



# Effective error recovery strategies for multimodal form-filling applications

Janienke Sturm \*, Lou Boves

*Department of Language and Speech, University of Nijmegen, Postbus 9103, 6500HD, Nijmegen, The Netherlands*

Received 1 February 2004; received in revised form 8 November 2004; accepted 11 November 2004

---

## Abstract

The goal of the research described in this article is to determine in what way speech recognition errors can be handled best in a multimodal form-filling interface. Besides two well-known error correction mechanisms (re-speaking the value and choosing the correct value from a list of alternatives), the interface offers a novel correction mechanism in which the user selects the first letter of the target word from a soft-keyboard, after which the utterance is recognized once again, with a limited language model and lexicon. The multimodal interface that was used is a web-based form-filling GUI, extended with a speech overlay, which allows for pen and speech input. The effectiveness and efficiency of the error correction mechanisms, the error correction strategies that are applied by the users and the effects on user satisfaction were studied in an evaluation in which the interface was tested in two conditions: in one condition (LIST), the interface provides only re-speaking and the alternatives list as error correction facilities. In the other condition (LETTER), the interface provides the soft-keyboard technique as an additional error correction facility. The study shows that error correction was more effective in the LETTER condition than in the LIST condition. The Keyboard correction facility enables the users to solve errors that could not be solved using the Re-speak method or by choosing from a list of alternatives. In spite of its low effectiveness, subjects initially attempted to use Re-speaking for error correction in both interfaces. However, we also found that subjects rapidly learned to choose the most effective option (Keyboard) immediately as they gain experience. The user satisfaction turned out to be higher for the LETTER interface than for the LIST interface: subjects considered the LETTER interface to be more useful and less frustrating and they felt more in control. As a result, most subjects clearly preferred the LETTER interface.

© 2005 Elsevier B.V. All rights reserved.

*Keywords:* Multimodal interfaces; Error correction; Speech recognition

---

\* Corresponding author. Tel.: +31 40 247 8396; fax: +31 40 247 5376.  
E-mail addresses: [j.sturm@tue.nl](mailto:j.sturm@tue.nl) (J. Sturm), [l.boves@let.ru.nl](mailto:l.boves@let.ru.nl) (L. Boves).  
URL: <http://lands.let.ru.nl>.

## 1. Introduction

This article addresses error correction in speech-centric multimodal information systems: how are speech recognition errors handled best in a multimodal form-filling interface? Automatic speech recognition (ASR) is applied increasingly often in information systems: a great deal of research effort is put into finding ways to enhance graphical user interfaces (GUIs) with a speech overlay for situations in which the keyboard and mouse cannot be used as an input devices, as in services that are accessed through a PDA (for examples of prototype systems, cf. Cohen et al., 1997; Huang et al., 2001; Johnston et al., 2002). Although speech recognition accuracies nowadays can be quite high (in limited domains and in relatively quiet environments), speech recognition errors will always occur, if only because of the existence of (near) homophones and ambiguous pronunciations. Speech recognition errors strongly influence the way people perceive a speech driven information system: they dislike interactive systems that make a lot of errors (Karat et al., 1999). The appreciation of an interface depends also on the ease of use of the facilities that are offered to correct errors (Mankoff and Abowd, 1999; Zajicek and Hewitt, 1990).

In error handling in interactive systems three issues are at stake (Mankoff and Abowd, 1999). The first issue is *error reduction* or *error prevention*. Obviously, the number of errors should be as low as possible, which can be achieved by optimizing the accuracy and robustness of the speech recognizer (or other recognition components) for the given task. Furthermore, it has been shown that multimodal interaction can enhance ASR performance in several ways (Oviatt, 2000). Firstly, the language people use when interacting multimodally tends to be brief and simple from a grammatical point of view. Moreover, people tend to choose the modality that they think is best suited for the information that has to be conveyed. Finally, some applications allow combining multiple modalities in such a way that mutual disambiguation can take place. Whereas the first two advantages apply to many different types of multimodal interfaces, mutual disambiguation mostly applies

to applications where referring expressions or spatial commands play an important role; thus, it is not likely to be successfully applied in form-filling interfaces (Oviatt, 1999).

The second issue in error handling is *error detection*. Either the user or the system must detect that an error has occurred, before steps can be taken to correct the error. Although it may be very hard for a system to detect its own mistakes, it may decide that a specific value is likely to be incorrect on the basis of confidence measures or by looking at the dialogue history. By providing feedback of the recognition result, either in a spoken confirmation question or visually, the system enables the user to discover the error and to notify the system that a mistake has been made.

The final issue in error handling is *error correction*: if an error has been detected, either the user or the system must try to solve it. A multimodal interface in which speech is combined with pen input, may support error correction in various ways, for example by allowing the user to select the correct option from a list, or type the correct value on a soft-keyboard, in addition to re-speaking. Facilities for correcting recognition errors in multimodal human-machine communication and the strategies users apply when faced with errors have been studied before by various authors in several conditions. Oviatt and VanGent (1996) studied error correction in the context of a Wizard-of-Oz study with a multimodal transaction system. They examined how users strategically adapt and integrate their use of input modes and lexical expressions while resolving recognition errors. They found that speech was preferred over writing as input mode; in addition, if a recognition error was detected, their subjects tried to correct it by re-speaking. However, if the error persisted, subject switched to writing. In the users' perception, speaking more slowly (with a tendency towards hyper-articulation) and switching to another modality were the most effective means to resolve errors, although in this study errors were not dependent on input modality or style.

Several authors studied error correction in the context of multimodal *dictation* systems. Suhm et al. (2001) investigated whether multimodal

error correction was faster and more accurate than unimodal correction, and whether users learn which modality works best for them. The study shows that multimodal error correction was indeed faster than unimodal correction by re-speaking. Furthermore, they found that ASR accuracy decreased in repeated correction attempts, unless people switched to a different modality. The explanation is that in spoken repairs (repetitions) people tend to hyper-articulate. It is well known that hyper-articulate speech deteriorates the accuracy of automatic speech recognition, because it increases the mismatch between the spoken input and the acoustic models of the ASR system (Oviatt et al., 1996; Levow, 1998). In spite of the fact that the speech recognition accuracy decreases substantially in correction attempts, Suhm et al. (2001) saw that users initially preferred speech for error correction. However, they learned to switch to the most efficient modality when evidence that certain modalities are less effective than others accumulated, which is in line with what Oviatt and VanGent (1996) found. Thus, recognition accuracy had a significant influence on the users' choice between modalities. Karat et al. (1999) and Halverson et al. (1999) also studied the efficiency and effectiveness of error correction facilities in dictation systems (re-dictation, spelling, and choosing from a list of alternatives). They found that subjects stick to re-dictation to correct errors in spite of decreased recognition accuracy. Spelling and choosing from a list of alternatives were used much less often. In contrast to Oviatt and VanGent, Karat et al. and Halverson et al. found that subjects stayed in the speech mode just as often as switching to the keyboard mode to correct errors. Larson and Mowatt (2003) studied the use of four error correction mechanisms used in commercial dictation systems on a Tablet PC in order to examine users' preferences for and combinations of error correction methods. Again, re-dictation turned out to be the correction method that was used most frequently. Subjects also liked this method best. The initial effectiveness of the alternatives list was only around 50%. Efforts to make the use of the alternatives list easier, such as making

it easier to access and dismiss the alternatives list and improving its accuracy, resulted in a large increase in its frequency of use compared to what was observed by Karat et al. (1999). Eventually, the most popular repair strategy was to try the alternates list first and then switch to re-dictation if the correct alternative would not be in the list.

The goal of the research described in this article is to investigate how subjects handle speech recognition errors in a *fully implemented* speech-centric multimodal *form-filling* interface to a service that is *routinely used* by the subjects with a desktop GUI interface. This guarantees that there is no confounding between learning to use the interface and learning to use the service. Although error correction has been studied before, many of these studies were based on dictation systems (which may pose learning problems in their own right) or Wizard-of-Oz evaluations (which may 'handle' speech recognition errors differently than a real speech recognition system). Furthermore, whereas in most studies the correction facilities were limited to re-speaking, choosing from a list of alternatives or typing on a soft-keyboard, we investigate a novel method to correct speech recognition errors, which we expect to be more efficient and more effective in our application than the conventional methods. The factual questions that we aim to answer with this research are: Which of the error correction methods that can be offered in a multimodal form-filling interface is most effective and efficient? What strategies are applied by users when they try to correct a speech recognition error? And what is the effect of different error correction mechanisms on user satisfaction? Based on the answers to these factual questions we hope to be able to formulate guidelines for the design of multimodal form-filling applications.

The remainder of this article is organized as follows: in Section 2 we first describe the interface that was used and the error correction facilities that are offered in this interface, including the novel error correction method that we propose. We continue this section by providing information about the subjects, their tasks and the evaluation measures that were used. Section 3 presents and discusses the results for the objective measures,

the observed user strategies and the user satisfaction scores. In Section 4 we conclude with a general discussion and we summarize the most important conclusions and we give some recommendations for the design of multimodal form-filling applications.

## 2. Methods

### 2.1. The multimodal interface

The multimodal interface that is used in this study provides timetable information for railway connections in the Netherlands. In order to retrieve a travel advice for a specific trip, users must provide five information items: the departure and destination stations, the date, the time, and a switch indicating whether the information database should be queried with an estimated arrival or departure time.

The speech driven interface (cf. Fig. 1) is adapted from the web-based GUI service offered by the Dutch Railways (Sturm et al., 2001, 2002a,b). The ASR system replaces the keyboard, while a pointing device offers the point-and-click



Fig. 1. Screen shot of the multimodal form-filling interface (translated to English).

functions of the mouse. This functionality makes that the interface is representative of many multimodal pilot services for PDAs.

The interface has a so-called tap-and-talk functionality: to enter a value for a specific field, the user must tap the microphone button associated with this field, after which one can speak the corresponding value (Huang et al., 2001). Once the recorded utterance has been processed by the speech recognizer, the recognized value is shown on the screen. If nothing could be recognized, “???” is displayed in the field. Besides feedback of the recognition result, the graphical interface provides information about the system status, by coloring the microphone buttons green when the microphone is open and by showing an hour glass when the system is busy recognizing speech. Several values (i.e. *today/tomorrow* and *departure/arrival*) can be provided without using speech, by tapping one of the radio buttons. When the form is complete, pressing the Search button forces the system to query the database for a travel advice. The resulting travel advice is shown on the screen.

Although the interface has been devised for use on small mobile devices, such as a PDA, in this study it was simulated on a desktop PC. Pen input could be provided by pushing buttons on a touch screen; speech input could be provided through a head-mounted close-talking microphone.

In this study, the interface is tested in two conditions, which differ only in the error correction facilities they offer. This will be explained in more detail in the next subsection.

### 2.2. Error correction facilities

The interface in our experiment offers several facilities to correct speech recognition errors. In this study we focus on correction facilities for the departure and destination station names; most speech recognition errors occur in these concepts, as these are the ones with the largest vocabularies. Since recognition accuracies for dates and times (the other two fields for which speech is required) were quite high, we did not take special measures to facilitate error correction for these fields. Moreover, the correction facilities that our interface offers for the station names are probably sub-

optimal for correcting dates or times; these can best be corrected using graphical methods such as clicking on a calendar or clock.

For the station names the interface offers the following possibilities to correct recognition errors:

1. *Re-speaking*: The user may identify an error by pressing the microphone button associated with the field in which the error occurred. Pressing the microphone button causes the field to be emptied, after which the user can speak the value once again.
2. *Alternatives list*: Users may also correct recognition errors by selecting the correct word from a list of alternatives. The speech recognizer that is used in our system delivers an N-best list. The application system augments this list with all stations that may be present in the city that was recognized as first best. For example the alternatives list of ‘Venlo’ may contain ‘Vleuten’ and ‘Vught’ as recognition alternatives, but it will also contain ‘Venlo Blerick’, which is another station in the city of Venlo. The resulting alternatives list can be accessed in the form of a drop-down window by tapping the down arrow that appears in the right hand corner of the field that shows the first best recognition result (cf. Figs. 1 and 2). The alternatives list will be useless if the ASR output consisted of the first best only, for a city with only a single station. Also, the alternatives list does not necessarily contain the correct value. The alternatives list is also empty when the ASR system was not able to recognize anything, i.e. when ‘???’ appears in the field.



Fig. 2. Screen shot of a dropdown list with alternatives.

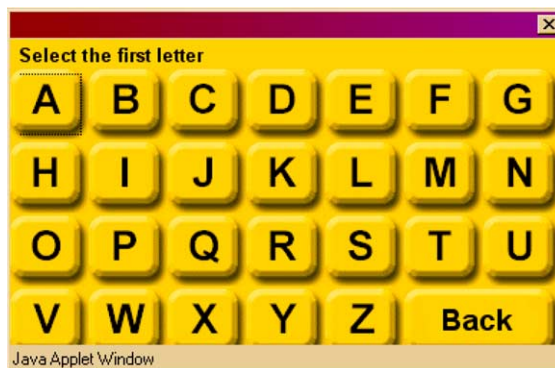


Fig. 3. Screen shot of the soft-keyboard.

3. *Keyboard*: This error correction mechanism is a combination of a soft-keyboard and adaptation of the active recognition lexicon. If an utterance is recognized incorrectly, the user can invoke a soft-keyboard (cf. Fig. 3) by pushing a special button on the screen. Note that Figs. 1 and 2 show the interface without this keyboard button. From the soft-keyboard one may select the first letter of the target word by tapping it. Selecting the first letter causes the speech recognizer to re-process the previous utterance, but with a lexicon and language model that contain only words starting with the selected letter. The Keyboard option does not require the user to speak the value again: recognition is done on the previously recorded utterance. The keyboard buttons only become active after a station name has been spoken, and once the first letter has been selected, it disappears immediately; it is therefore not possible to input values by typing on the soft-keyboard.

Using the Keyboard correction facility, confusions between acoustically similar station names, such as /Baarn/ and /Maarn/ or /Zwolle/ and /Swalmen/ can be solved in a way that may be more effective than re-speaking and perhaps also selecting from an alternatives list.

It is important to note that none of the correction facilities described above guarantees success. It may therefore happen that the user is not able to convey the correct information, in which case

a subject may decide to end the interaction without having obtained the desired information.

In this study the interface is used in two conditions. In one condition, the interface provides only re-speaking and the alternatives list as error correction facilities. This will be referred to as the LIST condition. In the other condition, the interface provides the Keyboard technique as an additional error correction facility. This will be referred to as the LETTER condition. The LETTER interface looks the same as the LIST interface, except for the two additional buttons showing a keyboard, which can be used to invoke the soft-keyboard. These keyboard buttons are placed next to the microphone buttons.

### 2.3. *Subjects, tasks and evaluation measures*

Twenty four subjects took part in an evaluation with a within-subjects design, in which each subject tested both interfaces. To avoid order effects, the subjects were divided into two groups; each group tested the two interfaces in a different order. The subjects (15 male and 9 female) represent different ages (from 19 to 58) and different education levels. All subjects had experience with computers and half of the subjects had used a touch screen before. In order to get timetable information, most subjects consult the official website of the Dutch Railways. Only a few subjects had previous experience with the telephone based spoken dialogue system of the Dutch Railways or with any other spoken dialogue system.

Each session started with a short instruction in which the subject was told that the goal of the experiment was to investigate error correction in speech-centric multimodal information systems. During the instruction, the subjects were shown how the interface could be started and they were instructed how to operate it. After carrying out one practice task, the subjects were asked to obtain travel information about six trips both with the LIST interface and with the LETTER interface. In order to make sure that recognition errors would occur, several tasks contained station names that have a high confusability with other words in the vocabulary and therefore are hard to recognize, such as Tegelen, IJlst, Coevorden,

Echt, etc. Table 1 shows the six tasks that were used for each system.<sup>1</sup>

After completing a series of tasks with one of the two interfaces, subjects were asked to fill out a questionnaire consisting of 20 Likert-scale statements about different aspects of the interface, such as “I thought the interface was efficient” and “I felt in control when using the interface”. The full questionnaire is given in Appendix A. Finally, after having tested both interfaces, subjects filled out a small comparative questionnaire, which was used to measure user preferences.

Speech and pointing actions of all interactions were automatically logged, including time stamps. All speech data were orthographically transcribed. The transcribed dialogue loggings were used to calculate objective performance measures, such as the effectiveness and efficiency of the different error correction facilities, and to analyze user preferences for error correction facilities and the strategies users apply for correcting errors. Dialogue duration was measured in seconds from the start of the first user action (clicking on a microphone or radio button) until the moment the *Search* button is pressed. Speech recognition performance was measured in terms of recognition error rate, which we define as the percentage of relevant information items that were recognized incorrectly. The length of the correction sub-dialogues was measured as the number of user turns dedicated to correcting one error. Here, a user turn is defined as a set of user actions aimed at providing one value (clicking on the microphone button and speaking, pulling down the drop-down menu and choosing one of the values, etc.). The frequency of occurrence was defined as the number of times subjects chose a specific error correction facility, and the effectiveness of the three error correction facilities was measured in terms of the percentage

---

<sup>1</sup> Station names were carefully selected to be equally error prone based on ASR output obtained with a large corpus of relevant speech recordings. However, in actual practice it appeared that some names caused considerably more recognition errors than others. To circumvent this effect, it would have been better to balance station names between the two conditions. However, we are confident that this methodological weakness does not affect the validity of the results of the experiment.

Table 1  
Description of tasks

|         | From       | To         |             | Date     | Time     |
|---------|------------|------------|-------------|----------|----------|
| LIST1   | Maastricht | Breda      |             | Tomorrow | 2:30 pm  |
| LIST2   | Delft      | Tegelen    |             | Tomorrow | 10:30 am |
| LIST3   | Swalmen    | Tilburg    | West        | Thursday | 8:00 am  |
| LIST4   | Den Helder | Rotterdam  | Noord       | Tonight  | 7:00 pm  |
| LIST5   | Zwolle     | Warffum    |             | Sunday   | 8:30 am  |
| LIST6   | Coevorden  | Almelo     |             | Tonight  | 10:00 pm |
| LETTER1 | Rotterdam  | Leeuwarden |             | Tomorrow | 8:30 am  |
| LETTER2 | Roosendaal | Echt       |             | Sunday   | 1:30 pm  |
| LETTER3 | Obdam      | Breda      | Prinsenbeek | Today    | 5:00 pm  |
| LETTER4 | Venlo      | Utrecht    | Lunetten    | Tonight  | 11:00 pm |
| LETTER5 | Ommen      | IJlst      |             | Tomorrow | 4:30 pm  |
| LETTER6 | Nijmegen   | Culemborg  |             | Thursday | 7:00 am  |

of correction attempts in which the error was solved. User satisfaction was measured by means of the Likert-scale scores and user preferences were measured from the preferences questionnaire.

### 3. Results and discussion

In this section we first present and discuss the data about error correction, such as how many errors were made, how many of these errors could be solved and in what way they were solved. Table 2 gives a summary of all objective performance data. We first discuss the data in Table 2; in doing so, we will defer the general interaction data in the first panel until the end, because they are easier to explain using the information in the subsequent panels. We then proceed with a discussion of the interaction patterns the subjects applied after which we present the user satisfaction data. The error correction and interaction details are based on the station names only, since these are the only fields for which multiple correction facilities were provided (see Section 2.2).

#### 3.1. Error correction

All 24 subjects carried out six tasks with each of the two interfaces, yielding a total of 144 interactions for each interface. With the LIST interface 106 interactions (74.3%) were completed successfully (i.e. subjects obtained the desired travel ad-

Table 2  
Overview of objective measures

|  | LIST        | LETTER      |
|--|-------------|-------------|
| <i>General interaction data</i>                            |             |             |
| Total number of interactions                               | 144         | 144         |
| Number of successful interactions                          | 106         | 139         |
| Mean duration of successful interactions (s)               | 44.4        | 50.2        |
| Mean number of turns in successful interactions            | 7.6         | 8.0         |
| Mean overall recognition error rate                        | 43.5        | 31.7        |
| <i>Error correction</i>                                    |             |             |
| Total number of station names                              | 281         | 287         |
| Total number of error correction dialogues (station names) | 96 (34.2%)  | 122 (42.5%) |
| Total number of errors solved                              | 59 (60.6%)  | 118 (96.7%) |
| Mean number of user turns in error correction dialogues    | 3.7         | 2.3         |
| <i>Effectiveness</i>                                       |             |             |
| Re-speak   | 16.6%       | 11.2%       |
| Alternatives list  | 16.0%       | 36.2%       |
| Keyboard   | n/a         | 76.4%       |
| <i>Usage frequencies</i>                                   |             |             |
| Total # re-speak (for correction only)                     | 277 (78.7%) | 89 (33.7%)  |
| Total # alternatives list                                  | 75 (21.3%)  | 65 (24.6%)  |
| Total # keyboard   | n/a         | 110 (41.7%) |

vice); with the LETTER interface 139 interactions (96.5%) were completed successfully. A *t*-test showed that this difference is significant ( $t(23) = 6.78, p < .01$ ). Two interactions in the

LETTER condition failed due to problems in entering information other than the station names; all other failures in both conditions were due to problems with the station names.

The LIST interactions contained a total of 281 station names; the LETTER interactions contained 287 station names. In principle, 144 interactions with two station names each would yield 288 station names; the difference is caused by the fact that in a number of dialogues the subject gave up before both station names had been entered, because of persistent recognition errors.

Table 2 shows that more station names were recognized incorrectly in the first attempt to enter them in the LETTER interface than in the LIST interface (42.5% and 34.2%, respectively). This difference is significant ( $t(23) = 3.15, p < .01$ ). Obviously, this leads to the conclusion that the station names that were used in the tasks were not equally hard for the recognizer, despite the fact that they showed the same performance in off-line tests. The fact that the recognition error rate is very high to begin with is not surprising: the station names were deliberately chosen to cause a large proportion of errors, so as to facilitate our investigation of error correction procedures.

While the absolute number of error correction dialogues in the LETTER condition was higher,

it appeared that the average number of turns to correct them was lower than in the LIST condition. Fig. 4 shows that the majority of the error correction dialogues in the LETTER condition require only one or two user actions, whereas in the LIST condition the proportion of error correction dialogues that require more than two user actions is much larger. In the LIST condition correcting an error took 3.7 turns on average, with the LETTER interface the average number of correction turns is significantly shorter: about 2.3 turns ( $t(23) = -3.62, p < .01$ ).

Error correction using the LETTER interface was also more *effective* than error correction using the LIST interface: 118 of the errors could be solved in the LETTER interface (97%), whereas in the LIST interface only 59 errors could be solved (61%). This difference is significant ( $t(23) = -6.52, p < .01$ ). Obviously, the Keyboard correction facility in the LETTER interface enabled the users to solve errors that could not be solved using the Re-speak method or by choosing from a list of alternatives that were available in the LIST condition.

In the LIST condition 79% of all correction attempts were done using the Re-speak method, whereas the Alternatives list was used in only 21% of the cases. The frequencies of use of the

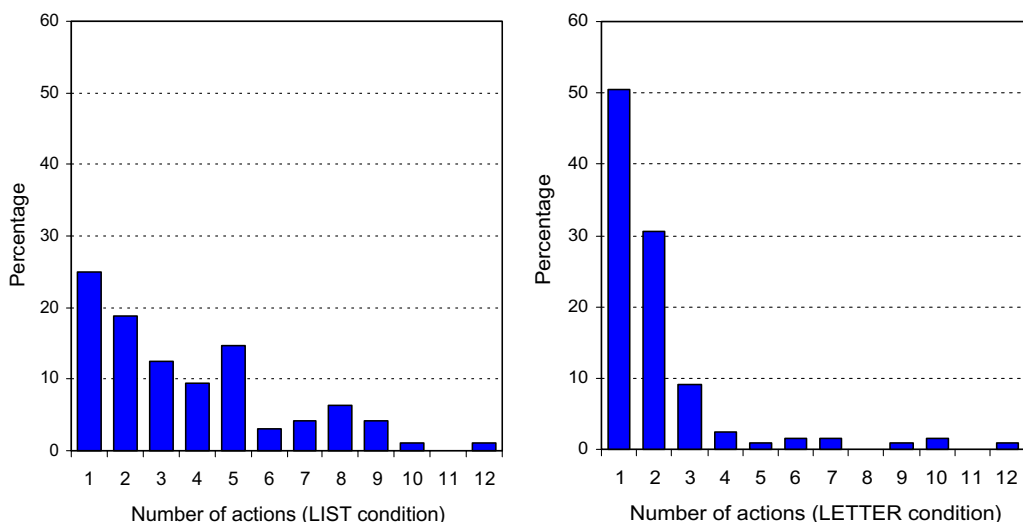


Fig. 4. Histogram of the number of user actions in error correction subdialogues.

two methods deviate significantly from the assumption that they are equal ( $\chi^2 = 11.52$ ,  $p < .01$ ). The main explanation for the low use of the Alternatives list is that in about half of the cases the Alternatives list was empty, because the ASR system recognized something else than a station name or nothing at all (resulting in the “???” display, without the down error to open a drop-down list). In this situation, Re-speak was the only available option. The Alternatives list was used in 75% of the cases where it was actually available, whereas the Re-speak method was preferred only in 25% of these cases. This suggests that our subjects did not default to Re-speaking as the preferred error correction strategy, simply because staying in the speech mode would require least cognitive effort.

Interactions with the LETTER interface show a different tendency. Here, subjects tried to correct most errors using the Keyboard (42%). In 34% of the errors in the LETTER condition subjects tried to correct them using the Re-speak option and in 25% of the cases the Alternatives list was tried. The differences are significant ( $\chi^2 = 11.01$ ,  $p < .01$ ). It should be noted however, that the Keyboard facility can only be used if the recognized word starts with a different letter than the target word; else re-recognition with a restricted vocabulary and language model will not help. Analysis of the recognition results showed that 93% of the errors started with a different letter than the target word. The Keyboard was used in 52% of these cases. Furthermore, again the Alternatives list was only available in about 50% of the errors. Subjects used the Alternatives list in 54% of the cases where it was actually available. Thus, the data in the bottom panel of Table 2 tell only part of the full story. The preference for the Keyboard option that appears from the raw data must be qualified, because it is inflated by the fact that its competitor (the Alternatives list) was often not available as a result of the operation of the ASR system. When subjects had the choice between Keyboard and Alternatives, they used Alternatives twice as often as Keyboard. There was no confounding effect of the order in which the interfaces were used. The preference for the Alternatives list -if available- was equally evident in both groups.

In the LETTER condition, the Keyboard option turned out to be by far the most *effective* method to correct errors: 76.4% of the correction attempts for which the Keyboard was used were successful. The effectiveness of the Alternatives list in the LETTER condition was 36.2%. The Re-speak option is least effective (11.2%). In line with studies by Suhm et al. (2001) and Levow (1998), we found that the ASR performance in terms of recognition error rate dropped substantially from 71% for the first attempt to 29% for the correction attempts. The explanation for this observation is twofold. First of all, the set of values that need to be corrected is obviously very much biased towards the difficult station names, since these are the ones that were misrecognized in the first place. Furthermore, recognition of repetitions is harder, because people tend to hyper-articulate if they have to repeat the same value and cannot resort to changes in wording (which is virtually impossible in the case of station names) (Oviatt et al., 1996). This strongly suggests that there is a large premium to be had if multimodal form-filling interfaces can be designed in such a way that it is natural for the users to avoid the Re-speak method for correcting errors.

In the LIST condition the two available error correction facilities, the Alternatives list and the Re-speak method, were equally effective (16.0% and 16.6%, respectively). We have not been able to find an explanation for the difference between the effectiveness of the Alternatives list in the LIST and LETTER conditions.

The scene is now set for a discussion of the data in the top panel of Table 2. The average duration of a successfully completed interaction was 44.4 s for the LIST interface and 50.2 s for the LETTER interface. This difference is significant ( $t(23) = 2.28$ ,  $p < .05$ ). The main explanation for the observed difference in efficiency is the fact that the average duration is based on the successfully completed dialogues only. The longer dialogues, those in which many recognition errors had to be corrected, more often ended successfully in the LETTER condition than in the LIST condition, which has caused a higher average number of turns in the LETTER condition than in the LIST condition (8.0 versus 7.6 ( $t(23) = 1.61$ ; n.s.)).

The higher mean overall recognition error rate in the LIST condition (43.5% vs. 31.7% for the LETTER condition) is due to the higher number of Re-speak attempts in the LIST condition, where the Keyboard option was not available. Thus, repeated failures to correct errors by Re-speaking in the LIST condition appear to outnumber the larger number of initial ASR errors for station names in the LETTER condition.

### 3.2. Interaction patterns

Observing the behavior of subjects in the case of errors across several correction turns, we found that, despite the fact that there were large differences between the subjects, a number of interaction patterns can be defined that are shared by groups of subjects. These interaction patterns are summarized in Table 3 (LIST condition) and Table 4 (LETTER condition) for the first two turns in the correction sub dialogues.

Both in the LETTER and in the LIST condition a large group of subjects had a preference for the Re-speak method in their first correction attempt. If their attempt to correct the error using the Re-speak method failed, in the LETTER condition subjects would switch from the speech mode to the Keyboard option. Re-speak and the Alternatives list were both used far less in the second correction attempt. Another group of subjects immediately switched to the Keyboard option for the first correction attempt; in this case, because of the high effectiveness of the Keyboard option,

Table 3  
Observed error correction strategies in the LIST condition

| Ist attempt                                | Ist correction attempt | 2nd correction attempt | No. of occurrences |
|--|------------------------|------------------------|--------------------|
| Speak                                      | Re-speak               | –                      | 19                 |
| Speak                                      | Re-speak               | Re-speak               | 44                 |
| Speak                                      | Re-speak               | Alternatives list      | 15 (78 total)      |
| Speak                                      | Alternatives list      | –                      | 5                  |
| Speak                                      | Alternatives list      | Re-speak               | 13                 |
| Speak                                      | Alternatives list      | Alternatives list      | 0 (18 total)       |
| Total number of error correction dialogues |                        |                        | 96                 |

Table 4  
Observed error correction strategies in the LETTER condition

| Ist attempt                                | Ist correction attempt | 2nd correction attempt | No. of occurrences |
|--|------------------------|------------------------|--------------------|
| Speak                                      | Re-speak               | –                      | 8                  |
| Speak                                      | Re-speak               | Keyboard               | 27                 |
| Speak                                      | Re-speak               | Alternatives list      | 9                  |
| Speak                                      | Re-speak               | Re-speak               | 8 (52 total)       |
| Speak                                      | Keyboard               | –                      | 38                 |
| Speak                                      | Keyboard               | Alternatives list      | 3                  |
| Speak                                      | Keyboard               | Re-speak               | 0                  |
| Speak                                      | Keyboard               | Keyboard               | 0 (41 total)       |
| Speak                                      | Alternatives list      | –                      | 15                 |
| Speak                                      | Alternatives list      | Keyboard               | 8                  |
| Speak                                      | Alternatives list      | Re-speak               | 6                  |
| Speak                                      | Alternatives list      | Alternatives list      | 0 (29 total)       |
| Total number of error correction dialogues |                        |                        | 122                |

most errors would be solved right away. The Alternatives list was used least often, for the reason explained above. It is interesting to note that, although Re-speaking was used more often than either Keyboard or Alternatives list in the LETTER condition, it still was used in less than half the cases where errors had to be corrected. Thus, it appears that the tendency for subjects to stick to speaking for correcting errors in a form-filling application is not particularly strong if other options are available.

In the LIST condition, the Re-speak method was most popular as well for the first correction attempt; the alternatives list was used far less. Obviously, as mentioned in the previous section, in 50% of the cases, subjects were compelled to re-speak the value because there was no alternatives list. Whereas in the LETTER condition most subjects would switch to the Keyboard option if re-speaking the value had not solved the error, in the LIST condition most subjects would stick to the Re-speak option in the second attempt. Obviously, the preference for Re-speaking is strongly conditioned on the availability of alternative ways to correct ASR errors.

We found that even though there was only a limited amount of time to learn (subjects carried out only six tasks in each of the two conditions) subjects adjusted their interaction behavior over time. Whereas in the first tasks the user preferences

for the three options were fairly equal, especially for the last two tasks we observed that these proportions shifted in favor of the Keyboard method in the LETTER condition. So, instead of staying in the speech mode, in the later dialogues subjects tended to switch to the most effective option immediately. Apparently, it does not take much time to learn how to correct errors in the most effective and efficient way. We observed the same behavior in a more extensive study of the effect that experience has on the way users interact with a multimodal interface (Sturm et al., 2002b). Although in that study we did not address error correction specifically, we found that users need some time to find out what the optimal way is to operate the system, and that their behavior evolves towards the most efficient way.

### 3.3. User satisfaction

Almost all Likert-scale judgments were more positive for the LETTER interface than for the LIST interface. Table 5 shows the mean scores of the ten statements for which a Wilcoxon

Signed Ranks Test for related samples found significant differences between the LIST and the LETTER condition. The complete list of statements and associated scores can be found in Appendix A.

For each of these statements, the LETTER interface was judged more positively than the LIST interface. Subjects found the LETTER system to be more efficient, they felt more in control and they found this interface less frustrating. The combination of speech and pen input was also considered to be more useful in the LETTER system. In line with the objective performance data, almost all subjects indicated that correcting errors was easier with the LETTER interface than with the LIST interface. The LIST interface needed more improvements than the LETTER interface. Consequently, subjects liked the LETTER interface better and would be happier to use it again than the LIST interface. Still, for both interfaces subjects indicated that they would prefer to use the PC travel planner, although they would prefer a human operator only to the LIST interface.

The higher user satisfaction for the LETTER interface is confirmed by the user preferences.

Table 5  
Mean Likert scale scores (scales from 1 to 5; high scores denote agreement with the statement, low scores denote disagreement)

| Statement   | LIST | LETTER | Significance          |
|---|------|--------|-----------------------|
| I thought the service was efficient                                   | 2.9  | 3.7    | $z = -2.976, p < .05$ |
| I thought that it took too long to get the information                | 2.8  | 2.4    | $z = -2.586, p < .05$ |
| I felt in control while using the service                             | 3.3  | 4.2    | $z = -3.331, p < .05$ |
| I felt frustrated when using the service                              | 2.8  | 1.9    | $z = -2.825, p < .05$ |
| I found it useful that I could use speech as well as the touch screen | 3.3  | 4.0    | $z = -2.559, p < .05$ |
| I enjoyed using the service   | 3.4  | 4.1    | $z = -2.970, p < .05$ |
| I would be happy to use the service again                             | 2.6  | 3.8    | $z = -3.581, p < .05$ |
| I would prefer to use the PC version of the travel information system | 4.4  | 3.6    | $z = -3.087, p < .05$ |
| I would prefer to speak to a human operator                           | 3.4  | 2.3    | $z = -3.358, p < .05$ |
| I feel the service needs a lot of improvement                         | 3.6  | 2.8    | $z = -2.584, p < .05$ |

Table 6  
User preferences

| Question   | LIST (%) | LETTER (%) | No preference (%) |
|--|----------|------------|-------------------|
| 1. Which system do you think was easiest to use?       | 0        | 75         | 25                |
| 2. With which system is correcting errors easier?      | 0        | 96         | 4                 |
| 3. Which system do you think was most fun to use?      | 4        | 67         | 29                |
| 4. Which system would you prefer to use in the future? | 4        | 88         | 8                 |

Table 6 shows the results of the four preferences questions.

As can be seen, most subjects preferred the LETTER interface on all four aspects, although a number of subjects thought there was no difference between the two conditions regarding the ease of use and the fun to use it. Almost all subjects considered correcting errors to be easiest using the LETTER interface. It is of some interest to note that the higher proportion of recognition errors for station names (cf. Table 2) in the LETTER interface had no impact on the subjects' preference. There are two possible explanations: either subjects do not experience the difference between 42% and 34% error rate as 'significant', because both are annoyingly high; or the higher frustration due to errors in the LETTER interface is completely compensated by the superior error correction facilities.

### 3.4. *Correlation between objective performance data and user satisfaction judgments*

We carried out a Stepwise Multiple Regression Analysis to investigate to what extent the objective performance data explain the user satisfaction judgments. The results of this analysis show that in both interfaces the proportion of successfully completed tasks and the number of turns spent in correcting errors are the most important predictors. Interestingly, the proportion of explained variance was consistently higher for the LIST interface. However, in no case could more than approximately 45% of the variance in the judgments be explained by the objective measures. The limited success of the regression analysis should not come as a surprise. In their analysis of a unimodal speech driven service Walker and Passonneau (2001) found similar cumulative predictive power for a range of objective measures. In multimodal interaction, where the number of different ways to reach the same goal is substantially larger, increasing the degree of idiosyncrasy in the objective measures, one would only expect that the power of objective measures to explain subjective judgments should be weaker.

## 4. General discussion and guidelines for designing multimodal form-filling applications

The goal of the research described in this article was to determine how subjects handle speech recognition errors in a multimodal form-filling interface, depending on the availability of alternative options, in a fully implemented service that all subjects routinely use via a desktop GUI application. Besides well-known error correction mechanisms, such as re-speaking and choosing from a list of alternatives, the interface offers a novel hybrid correction mechanism in which the user taps the first letter of the target word on a 'soft-keyboard', after which the system re-processes the initial spoken input with a restricted language model and lexicon. The effectiveness and efficiency of different error correction mechanisms, the error correction strategies that were applied and the effects on user satisfaction were studied in an evaluation in which the interface was tested in two conditions: in one condition (LIST), the interface only provides re-speaking and the alternatives list as error correction facilities. In the other condition (LETTER), the interface provides the soft-keyboard technique as an additional error correction facility.

The Keyboard option that was available in the LETTER condition proved to be by far the most effective way to correct errors: in 76% of all corrections using the Keyboard option the error was solved. In our implementation, however, if the station name that was spoken is recognized as a different station name starting with the same letter, the error could not be corrected by choosing the first letter and re-recognizing the utterance with a smaller lexicon. The effectiveness of the Keyboard option could be improved further if in this case the station name that was recognized in the first place would be removed from the lexicon. This can be done safely, since the user has already indicated the word to be wrong (Ainsworth and Pratt, 1992). In this context, it must be noted that the Keyboard option has been successfully applied in experiments with a multimodal Routefinder application in which the lexicon was about ten times bigger than the lexicon that was used in our experiments (Niklfeld et al., 2002). To reap the full advantage of the restriction of the lexicon, one

needs an ASR system that can dynamically adapt its lexicon and language model on the fly. In our test system dynamic adaptation of the lexicon was simulated by running 27 ASR systems in parallel, one with the full lexicon, and 26 with reduced lexicons.

The Alternatives list was much less effective than the Keyboard method: only 16% of the attempts to solve an error using the Alternatives list were successful in the LIST interface versus 36% in the LETTER interface. Moreover, the Alternatives list was only available in about 50% of the erroneously recognized station names. Nevertheless, in those situations where it could be applied it was actually used in 54% (LETTER condition) or 75% (LIST condition) of the cases. In fact, in these situations the Alternatives list was used more often than the Keyboard option, which had a much higher effectiveness.

Our results support the findings of Oviatt and VanGent (1996) and Levow (1998) that spoken repetition is not a very effective way of repair: only between 17% (LIST condition) and 11% (LETTER condition) of the attempts to correct an error by repeating the value were successful. Although it is certainly true that the effectiveness of the Re-speak method for error correction was affected by the intrinsic difficulty of recognizing confusable station names, informal observations confirmed that there is probably also a substantial adverse effect of hyper-articulation. This is difficult to avoid, since people know that speaking more clearly helps to solve speech understanding problems in human–human communication. However, it is well known that hyper-articulation has a negative effect on ASR performance. Therefore, it seems that interaction designers should strive towards multimodal interfaces that encourage other error correction methods than re-speaking.

In accordance with what Suhm et al. (2001), Karat et al. (2000) and Halverson et al. (1999) found, our results indicate that *initially* users tend to stay in the speech mode: most subjects started with a preference for the Re-speak option and only switched to another modality after they had experienced that repetition was not likely to correct the error. However, we also found that as subjects gained experience, they quickly learned to choose

the most effective option (Keyboard) immediately. This result shows that previous suggestions that users have a strong preference to stay in the speech mode, because switching to an alternative mode requires an additional cognitive effort, must be qualified. From our results it seems more likely that users learn quite rapidly that re-speaking is not very effective, so that alternative error correction techniques become more attractive, provided that they are obvious and easy to use. Thus, the preference for error correction facilities seems to depend more on their effectiveness and efficiency, than on a tendency towards ‘cognitive laziness’. This only increases the importance of interface design that promotes the use of other error correction methods than re-speaking. Admittedly, our finding that subjects rapidly learn to avoid re-speaking if other options are available may have been caused in part by the fact that the service used to test error correction facilities did not make large demands on cognitive processing: all subjects knew how to use the train timetable information service. In experiments where subjects have to learn the application at the same time as the interface, the need for focusing cognitive resources on the application, at the cost of selecting the most appropriate error correction facility, may be more compelling.

If re-speaking for error correction cannot be avoided, or if users prefer this method for whatever reason, both the effectiveness and efficiency of error correction can be improved substantially if the ASR system is able to dynamically generate lexicons and language models, and if that capability is used to its fullest extent by the dialogue manager of a multimodal service. If the lexicon can be adapted on the fly, it should be easy to avoid repeating the same recognition errors. The very large proportion of recognition errors that was corrected in the first attempt with the Keyboard option suggests that it is worthwhile considering a design in which users can indicate the first letter of the word before they speak a name in the first place.

User satisfaction judgments were generally higher for the LETTER interface than for the LIST interface. In a way, the higher satisfaction scores for the LETTER interface were predictable,

because this interface offered an additional error correction facility. However, both the objective measures and the evaluation scores confirm that the option to indicate the first character of a name is easy to learn and to use for the subjects and at the same time a very powerful means to improve recognition performance. It is interesting to note that even in the LETTER condition subjects say that they would rather use the PC Travel Planner than the multimodal service. This is probably due to the fact that our subjects had much more experience in using that service. However, it must also be acknowledged that a PC with a full keyboard (and a large screen) is inherently more suitable for form-filling information services than a keyboardless PDA with a small screen. Therefore, prospective providers of mobile services designed for PDAs should not expect that users will readily abandon old-fashioned PC services in favor of PDAs.

A final note must be made on the distinction between modes/modalities, their functionality in a service, and the procedures that are available to users for combining modalities and functionalities. Both interfaces tested in our experiment offer the

same modalities, viz. a combination of speech and pen for input and text and graphics for output. Yet, the Keyboard option offers very powerful additional functionality for correcting errors. Thus, if speech, pen and graphics are available, finding clever combinations for exploiting these modalities in specific conditions of a dialogue in a form-filling service is much more promising than considering the addition of new modes. But at the same time it remains true that there is still much to learn about how speech, pen and graphics can and should be combined, and how the decisions vdepend on specific characteristics of an application.

### Acknowledgement

The research reported here was funded by the European Commission as part of the FP5 IST project SMADA (IST-1999-10667).

The authors would like to thank several members of our research group for useful comments on earlier versions of this article. Furthermore, we are grateful to Gies Bouwman for implement-

Table 7

User satisfaction questionnaire (on a scale from 1 to 5, high scores denote agreement with the statement, low scores denote disagreement)

| Statement   | LIST | LETTER |
|---|------|--------|
| 1. I had to concentrate hard while using the service                      | 3.5  | 3.3    |
| 2. When I was using the service, I always knew what I was expected to do  | 4.5  | 4.4    |
| 3. The service was easy to use  | 4.1  | 4.3    |
| 4. I would be happy to use the service again                              | 2.6  | 3.8    |
| 5. I thought the service was efficient                                    | 2.9  | 3.7    |
| 6. I thought that it took too long to get the information                 | 2.8  | 2.4    |
| 7. I found the service confusing to use                                   | 2.0  | 1.8    |
| 8. I felt frustrated when using the service                               | 2.8  | 1.9    |
| 9. I thought the messages were easy to follow                             | 4.3  | 4.3    |
| 10. I felt in control while using the service                             | 3.3  | 4.2    |
| 11. I would prefer to use the PC version of the travel information system | 4.4  | 3.6    |
| 12. The service was too fast for me                                       | 1.2  | 1.2    |
| 13. I felt under stress using the service                                 | 2.1  | 2.0    |
| 14. I thought that the service was too complicated                        | 1.4  | 1.5    |
| 15. I enjoyed using the service   | 3.4  | 4.1    |
| 16. I felt that the service was reliable                                  | 3.2  | 3.6    |
| 17. I got flustered when using the service                                | 1.9  | 1.6    |
| 18. I would prefer to speak to a human operator                           | 3.4  | 2.3    |
| 19. I feel that the service needs a lot of improvement                    | 3.6  | 2.8    |
| 20. I found it useful that I could use speech as well as the touch screen | 3.3  | 4.0    |

ing the Keyboard facility used in the LETTER interface.

### Appendix A. User satisfaction questionnaire

The user satisfaction questionnaire used in this study is based on the usability questionnaire developed by the Centre for Communication Interface Research (CCIR), Edinburgh University together with British Telecom (BT) (Love et al., 1994). We adjusted this questionnaire to our situation by leaving out a number of statements that were not applicable and by adding a statement concerning the multimodal aspect of the interfaces. Table 7 shows the mean Likert-scale judgments for each statement both for the LIST interface and for the LETTER interface.

### References

- Ainsworth, W.A., Pratt, S.R., 1992. Feedback strategies for error correction in speech recognition systems. *Internat. J. Man–Machine Studies* 36 (6), 833–842.
- Cohen, P.R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., Clow, J., 1997. QuickSet: Multimodal interaction for distributed applications. In: *Proc. Fifth Annual Internat. ACM Multimedia Conf.* pp. 31–40.
- Halverson, C.A., Horn, D.B., Karat, C.-M., Karat, J., 1999. The beauty of errors: patterns of error correction in desktop speech systems. In: *Proc. INTERACT'99.* pp. 133–140.
- Huang, X.D., Acero, A., Chelba, C., Deng, L., Droppo, J., Duchene, D., Goodman, J., Hon, H., Jacoby, D., Jiang, L., Loynd, R., Mahajan, M., Mau, P., Meredith, S., Mughal, S., Neto, S., Plumpe, M., Steury, K., Venolia, G., Wang, K., Wang, Y., 2001. Mipad: a multimodal interaction prototype. In: *Proc. IEEE Internat. Conf. on Acoustics, Speech, and Signal Process. (ICASSP-01).*
- Johnston, M., Bangalore, S., Vasireddy, G., Stent, A., Ehlen, P., Walker, M., Whittaker, S., Maloor, P., 2002. MATCH: An architecture for multimodal dialogue systems. In: *Proc. 40th Annual Meeting of the Association for Computational Linguistics.*
- Karat, C.-M., Halverson, C.A., Horn, D.B., Karat, J., 1999. Patterns of entry and correction in large vocabulary continuous speech recognition systems. In: *Proc. SIGCHI Conf. on Human Factors in Computing Systems.* pp. 568–575.
- Karat, J., Horn, D.B., Halverson, C.A., Karat, C.-M., 2000. Overcoming unusability: developing efficient strategies in speech recognition systems. In: *CHI '00 Extended Abstracts on Human Factors in Computing Systems.* pp. 141–142.
- Larson, K., Mowatt, D., 2003. Speech error correction: the story of the alternates list. *Internat. J. Speech Technology* 6 (2), 183–194.
- Levow, G.-A., 1998. Characterizing and recognizing spoken corrections in human-computer dialogue. In: *Proc. 36th Annual Meeting of the Association of Computational Linguistics.* pp. 736–742.
- Love, S., Dutton, R.T., Foster, J.C., Jack, M.A., Stentiford, F.W.M., 1994. Identifying salient usability attributes for automated telephone services. In: *Proc. Internat. Conf. on Spoken Language Process. (ICSLP-94).* pp. 1307–1310.
- Mankoff, J., Abowd, G.D., 1999. Error correction techniques for handwriting, speech and other ambiguous or error prone systems. *GVU Technical Report Number: GIT-GVU-99-18.*
- Niklfeld, G., Pucher, M., Finan, R., Eckhart, W., 2002. Steps towards multi-modal data services in GPRS and in UMTS or WLAN networks. In: *Proc. 2nd SIGdial Workshop on Discourse and Dialogue (IDS-02).*
- Oviatt, S., 1999. Mutual disambiguation of recognition errors in a multimodal architecture. In: *Proc. ACM Conf. on Human Factors in Computing Systems (CHI 99).* pp. 576–583.
- Oviatt, S., 2000. Taming speech recognition errors within a multimodal interface. *Comm. ACM* 43 (9), 45–51.
- Oviatt, S., VanGent, R., 1996. Error resolution during multimodal human-computer interaction. In: *Proc. Internat. Conf. on Spoken Language Process. (ICSLP-96).* pp. 204–207.
- Oviatt, S., Levow, G.-A., MacEachern, M., Kuhn, K., 1996. Modeling hyperarticulate speech during human-computer error resolution. In: *Proc. Internat. Conf. on Spoken Language Process. (ICSLP-96).* pp. 801–804.
- Sturm, J., Wang, F., Cranen, B., 2001. Adding extra input/output modalities to a spoken dialogue system. In: *Proc. 2nd SIGdial Workshop on Discourse and Dialogue.*
- Sturm, J., Bakx, I., Cranen, B., Terken, J., Wang, F., 2002a. Usability evaluation of a Dutch multimodal system for train timetable information. In: *Proc. Second Internat. Conf. on Language Resources and Evaluation (LREC2002).*
- Sturm, J., Bakx, I., Cranen, B., Terken, J., Wang, F., 2002b. The effect of prolonged use on multimodal interaction. In: *Proc. ISCA Workshop on Multimodal Interaction in Mobile Environments.*
- Suhm, B., Myers, B., Waibel, A., 2001. Multimodal error correction for speech user interfaces. *ACM Trans. Computer–Human Interaction* 8 (1), 60–98.
- Walker, M.A., Passonneau, R., 2001. DATE: A dialog act tagging scheme for evaluation of spoken dialog systems. In: *Proc. Human Language Technology Conf. (HLT 2001).*
- Zajicek, M., Hewitt, J., 1990. An investigation into the use of error recovery dialogues in a user interface management system for speech recognition. In: *Proc. INTERACT'90.* pp. 755–760.