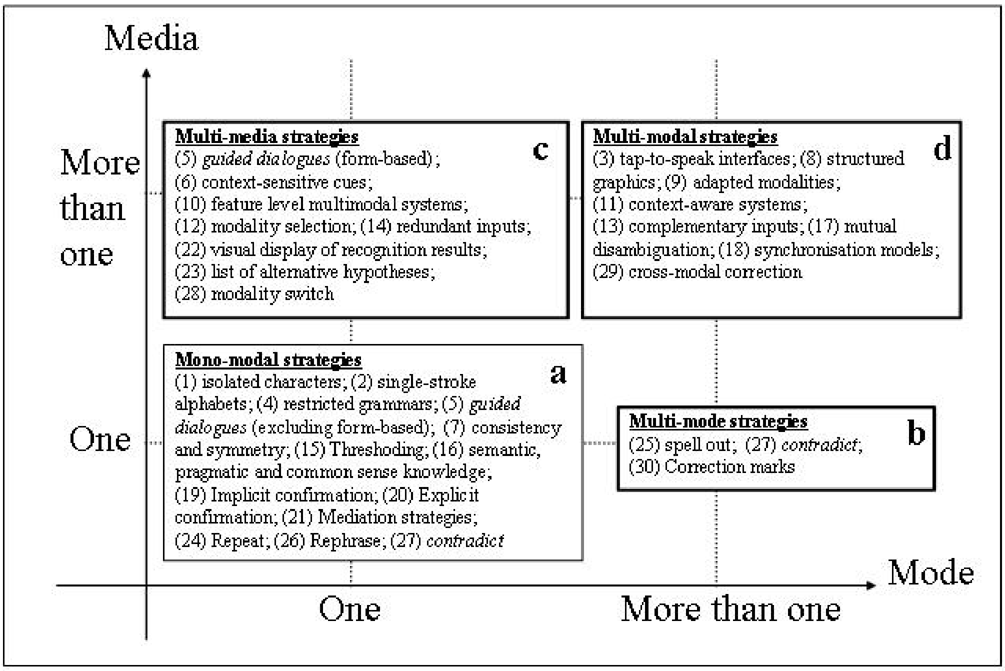

Figure 3. Multimodal properties: a) mono-modal strategies, b) multi-mode strategies, c) multi-media

strategies, and d) multi-modal strategies.

3.2.1. Mono-modal strategies

In mono-modal strategies, there is only one mode and one media involved. Mono-modal strategies typically characterise speech-only interfaces, and include speech-only strategies like *restricted grammars*; *guided dialogues* (excluding form-based dialogues); *consistency and symmetry*; *thresholding*; *semantic, pragmatic and common sense knowledge*; *implicit confirmation*; *explicit confirmation*; *mediation strategies*; *repeat*; *rephrase*; and *contradict*. This rather long list of speech-only strategies reveals two things about speech interaction. First, it is a testimony to the relatively long and strenuous history of speech recognition software, compared with more recent technologies such as pen computing and computer vision. Second, it shows that, even in a speech-only interface there is a large range of strategies available for handling recognition errors.

When robustness is a high priority, restricted grammars supported by guided dialogues can contribute to minimising recognition errors, but at the price of a somewhat constrained and unnatural interaction. Consistency and symmetry aims at achieving the same result but in a less intrusive way, with also less guarantee of success. Common sense knowledge sounds like a very promising and seamless strategy, which is nevertheless dependent on the amount and quality of common sense that can be instilled into the system and on the system's capability to use it. Error discovery strategies such as implicit and explicit confirmation are very cumbersome. Explicit confirmation is nonetheless required in safety critical tasks. As far as the repeat user correction strategy is concerned, it has been shown to be inefficient because often leading to spiralling error situations. To avoid such situations, mediation strategies should be used in conjunction with Repeat. They are a form of guided dialogues, useful to get interaction back on track after a series of errors. They mostly consist in temporarily shifting dialogue control from the user to the machine.

In brief, a speech-only interface should not rely on a unique error-handling mechanism, but should implement strategic sets of mechanisms that effectively support each others. Mono-modal strategies are not a monopoly of speech interfaces, they can also be found in pen computing (*isolated characters* and *single-stroke alphabets)*.

3.2.2. Multi-mode strategies

Speech-only and pen-only interfaces are not limited to mono-modal error handling strategies, as speech and pen can be used through different modes [69], which makes multi-mode error handling strategies possible.

The *spell out* strategy for error correction by users is a strategy that involves a single media (speech), which is used through two different modes (natural language and spelling). Although spelling has not usually been classified as a multimodal error-handling strategy in the literature, it does in effect force the system to recognise a second interaction mode that is different from the normal language of the application. Even the strategy *contradict* could be implemented as a multi-mode strategy, if the speech recognition system enters a special “contradiction mode” after an implicit confirmation, which enables it to recognise with better accuracy the specific constructions that typically enter in the formulation of contradiction.

Similarly, the *correction marks* strategy for handwriting involves a single media (a pen) and two modes (writing and correction marks).

Multi-mode strategies require that the unique media is capable of supporting several different encoding schemes or “modes”. Speech and pen are two media that are capable of this. Pen, for example, can be used for pointing, writing and drawing. However, in multi-mode interaction there is always a little danger of confusion: the system must be capable of recognising in which mode the user is operating, and vice versa. In particular, the system must be able to handle mode switch and be explicit in its reliance on specific modes.

3.2.3. Multi-media strategies

Multi-media strategies involve a unique mode that is expressed through different media. *Guided dialogues* (form-based); *context-sensitive cues*; *feature level multimodal systems*; *modality selection*; *redundant inputs*; *visual display of results*; *list of alternative hypotheses*; and *modality switch* can all be considered multi-media strategies.

Form-based guided dialogues combine user spoken input and textual system output, where users and system share the same language (i.e. the same mode). The context-sensitive cues strategy, which consists in placing command names as visual labels in a 3D scene in order to inform users about possible spoken commands in the current context of interaction, uses a single mode (the language of the application) and two different media (visual labels and speech). Similarly, visual display of results and list of alternative hypotheses combine the uses of spoken input and visual output of the recognised utterances. All of these strategies require the availability of a visual display and thus are not applicable in a speech-only interface.

Feature level multimodal systems, modality selection, redundant inputs, and modality switch, all require that several input media (e.g. speech and lips, or speech and pen) have similar powers of expression. The different media must be able to support similar input modes and convey similar information. As explained earlier, such media are more the exception than the norm, which considerably reduces the applicability of these four strategies (this is further discussed in section 4).

3.2.4. Multi-modal strategies

Multi-modal strategies, finally, are strategies that involve more than one mode and more than one media. They include: *tap-to-speak interfaces; structured graphics; adapted modalities; context-*

aware systems; complementary inputs; mutual disambiguation; synchronisation models; and *cross-modal correction.*

Tap-to-speak interfaces require a pen tap before each spoken input. If the pen is also used for other interaction modes such as handwriting, gestures, or drawing, the necessity of tapping before speaking can dramatically alter the way users combine pen inputs with speech. We see this as a major drawback of the strategy.

Structured graphics combine spoken utterances with visual display of graphics. The degree of efficiency of the strategy is highly dependent on the nature of the task. Its efficiency has been demonstrated for interactive maps and for other tasks that are largely spatial in nature.

Context-aware systems combine, for example, visual information captured by a camera, with user spoken utterances. This is a very promising strategy, which is still in its infancy. As explained earlier, the processes of capturing, representing and then exploiting contextual information are currently the objects of very intense research.

The synchronisation model strategy is dependent on the availability of quantitative models that characterise timing relationships between sets of modalities. The model presented in section 2.4 describes the synchronisation of speech and 3D hand pointing gestures, but very few other attempts at eliciting such models have been made. Speech and pen synchronisation provides another rare example, where pen input onset has been shown to consistently precede speech onset [70]. These models are the exception, and the synchronisation model strategy is dependent on more models being developed.

Cross-modal correction expresses a shift of method for inputting information that usually involves several media and modes. Several user studies have shown that cross-modal correction is one of the most efficient and speedy method for correcting recognition errors, once users have learnt to use it.

Finally, adapted modalities, complementary inputs and mutual disambiguation exploit the asymmetric natures of different input modes and media. These three strategies share complex relationships with different modality properties and modality combinations, which are discussed in details in the next section.

4. Multimodal interaction design

Many error-handling strategies originate from previous work on spoken dialogue systems, but more recently, strategies that specifically leverage from diverse combinations of modalities have emerged. It is thus important to understand the requirements these strategies have on different combinations of modalities and the implications of these requirements for multimodal interaction design. Especially, when designing a multimodal interface, careful consideration must be given to the choice of input and output modalities available in the interface, the allocation of modalities to tasks and information types, and the range of possible combinations of modalities. The requirements different multi-media and multi-modal error-handling strategies have on different sets of modalities have been summarized in figure 4.

Classes of interfaces / Multimodal strategies	a Speech input and visual output	b Mutimodal input			
		C (e.g. speech and gestures or drawing)	A	R	E (e.g. lip reading, speech and keyboard or handwriting)
Multi-media strategies					
Guided dialogues (form based)	X				
Context sensitive cues	X				
Feature level integration				X	X
Modality selection			agent		X
Redundant inputs				X	X
Visual display of results	X				
List of alternative hypotheses	X				
Modality switch					X
Multi-modal strategies					
Tap-to-speak		X	strict		
Structured graphics	X				
Adapted modalities		X	agent		X
Context aware systems		X			
Complementary inputs		X			
Mutual disambiguation		X		X	
Synchronisation models		X			
Cross-modal correction		X			

Figure 4. Multi-media and multi-modal error-handling strategies, and classes of multimodal interfaces. CARE = Complementarity, Assignment, redundancy, and Equivalence [36].

The availability of a visual display together with the speech input modality (column a of figure 4) opens the door to a number of error-handling strategies: form-based guided dialogues, context sensitive cues, visual display of results, list of alternative hypotheses, and structured graphics. In a multimodal interactive map for example (spatial task), context sensitive cues and structured graphics are particularly appropriate. In a speech dictation application (verbal task), visual display of results and list of alternative hypotheses are recommended. And for a dialogue application, form-based guided dialogues will guarantee a high level of robustness.

When several input modalities are available (column b of figure 4), possible error-handling strategies will depend on the set of modalities available. A framework for reasoning about different sets of modalities is thus necessary. To date, the most useful framework for reasoning about sets of modalities is the CARE (Complementarity, Assignment, Redundancy, and Equivalence) usability framework [36] introduced in section 2.3. The CARE framework is not perfect as it does not provide a clear distinction between the intrinsic properties of sets of modalities and modality usage. For example, Equivalence is a property, as two or more modalities can be described as being equivalent, independently of any particular use, whereas Redundancy describes the simultaneous use of two or more equivalent modalities. The CARE properties are not independent from each others and share complex relationships; nevertheless, they are useful for describing multimodal interaction.

Modality Equivalence is necessary if one wants to offer users the possibility of exercising their natural intelligence about which modality to use (*modality selection*), and of modifying their original choice when necessary (*modality switch*). Equivalent modalities are also required for the production of *redundant inputs*, which can then trigger *mutual disambiguation*; and for the *feature level integration* of tightly coupled multimodal inputs. Examples of equivalent modalities include: speech and lip movements (leading to feature level integration); speech and symbolic hand gestures (e.g. the word “five” accompanied by a gesture clearly showing the five fingers of the hand); and speech and typing or handwriting.

Speech, drawing and pen gestures are more likely to be used complementarily than redundantly. In interactive maps, speech is used for entering commands, whereas pen gestures are used for designating geographical areas or graphical elements of the map. These modalities are complementary and not interchangeable (i.e. not equivalent). It has been shown in [71] that user attempts to self-manage limitations on working memory when task complexity increases is accomplished by distributing communicative information across modalities (i.e. by producing more *complementary inputs*). For example, during speech and pen interaction with a map, user combined use of both modalities increases as the task becomes more challenging. So ideally, a multimodal interface should implement a large number of semantically rich, heterogeneous and maximally complementary modalities [72]. Increased use of complementary inputs also increases the chance of successful *mutual disambiguation* in the multimodal architecture. However, it has been shown that the degree to with which mutual disambiguation of complementary inputs can operate in a given multimodal interface is dependent on the set of multimodal constructions that the system is able to interpret [73]. To determine how efficiently complementarity can lead to mutual disambiguation, a more practical approach to multimodal interaction design is necessary. For example, it is suggested in [74] that a simple formalism such as Finite State Machines (FSM) can be used to rapidly build several multimodal interaction designs and then assess their respective potential for mutual disambiguation of input signals.

Context aware systems, *synchronization models* and *tap-to-speak* interfaces are special cases of modality complementarity, where only one modality conveys semantically rich content. For example, information captured by a camera about the location of a speaker will help managing the speech recognition process better and provide context to the speech, but speech is the only semantically significant input. Similarly, timing information provided by hand gestures can help locate in the speech signal the parts that are more semantically significant, without contributing any additional meaning. Tap-to-speak interfaces provide the only example of complementary modalities with strict assignment: users are required to tap with the pen before talking; the combination of the tap signal with the speech will then trigger the speech recognition process. In all three cases, the speech modality is said to be dominant.

Simultaneity or some degree of synchronisation are required in the use of complementary modalities for *complementary inputs*, *context aware systems*, *mutual disambiguation,* and *synchronisation models* strategies. In contrast, the *cross-modal correction* strategy uses sequential complementary inputs, as the second input is entered only after the first input has been misrecognised. This particular property of cross-modal correction allows the conjoint use of modalities such as handwriting and pen gestures that can hardly be used simultaneously to produce complementary inputs, but that can enter in the formulation of sequential inputs (e.g. when a pen gesture is used to delete a character from a misrecognised handwritten input).

5. Conclusion

In recognition-based multimodal interfaces, the design of effective means of error prevention, detection, and correction remains one of the main determinant factors of usability and users' acceptance. Many error-handling mechanisms have been proposed and tried. The main aim of the classification presented in this paper is to serve as a tool to guide the choice and design of adequate error-handling strategies in error-prone multimodal systems.

One obvious way of using the taxonomy is to ensure that at least one error-handling strategy in each of the six categories defined in the classification is implemented. In systems where the strategies for error correction by users are likely to be inefficient (e.g. repeat in speech only interfaces in noisy environments), or where the potential costs for error correction and their consequences are high (e.g. safety critical systems), particular attention should be made to implementing strategic sets of error prevention strategies.

The taxonomy also highlights important relationships between strategies that need to be taken into account when designing a system. For example, error correction strategies by users should be supported by adequate error discovery strategies. Similarly, some error-handling strategies rely on particular system's characteristics such as the provision of a visual display or the availability of statistical data on the recognition process (e.g. confidence scores).

The design of the language of the application (grammar and prompts) and the choice of modalities of interaction that are made available in the user interface are two critical issues for the handling of recognition errors. In particular, sensitive trade-offs need to be made between constraining the interaction to help error prevention by machine, and allowing more flexibility to enable error prevention and correction by users. Well adapted modalities of interaction for given types of inputs and situations have been shown to contribute to error reduction. One important challenge in multimodal interaction design is thus to select the best set of interaction modalities. Every effort should be made to implement both equivalent and complementary modalities in a multimodal interface.

Two strategies seem particularly promising for future multimodal human-machine interfaces. Context-aware systems, capable of sensing a user's environment, will be able to provide a plethora of useful information for error prevention and discovery. Much research is currently being done to define the "context of interaction" and to develop technologies for the automatic detection and understanding of this context. Recognition systems capable of recognising and interpreting natural users' correction marks in speech and handwriting inputs constitute another promising technology. These systems will not only improve the robustness of multimodal interaction, but also, and maybe more importantly, its naturalness and spontaneity.

References

[1] E.G. Devine, S.A.Gaehde and A.C. Curtis, Comparative evaluation of three continuous speech recognition software packages in the generation of medical reports. Journal of the American Medical Information Association 7 (2000) 462-68.

[2] H.H. Koester, User performance with speech recognition: a literature review. Journal of Assistive Technology 13 (2) (2001) 116-30.

[3] M.S. Entwistle, The Performance of Automated Speech Recognition Systems Under Adverse Conditions of Human Exertion. International Journal of Human-Computer Interaction 16 (2) (2003) 127-140.

[4] S. Furui, Recent Progress in Corpus-Based Spontaneous Speech Recognition, IEICE Transactions on Information and Systems E88-D (3) (2005) 366-375.

[5] L.R.B. Schomaker, User-interface aspects in Recognising Connected-Cursive Handwriting, Proc. IEE Colloquium on handwriting and Pen-based input, London, UK, March 1994.

[6] I.S. MacKenzie and L. Chang, L, A performance comparison of two handwriting recognisers, Interacting with Computers 11 (1999) 283-297.

[7] J.C. Read, S.J. MacFarlane and C. Casey, Measuring the Usability of Text Input Methods for Children, Proc. HCI2001, Lille, France, September 2001, pp. 559-572.

[8] I.S. MacKenzie and R.W. Soukoreff, Text entry for mobile computing: Models and methods, theory and practice, Human-Computer Interaction 17 (2002) 147-198.

[9] R. Plamondon and S.N. Srihari, On-line and off-line handwriting recognition: A comprehensive survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (1) (2000) 63-84.

[10] B. Suhm, B. Myers and A. Waibel, Model-based and empirical evaluation of multimodal interactive error correction, Proc. ACM CHI'99, Pittsburgh, Pennsylvania, May 1999, pp. 584-591.

[11] C-M. Karat, C. Halverson, D.Horn and J. Karat, Patterns of entry and correction in large vocabulary contentious speech recognition systems, Proc. ACM CHI'99, Pittsburgh, Pennsylvania, May 1999, pp. 568-575.

[12] C. Halverson, D. Horn, C. Karat and J. Karat, The Beauty of Errors: Patterns of Error Correction in Desktop Speech Systems, Proc. IFIP INTERACT'99, Edinburgh, Scotland, UK, September 1999, pp. 133-140.

[13] J.M.Rieger, The effect of automatic speech recognition systems on speaking workload and task efficiency, Disability and Rehabilitation 25 (2003) 224-35.

[14] J.R. Lewis, Effect of Error Correction Strategy on Speech Dictation Throughput, Proc. Human Factors and Ergonomics Society 43rd Annual Meeting, 1999, pp. 457-461.

[15] J. Mankoff, S.E. Hudson and G.D. Abowd, Interaction techniques for ambiguity resolution in recognition-based interfaces, Proc. ACM UIST'00, San Diego, California, November 2000, pp. 11-20.

[16] N. Yankelovich, G.A. Levow and M. Marx, Designing SpeechActs: Issues in Speech User Interfaces, Proc. CHI'95, Denver, CO, May 1995, pp. 369-376.

[17] H.H. Koester, Usage, performance, and satisfaction outcomes for experienced users of automatic speech recognition, Journal of Rehabilitation Research & development 41 (5) (2004) 739-754.

[18] C. Frankish, R. Hull and P. Morgan, Recognition Accuracy and User Acceptance of Pen Interfaces, Proc. ACM CHI'95, Denver, Colorado, May 1995, pp. 503-510.

[19] J. Mankoff and G.D. Abowd, Error Correction Techniques for Handwriting, Speech, and other ambiguous or error prone systems, GVU TechReport GIT-GVU-99-18, June 1999. http://www.cc.gatech.edu/fce/errata/publications/survey/

[20] D. Goldberg and C. Richardson, Touch-Typing with a Stylus, Proc. INTERCHI'93, Amsterdam, the Netherlands, April 1993, pp. 80-87.

[21] I.S. MacKenzie and S.X. Zhang, The immediate usability of Graffiti, Proc. Graphics Interface, Kelowna, British Columbia, Canada, May 1997, 120-137.

[22] S. L. Oviatt, P. R. Cohen and M. Wang, Toward interface design for human language technology: Modality and structure as determinants of linguistic complexity. Speech Communication 15 (1994) 283-300.

[23] S. Oviatt, Mutual disambiguation of recognition errors in a multimodal architecture, Proc. ACM CHI'99, Pittsburgh, Pennsylvania, May 1999, pp. 576-583.

[24] G.W. Furnas, T.K. Landauer, L.M. Gomex and S.T. Dumais, The vocabulary problem in human system communication, Communications of the ACM 30 (1987) 964-971.

[25] N. Dahlbäck, A. Jönsson and L. Ahrenberg, L. Wizard of Oz Studies Why and How, Proc. International Workshop on Intelligent User Interfaces, Orlando, Florida, January 1993, pp. 193-200.

[26] M.A. Walker, J.F. Di Fabbrizio, C. Mestel and D. Hindle, What can I say?: evaluating a spoken language interface to email. Proc. CHI'98, Los Angeles, California, April 1998, pp. 582-589.

[27] P.A. Heeman, M. Johnston, J. Denney and E. Kaiser, Beyond Structured Dialogues: Factoring Out Grounding, Proc. International Conference on Spoken Language Processing (ICSLP), Sydney, Australia, December 1998, 863-866.

[28] M. Turunen, Error handling in speech user interfaces in the context of virtual worlds, Proc. ACHCI'98, Tampere, Finland, 1998, pp. 68-75.

[29] J. Heer, N. Good, A. Ramirez, M. Davis and J. Mankoff, Presiding Over Accidents: System Direction of Human Action, Proc. ACM CHI'04, Vienna, Austria, April 2004, pp. 463-470.

[30] E. Zoltan-Ford, How to Get People to Say and Type What Computers Can Understand, International Journal of Man-Machine Studies 34 (4) (1991) 527-547.

[31] S.E. Brennan and E.A Hulteen, Interaction and Feedback in a Spoken Language System: A Theoretical Framework, Knowledge-Based Systems 8 (2-3) (1995) 143-151.

[32] J.A. Larson, VoiceXML: Introduction to Developing Speech Applications, Prentice Hall, 2002.

[33] S. Pan, S. Shen, M. Zhou and K. Houck, Two-Way Adaptation for Robust Input Interpretation in Practical Multimodal Conversation Systems, Proc. IUI'05, San Diego, California, January 2005, pp. 35-42.

[34] S. Oviatt, Multimodal Interactive Maps: Designing for Human Performance, Human-Computer Interaction 12 (1997) 93-129.

[35] S. Oviatt, Predicting spoken disfluencies during human-computer interaction. Computer Speech and Language 9 (1995) 19-35.

[36] J.L. Coutaz, L. Nigay, D. Salber, A. Blandford, J. May and R.M. Young, Four Easy Pieces for Assessing the Usability of Multimodal Interaction: the CARE Properties, Proc. INTERACT'95, Lillehammer, Norway, 1995, pp. 115-120.

[37] G. Potamianos, C. Neti, G. Gravier, A. Garg and A.W. Senior, Recent advances in the automatic recognition of audio-visual speech, Proc. IEEE 91 (9) (2003) 1306-1326.

[38] U. Meier, R. Stiefelhagen, J. Yang, and A. Waibel, Towards unrestricted lipreading, International Journal of Pattern Recognition and Artificial Intelligence 14 (2000) 571-585.

[39] S.L. Oviatt, P.R. Cohen, L. Wu, J. Vergo, L. Duncan, B. Suhm, J. Bers, T. Holzman, T. Winograd, J. Landay, J. Larson and D. Ferro, Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions, Human Computer Interaction

15 (4) (2000) 263-322.

[40] G.D. Abowd and E.D. Mynatt, Charting Past, Present, and Future Research in Ubiquitous Computing, ACM Trans. Computer Human Interaction 7 (1) (2000) 29-58.

[41] B.H. Yoshimi and G.S. Pingali, A multimodal speaker detection and tracking system for teleconferencing, *Proc*. ACM Conference on Multimedia, Juan-les-Pin, France, December 2002, pp 427-428.

[42] H.J. Nock, G. Iyengar and N. Chalapathy, Multimodal processing by finding common cause, Communications of the ACM 47 (1) (2004) 51-56.

[43] W. Yue, S. Mu, H. Wang and G. Wang, TGH: A Case Study of Designing Natural Interaction for Mobile Guide Systems, Proc. Mobile HCI'05, Salzburg, Austria, September 2005, pp. 199-206.

[44] M.A. Feki, S. Renouard, B. Abdulrazak, G. Chollet and M. Mokhtari, Coupling Context Awareness and Multimodality in Smart Homes Concept, Lecture Notes in Computer Science 3118 (2004) 906-913.

[45] S.L. Oviatt, Ten myths of multimodal interaction, Communication of the ACM, 42 (11) (1999) 74-81.

[46] M. Marx and C. Schmandt, Putting People First: Specifying proper names in speech interfaces, Proc. ACM UIST’94, Marina del Rey, California, November 1994, pp. 30-37.

[47] C. Baber and K.S. Hone, Modelling Error Recovery and Repair in Automatic Speech Recognition, International Journal of Man-Machine Studies 39 (3) (1993) 495-515.

[48] P. Singh, The public acquisition of commonsense knowledge, Proc. AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access, Palo Alto, California, 2002.

[49] H. Lieberman, A. Faaborg, W. Daher and J. Espinosa, How to Wreck a Nice Beach You Sing Calm Incense, Proc. IUI'05, San Diego, California, January 2005, 278-280.

[50] M.L. Bourguet, Software Design and Development of Multimodal Interaction, Proc. IFIP 18th

World Computer Congress Topical Days, Toulouse, France, August 2004, 409-414.

[51] M. Minsky, A framework for presenting knowledge, The Psychology of Computer Vision, P.H. Winston (ed.), McGraw-Hill, 1975.

[52] M. Johnston, Unification-based Multimodal Parsing, Proc. 36th COLING-ACL, Montreal, Canada, August 1998, pp. 624-630.

[53] M.L. Bourguet and A. Ando, Synchronisation of Speech and Hand Gestures during Multimodal Human-Computer Interaction, Proc. ACM CHI'98 Volume 2, Los Angeles, California, April 1998, pp. 241-242.

[54] M.L. Bourguet and A. Ando, Speech timing prediction in multimodal human-computer interaction, Proc. IFIP INTERACT'97, Sydney, Australia, July 1997, pp. 453-460.

[55] S. Narayanan, Towards modeling user behavior in human-machine interactions: Effect of Errors and Emotions, Proc. ISLE Workshop on Dialogue Tagging for Multi-modal Human Computer Interaction, Edinburgh, U.K, December 2002.

[56] M. Zajicek and J. Hewitt, An Investigation into the Use of Error Recovery Dialogues in a User Interface Management System for Speech Recognition, Proc. IFIP INTERACT'90, Cambridge, UK, August 1990, pp. 755-760.

[57] J. Mankoff, S. Hudson and G.D. Abowd. Providing Integrated toolkit-level support for ambiguity in recognition-based interfaces, Proc. ACM CHI'00, The Hague, The Netherlands, May 2000, pp. 368-375.

[58] B. Suhm, B. Myers and A. Waibel, Multimodal error correction for speech user interfaces, ACM Transactions on Computer-Human Interaction (TOCHI) 8 (1) (2001) 60-98.

[59] T. Igarashi, S. Matsuoka, S. Kawachiya and H. Tanaka, Interactive Beautification: A Technique for Rapid Geometric Design, Proc. ACM UIST'97, Banff, Alberta, Canada, October 1997, pp. 105-114.

[60] D. Goldberg and A. Goodisman, STYLUS User Interfaces for Manipulating Text, Proc. ACM UIST'91, Hilton Head, South Carolina, November 1991, pp. 127-135.

[61] C. Frankish, D. Jones and K. Hapeshi, Decline in accuracy of automatic speech recognition as a function of time on task: fatigue or voice drift? International Journal of Man-Machine Studies 36 (1992) 797-816.

[62] B. Suhm and A. Waibel, Exploiting repair context in interactive error recovery, Proc. Eurospeech 97, 5th European Conference on Speech Communication and Technology, Rhodos, Greece, September 1997, 1659-1661.

[63] S. Oviatt and R. van Gent, Error Resolution During Multimodal Human–Computer Interaction, Proc of the Fourth International Conference on Spoken Language Processing, Philadelphia, Pennsylvania, October 1996, pp. 204-207.

[64] S. Oviatt, Taming recognition errors with a multimodal interface, Communications of the ACM 43 (9) (2000) 45-51.

[65] G. McGlaun, F. Althoff, M. Lang and G. Rigoll, Towards Multimodal Error Management: Experimental Evaluation of User Strategies in Event of Faulty Application Behavior in Automotive Environments, Proc. 7th World Multiconference on Systemics, Cybernetics, and Informatics SCI, Orlando, Florida, July 2003, pp. 462-466.

[66] B. Suhm, Empirical evaluation of interactive multimodal error correction, Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, Santa Barbara, California, December 1997, pp. 583-590.

[67] K. Larson and D. Mowatt, Speech error correction: the story of the alternates list, International Journal of Speech Technology 6 (2) (2003) 183-194.

[68] W. Huerst, J. Yang and A. Waibel, Interactive Error Repair for an Online Handwriting Interface, Proc. ACM CHI'98 (Poster), Los Angeles, California, April 1998, pp. 353-354.

[69] J. Coutaz and J. Caelen, A taxonomy for multimedia and multimodal user interfaces, Proc. 1st ERCIM Workshop on Multimodal Human-Computer Interaction, Lisbon, Portugal, November 1991.

[70] S. Oviatt, A. DeAngeli and K. Kuhn, Integration and synchronization of input modes during multimodal human-computer interaction. Proc. ACM CHI '97, Atlanta, Georgia, March 1997, pp. 415-422.

[71] S. Oviatt, R. Coulston and R. Lunsford, When Do We Interact Multimodally? Cognitive Load and Multimodal Communication Patterns, Proc. ICMI 2004, State College, Pennsylvania, October 2004.

[72] S. L. Oviatt, S.L. Breaking the Robustness Barrier: Recent Progress on the Design of Robust Multimodal Systems, Advances in Computers 56, M. Zelkowitz Eds., Academic Press, 2002, pp. 305-341.

[73] M.L. Bourguet, Designing and Prototyping Multimodal Commands, Proc. INTERACT'03, Zurich, Switzerland, September 2003, pp 717-720.

[74] M.L. Bourguet, How Finite State Machines Can be Used to Build Error Free Multimodal Interaction Systems, Proc. HCI'03, Volume 2, Bath, UK, September 2003.