Pragmatics 4:1.63-79
International Pragmatics Association
DOI: 10.1075/prag.4.1.02che

THE PRAGMATICS OF FRIENDLINESS AND USER-FRIENDLINESS

An investigation of repairs in human-human dialogue and in human-computer dialogue

Christine Cheepen

1. Friendliness and user-friendliness

Throughout this paper the term 'friendliness' is used in its widest possible sense. I do not want to restrict the discussion to cases where only the most positive, supportive, 'chummy' aspect of friendliness is considered. Instead, I shall take the view that 'friendliness' can be equated with 'workability' in human-human dialogue. In other words I shall be focussing on those aspects of dialogue which make a speech encounter successful in terms of being a communicative act - rather than unsuccessful, where the communication channels break down into silence rather than continuing as a series of spoken exchanges which work towards completion of a dialogic goal or goals. Taking this view of friendliness in dialogue as the strategies and tactics which make the encounter work opens up the mechanics of human-human dialogue and allows a profitable set of parallels to be drawn with human-computer dialogue, where 'user-friendliness' is, again, a matter of the 'workability' of a system.

The term 'user-friendliness' is well known and has been part of our vocabulary for many years. It is, though, a possibly misleading term, implying (for the novice computer user) that perhaps the system is simply easy for the novice to approach and use without a great deal of training. Though this is indeed a requirement of user-friendliness, there are other design aspects which must be addressed if the system is to be fully user-friendly - i.e. usable under all circumstances.

Observing some of the major ways in which human interactants monitor, control and guide their dialogue to 'make it work' promotes a more systematic, human-centred approach to the study of human-computer interaction. Modelling the human-computer interface as far as possible on the structures used intuitively by human interactants in face-to-face dialogic encounters ensures that the maximum degree of user-friendliness is achieved, and this in turn ensures that whatever functionality is offered by an automated system is fully accessible and therefore exploitable by the human user.

The connection between human-human spoken dialogue and the human-computer interface is particularly relevant in the case of the automated systems I shall be discussing in this paper. These systems are totally speech driven - that is, they allow input by speech as an alternative to the keyboard. While output is by screen display, all input is by voice, and this means that they share many characteristics with human-human dialogue. One of the most important of these is the dynamic, 'live' nature of the input. Whereas a user who types in commands can see the command he has just typed, this is not the case where the input is by speech. Instead, the user must, as in human-human dialogue, *remember* what he has just said, so that he is properly able to interpret how far the response (from the machine) is relevant and adequate.

In operations which require several different steps the cognitive load on the user can become very high, and the potential for mistakes is greatly increased. At this point the user must have easy and unambiguous access to repair strategies so that progress towards successful completion of the task which is the goal of the human-computer encounter can be successfully restored. Efficient transfer and exploitation of the knowledge and expertise which human interactants exhibit in their face-to-face dealings with one another to the human-computer interface allows efficient repairs to be carried out, and preserves the user's access to the full functionality of the automated system.

2. Human-human dialogue

2.1. *Transaction and interaction*

Brown and Yule divide spoken dialogue into two major categories - *transaction* and *interaction*. Transactions are those encounters which can be thought of as 'task oriented', where the major goal of the encounter is to perform some action on the world *outside the encounter*, e.g. buying a newspaper, negotiating a contract etc. Interactions, on the other hand, are 'person oriented' encounters, where the overall goal is not to make some kind of change in the outside world, but to *promote the relationship between the speakers*, e.g. a chat between friends, or any kind of purely social discourse. For the purposes of this paper I will concentrate on transactional encounters, as any human-computer communication is clearly task oriented, which means that the overall aim is transactional, but as we will see, this does not mean that interactional aspects can be totally ignored.

Within even the most unambiguously transactional encounters, there is a requirement for participants to acknowledge and cater for one another's interactional needs. Only the most urgent and extreme encounters (e.g. a road accident, or a domestic fire) are permitted to operate without an interactional framework, even if that framework be of the most minimal kind (e.g. "please", "thank you" etc.). More usually, transactions have quite elaborate interactional frameworks, and exchanges such as the one below are regarded as perfectly normal:

(1) Good morning, can I help you?
Oh yes, thankyou. I wonder if you could tell me *where I can get a key like this cut.*
Yeah - *the bloke over on that counter'll do it for you.*
Thanks very much. That's wonderful.
OK. Bye.

Note that only the underlined sections in the example are strictly transactional - the rest acts as a kind of interactional framework for the question and answer pair which form the transaction.

Although such an exchange could be stripped of some of its interactional content and still be considered normal, if *all* interaction were removed, leaving only the question and answer, then one or both participants would class their interlocutor as extremely rude, and regard the encounter as in some way deviant. The interactional framework allows the participants to show themselves, while carrying out the transaction which is the primary focus of the encounter, to signal that they are polite and, in the broad sense of the word, *friendly*.

2.2. *Friendliness and repairs*

Tokens of friendliness are typically found at the boundaries of transactions in the form of interactional phases which take place in the opening closing stages of the encounter. Friendliness signals are not, however, restricted to only the opening and closing stages. They also occur frequently throughout transactions (particularly those transactions which are prolonged beyond the simple question and answer pair), and wherever they occur they can be seen functioning to progress the encounter towards its transactional goal.

Some transactions are competitive (or even hostile) in terms of the transactional goal. An example of this would be the kind of encounter where perhaps a customer is complaining to a shop assistant or similar about a service or an item. As far as the transactional goal is concerned, this kind of encounter cannot be considered as a cooperative event. The end result of such competitive dialogue will be that one participant will 'win' and the other will 'lose'. In terms of the *discourse*, however, such speech events are almost always *cooperative* events. That is, the participants work together in order to achieve a successful completion of the encounter and to reach the transactional goal. Encounters which are not cooperative at the discourse level are, in Goffman's terms, 'face threatening' to all participants. A dialogue which is extremely uncooperative at the discourse level will be incoherent, and appear as two competing monologues rather than one dialogue. Such encounters are extremely rare, as speech partners typically make substantial efforts in order to preserve cooperation and dialogic friendliness.

Problems for the successful, goal-oriented flow of the discourse towards completion are not, of course, restricted to potentially argumentative (or explosive) encounters. Obstacles can and do arise in all kinds of dialogue[1], and these occasions provide particularly strong evidence for the kind of work speech participants do in order to preserve dialogic friendliness. The strategies and tactics which are brought into effect at such points are generally categorised as dialogic *repairs*.

Dialogue repair has been the focus of a range of well-known work within the field of Conversation Analysis. Repairs fall into two major categories - the overt and the covert. Only overt repairs will be dealt with here, as they are most usually

---

[1] This is true of interactional as well as transactional encounters.

associated with the transactional aspect of human-human encounters, and have similarities with the repairs which are sometimes needed in the case of human-computer communication.[2]

2.3. *Overt repairs*

Jefferson (1972) deals in some depth with overt repairs[3]. She describes the kind of dialogic troubles which lead to such repairs and gives examples of the repair work which the speakers do in order to get over the trouble. She divides repairs into three major types - *correction/clarification, affirmation* and *expanded action*. For the purposes of this paper, only the first two are relevant[4].

2.3.1. Correction/clarification

These are cases where a second speaker calls into question a particular item from the previous speaker's utterance. He may be questioning an incorrect item, e.g.

(2) One, two, three, eight, five.
Eight?

or possibly asking for more specific information about a particular item, e.g.

(3) I saw John in the supermarket.
John who?

Jefferson claims that the questioner's position here is competitive, in that he is providing some kind of challenge to the previous speaker.

2.3.2. Affirmation

In this case the second speaker provides a piece of 'side' information which the previous speaker either doesn't know or has forgotten, and which is useful to the successful continuation of the ongoing turn, e.g.

---

[2] For a discussion of covert repairs, see Cheepen (1988) and Cheepen & Monaghan (1990).

[3] Much of Jefferson's work focusses on primarily interactional encounters, but the repairs she discusses are associated with transactional phases within those encounters.

[4] Expanded action repairs operate only in encounters where there are more than two participants, and therefore have no parallel in the kind of human-computer communication dealt with here. In automated systems which permit group working they may have a significant part to play in the design process, but such systems are outside the scope of this current paper.

(4) I saw John today. John (pause)
John Whitehouse.
Yeh. John Whitehouse ....

Jefferson classifies this kind of repair as *non-competitive*, in that it is designed to help the current speaker continue with his topic. Frequently the non-competitive nature of this kind of repair is clearly signalled in the discourse by the first speaker actually asking for the side information, e.g.

(5) I saw John today. John what's he called?
John Whitehouse.
Yeh. John Whitehouse .....

In this way, the first speaker asks for the necessary information in advance, so as to avoid making a mistake.

These discourse strategies noted by Jefferson are what can be seen as *practical* repairs, that is, they are concerned essentially with cases where an 'error' of some kind must be corrected before the ongoing dialogue topic can proceed[5].

Clearly this kind of situation, so common in human-human communication, is often paralleled in human-computer communication, and where the computer system is set up to operate by speech input, the similarities seem, intuitively, to be very close indeed. In the following sections I shall investigate repair strategies in human-computer communication in the light of the operation of discourse friendliness in transactions, or what should properly, in any discussion of automated systems, be referred to as *user-friendliness*. Two specific automated speech-input systems will be used for examples - the CALE system which is currently under development at the University of Hertfordshire[6], and the earlier ISDIP system, which involved the same team of researchers, and which ran as an academic research project from 1988-91.

3. Human-computer dialogue

3.1. *User friendliness*

As I have already outlined in section 1 above, user-friendliness is essentially a matter of usability. An automated system, however powerful its functionality, must not only allow the user accessibility to that functionality, it must, for the system to be properly workable, be, in the fullest possible sense, 'foolproof'. That is, it must

---

[5] Note that although Jefferson describes correction/clarification as "competetive", this competition operates only very locally, at the level of the interruption. In terms of the ongoing dialogue it operates in a cooperative way, to 'smooth the path' for the current topic.

[6] The CALE project is partly supported by a substantial financial donation from Save & Prosper Educational Trust, a UK charity.

actively offer and promote that accessibility. This is not a trivial matter, given the widely differing levels of user expertise (and lack of expertise) which may be encountered, and the consequent differences in end user requirements. While a large amount of helpful on-screen information may be necessary for some classes of user, such as novices, or those who use a system only rarely, this may be an irritant for more experienced or frequent users, and may in such cases be actually counter-productive.

In our research with speech-input systems and their end users over the past five years we have concluded that, firstly, general standards of usability must be observed in system design, and they must be geared to the specific needs of speech, which is a very different input medium from keyboard. Secondly, because of the widely differing requirements of end users, and the high cognitive load which speech input tends to put on the user, many aspects of the user-friendliness, in terms of the human-computer dialogue which the system provides, must be closely customised to the precise requirements of the particular end user. When individual user needs are not catered for with sufficient precision there is a great potential for communicative breakdown, and this can be very serious indeed, resulting in problems with the functionality of the system, and consequent thwarting of the transactional goals of the human-computer encounter.

In order to avoid the possibility of communicative breakdown in the operation of the current CALE system and the earlier ISDIP system, the development team focussed a major part of their research effort on the potential problem areas of the human-machine dialogue. In terms of the parallels with human-human dialogue this meant taking account of the occasions when what Jefferson refers to as affirmation and correction/clarification repairs become necessary.

As Jefferson points out, in human-human communication, an affirmation repair is performed by the participant who has the major responsibility at the time for *listening* rather than *speaking*. While the current speaker is progressing a particular topic, the 'listener' provides a piece of side information which functions not to interrupt the speaker in any competitive sense, but to help the speaker with his/her topic progression.

The correlate of this in an automated system is *feedback* by the system to the user. Feedback is an essential ingredient of usability, so that the user is protected from errors which may arise through simply forgetting what stage of the process he/she has reached, and the screen must be appropriately arranged so that the user knows where to look for that feedback[7].

In addition to these considerations, it is equally important to consider the design of the dialogue which deals with the situations where usability is most at risk, i.e. where the human-computer communication does not go according to plan. At such times, which will inevitably arise even in the best designed systems, usability can only be preserved by efficient repair strategies, so that whatever difficulties may arise can be overcome and the user can progress the transaction (continue with the task in hand). In Jefferson's studies of human-human dialogue, these repairs are

---

[7] See section 3.2.2. below.

known as correction/clarification repairs. In an automated system this can be regarded as simple *error correction*.

3.2. *User friendliness in customised systems*

3.2.1. Automated dialogues

The CALE system, which is currently in development at the University of Hertfordshire, is designed for a particular user group - manually disabled school children, who require totally hands-free access to a wide range of standard computer packages. Input is by speech, and output is by screen display. The underlying speech recognition package on which the system is based is IBM VoiceType™, which is an isolated word, speaker dependent recogniser, designed for use by an individual speaker with his/her own set of voice files. VoiceType™ provides a total vocabulary of 7000 words, made up of a 5000 word standard dictionary, and a further 2000 words which can be selected by the user. The vocabulary is unsectioned, and all stored items are freely available to the user.

The research and development is being carried out with the help of a group of end users at a local school, who are using the current prototype system to carry out their school work, and providing continuing evaluations to the research team, so that each new prototype incorporates features which cater for the particular requirements of the user group. Customisation of the human-machine dialogues is particularly important for this user group, as they all have learning difficulties in addition to their manual disability. The dialogue structure also has to cater for the problems which can arise because the system is being used by a group, all accessing one machine. Because the system can only be successfully used if personalised voice files are kept separate, it is very important to ensure that the users do not accidentally access the wrong voice files, or there will be corruption of files and consequent failure of recognition. This is a case where appropriate feedback is necessary, in order to keep the user on the right communicative track.

3.2.2. Feedback in automated systems

The default for VoiceType is to boot up with no voice files loaded. It is the responsibility of the user to load the appropriate files. We have altered this part of the interface so that the system, rather than the user, initiates the human-machine dialogue. The system now in use boots up in a general voice file, accessible to all users. Because VoiceType™ is designed to be speaker dependent, this can only operate in a speaker independent fashion if the vocabulary permitted at this stage is very small and is made up of items which are clearly phonetically different, so that the chances of misrecognition are kept to a minimum.

To ensure that the user is constrained to this very limited vocabulary, the screen displays a menu of choices which allows the user to select a particular set of voice files. The initial CALE screen is as in diagram 1 below.

To use the system you will need to load your own voice files.

To do this, you must first turn on the microphone, by saying "voice console", then "wake up".

Then say the number which appears next to your name in the list below.

The CALE system will then load your voice files.

When you see the prompt showing your name you can begin work.

JOHN	1
SALLY	2
GEOFF	3
LUCY	4

Diagram 1

To support this, we have written a macro for each user number, which unloads the general voice files, loads up the appropriate individual voice files, and changes the prompt to the name of the user, in order to provide screen feedback as to which files are currently active. This means that while the DOS prompt is showing, the user always knows which voice files are being accessed, and the possibility of corrupting voice files belonging to other users is therefore considerably reduced. There is, however, still a potential problem when a software package is in use, because when the user is, for example, producing text by a word processor, the DOS prompt is not visible. In such a case, if the user is interrupted in his/her work, it is possible for another user to take over, and if the voice files are not changed, then there could be corruption.

VoiceType does provide a facility for the user to request feedback in this context by activating a command to check which voice files are active, but our field tests showed that this was unsuitable for our end users. To activate this facility, the user must say "**voice console**" (to show the main menu), and "**configuration**" to see which files VoiceType is accessing.

The response from the system is as in diagram 2 below.

<u>Configuration</u>

VoiceType Version 1.00
User: C\VOICETYP\john
Memory available: 3175K
Say "OK" to continue ...

Diagram 2

This was a problem for the children in our user group, because they found the word "configuration" difficult to say, and were reluctant to use it. This meant that they were not properly able to access the functionality of the system in this respect. To overcome this, we devised a macro which allows the user to simply say "**who am I?**", in order to get the same response. This is much more successful, as it caters directly for the user group by bringing this command in line with the kind of vocabulary and dialogue which is familiar to them in their ordinary human-human discourse.

In the earlier ISDIP project, which provided a simple, custom-built, speech-input word processor for disabled users, the speech input system was based on a more primitive speech recognition device, which, although it allowed a very large total vocabulary (all chosen by the user), constrained the user to section that vocabulary into subsets of 64 words. Only one subset could be addressed by the recogniser at each input signal, and responsibility for choosing which subset was active was the responsibility of the user. This therefore involved the user in switching between vocabulary subsets in order to operate the system.

Word processing requires a very large total vocabulary, so this meant that the user had to interact with many different vocabulary sets, and a major problem here was the confusion as to which set was currently active. The result of such confusion was only too often a total breakdown in communication, and in order to avoid this, it was important to provide constant feedback to the user. The development team designed the screen to give constant feedback on the currently active vocabulary. In diagram 3 below, which shows the ISDIP screen, the feedback box is in the top right hand corner. As the user switched from one vocabulary to another, this box was updated, thus giving the user constant access to an affirmation repair, which provided the 'side information' of the currently available vocabulary, and allowed the user to efficiently progress the communication task - i.e. getting the system to correctly recognise the latest input item.

file volume message vocab.

message box 1

scrolling text area

message box 2

Diagram 3

3.2.3. Automated correction/clarification repairs

In the context of an automated system, the errors which require what, in human-human dialogue are known as correction/clarification repairs fall into two classes - user errors and system errors. The user errors are common to all automated systems, whether speech-driven or keyboard-driven - they are cases where the user performs an action (inputs an item) and then wishes to alter that item (dynamically, rather than by post-editing). The system errors, however, are peculiar to speech input systems, as they arise when the machine recognition of an input item is imperfect (a situation which cannot, of course, arise where the input is via the keyboard). In both the CALE and ISDIP projects, the research team placed great emphasis on the provision of repair strategies for dealing with both categories of error.

3.2.3.1. Repair of user error

In systems where the input is by keyboard, user errors can usually be dynamically repaired by holding down the delete or backspace key until the mistake has been removed, and then retyping the correction. It is possible to transfer this kind of operation to a speech input system, and indeed there are commercial systems which do provide. This kind of repair strategy. It is, however, an imperfect solution, as it requires the user to repeat the word "delete" (or some equivalent item) several times to obliterate one word. Although this is efficient in that it certainly does the job, it can seriously impair the user's sense of succeeding with the system, particularly when the word to be deleted is a long one, and this is often perceived as a problem for the progress of the human-machine discourse.

In the CALE system, which relies on the powerful and sophisticated functionality of VoiceType[TM], this problem is not a serious one. VoiceType[YM] itself provides a **scratch that** command, which removes the previous input item (i.e. whole word), and also allows the user to operate by voice all the keyboard facilities provided by commercial packages, so that if, for example, a package used <Control C> to cancel a command by keyboard, it is possible for the user to carry out the same operation by a spoken command, such as **cancel** or some other appropriate word.

In the case of the ISDIP word processor, because it was not designed to work with commercial software packages, special consideration was given to the most suitable way to provide for the correction of a user error. The research team built into the system an "undo" command which, through the medium of a buffer which stored all the user's actions, allowed the user to undo the last action, *whatever it was*. This strategy not only improved overall usability by allowing the user to remove a whole input item (possibly a multisyllabic word) with one simple command, it also provided the simplest possible way of correcting other, more serious input errors, such as moving from text input mode to editing mode, or, while in editing mode, accidentally cutting the wrong piece of text. In these cases too, the user needed only to say "undo" and the previous action was reversed.

3.2.3.2. Repair of system error

The system errors which occur in any speech input system are the result of misrecognition of the input item. In rather primitive speech input systems, such as the one developed in the ISDIP project, this requires careful attention to the provision of an efficient and workable repair strategy. The ISDIP research team carried out extensive work in the development of appropriate dialogue structures in order to guide the user through the necessary steps of the repair.

The recognition hardware underlying the ISDIP system relied on simple template matching. The user 'trained' the system to recognise input items by setting up sound templates for them and associating certain keystrokes with those templates. When a trained input item was recognised by the system, the appropriate keystrokes were 'understood' and output to the screen. As each item was input, the system 'tried' to find a good match among the stored templates.

When the match between the input item and the stored template was not close enough to be classed as good enough to output an item direct to the screen, the system initiated a dialogue with the user to check whether a repair was necessary. This is best understood by considering the example where the user would say a word such as "wait", for output to the screen. In such a case, the recognition system would frequently find two templates which nearly matched the input signal[8] - perhaps "wait" and "late". The system then initiated a dialogue with the user by displaying the message: **close match - do you mean "wait"?**, as in diagram 4 below. Note also the bottom message box, which required the user to answer with "yes" or

---

8 The underlying speech recognition card constrained the system to only two possible matches of the input item.

"no". This functioned in a similar way to the affirmation repair found in human-human dialogue, where a repair strategy is set up in advance, in order to avoid a mistake[9].

file volume message vocab.

close match - do you mean "wait"?

scrolling text area

please answer "yes" or "no" only

Diagram 4

If the user answered "no" to this system initiated question, the system would then offer "late" as an alternative. If the user answered "yes" to this, then the word "late" was printed to the screen. If the user answered "no", the system would reply with "no match", and the user had to input the item again.

4. Human reactions to automated repairs in the ISDIP system

Throughout the development of the ISDIP wordprocessor, tests were carried out with novice users, so that any problems could be quickly identified and fed back into the design process to provide a basis for system refinements. In the case of the system initiated repair strategies, user reactions were particularly helpful in that they clearly indicated problems with the original design. The problems fell into two categories - *feedback* and *discourse convention*.

[9] See example (5) in section 1.3.2. above.

4.1. *Problems with feedback*

As I have already mentioned above, a vital ingredient of user friendliness or usability is feedback, and an important aspect of feedback is the arrangement of the screen, so that the user is not only informed about current processes, but is also encouraged to look at the appropriate section of the screen to receive that information. As diagram 3 shows, the original design included two message boxes, one near the top of the screen and the other at the bottom, and, as illustrated in diagram 4, the bottom one was reserved for the sub-dialogue message **Please answer 'es' or 'no' only.**

Tests with a wide variety of users, including novices and also those who were very familiar with the system, indicated an overwhelming tendency not to notice this message box. Where there was a misrecognition and the system offered a first 'guess' which was wrong, most users simply repeated the original word, rather than answering 'yes' or 'no'. In an attempt to rectify this, we redesigned the screen to show the second message box immediately below the first, as in diagram 5 below.

file volume message vocab.

close match - do you mean "wait"?

please answer "yes" or "no" only

scrolling text area

Diagram 5

When the new design was tested on users, their performance in terms of answering 'yes' or 'no' at the appropriate time improved substantially. There was still, however, a noticeable tendency to repeat a word when the system made a wrong guess, which indicated a secondary problem of discourse convention.

4.2. *Problems with discourse convention*

As indicated in the preceding section, users tended to want to repeat partially misrecognised items, rather than enter into a yes/no dialogue with the machine. All users involved in the tests were interviewed about their impressions, and the vast majority explicitly stated that they found this dialogue counter-intuitive, and would feel happier using the system if they were allowed to repeat problematical items rather than becoming involved in a system initiated dialogue. Acting on these user evaluations, we then set up a second version of the system, where this feedback was reduced to a minimum. In cases where the system could not be certain of a good template match for an input item, it was programmed to use only the top message and to tell the user **Sorry, no match.** Faced with such a message, the user then had no option but to repeat the item. The two versions of the system were then tested on a group of users, in order to discover how successful this modification was. The results were rather complex, and are described in section 4.3 below.

4.3. *Human perception of automated discourse in the ISDIP system*

Tests using the two versions of the ISDIP system were designed to compare the total number of user and system errors in a given task (dictating a prepared letter). The users were also interviewed extensively about the two versions and asked to say a) which system they preferred and b) which system they thought they performed best with.

The majority of users said that they preferred the system with minimal repair dialogues, and stated that this was because they were allowed to simply repeat a misrecognised word rather than enter into a dialogue with the machine. Questioned further, these users said that their opinion was based on the similarity of this minimal kind of repair strategy to human-human conversation, where, if a hearer misrecognises what is said, the speaker will repeat it. These interview results coincided with the error count, which showed that the minimalist version produced fewer errors than the more elaborate version.

The other category of interview questions (which required the users to say how well they thought they had performed with the different versions), was, however, a complicating factor in the final results. All users reported that, although they preferred the minimalist version, they thought they performed better with the elaborate version (in spite of the fact that the error count showed the reverse to be true). The conclusion which the design team drew from these apparently contradictory results was that the dialogues which were used in the more elaborate system obscured the fact that a misrecognition had occurred, so that the users did not count them as 'mistakes', but simply continued to interact with the system in a pseudo-conversational way.

5. Automated repair dialogues in the CALE system

The problems of dialogue design for repairs have largely been overcome in the modern, more sophisticated CALE system, due to the provision of repair strategies within the VoiceType™ software. In this system, the recogniser can select up to nine dictionary items which are a possible match for the input item. These are arranged in a pop-up menu on the screen, and accompanied by a tenth choice, which is "reject". If a user says, for example "wait", he/she will see a menu as in diagram 6 below.

F1	**wait**
F2	**weight**
F3	**wheat**
F4	**eight**
F5	**late**
F6	**ate**
F7	**lake**
F8	**ache**
F9	**wake**
F10	**[reject]**

Diagram 6

The user can then simply say "choose 1" (or whatever number appears against the required item), and it will be output to the screen. If the required item does not appear on the list, he/she can say "choose 10", and repeat the item[10]. The advantages of this over the earlier, and more restricted, system used in the ISDIP project, are threefold.

Firstly, the provision of a list of nine 'guesses', rather than two, means that in most cases the required item *will* appear on the list. Like the ISDIP system, this listing of possible matches operates as a repair strategy set up *in case* a correction should be necessary, but the chances of providing the correct repair are greatly increased. Our user tests show that even when the desired item is not number 1 on the list (and is not, therefore, the system's 'best guess') users are unperturbed by this, and do not perceive that having to say "choose 2" or "choose 6" is, in fact, an

[10] There is also a facility for the user to begin spellin in the item, in which case the system responds, after one or two letters, with a new list of 'guesses', but that operation is not strictly relevant to my discussion here.

error correction.

Secondly, the constant availability of the "reject" option as part of the list is a clear and unambiguous dialogue signal to the user that all the nine guesses could be wrong. Our tests have shown that this is intuitively appealing to users, who have no trouble in understanding that they must first reject the list (by saying "choose 10") and then say the item again. The success of this particular repair provision is due to its similarity to human-human dialogue, where problems can arise at the level of the communication channel, and a listener does not hear what a speaker says and will say "Pardon?" or "What?", in order to make the speaker repeat his/her utterance.

Thirdly, this kind of repair strategy is advantageous for the user for the very practical reason that, while the system is in operation, it is always available - i.e. a pop-up menu of possible matches appears *when each item is input*[11]. This means that the user quickly becomes familiar with the dialogue set-up which is being offered, and is not required to learn a range of different tactics to correct misrecognition errors. The dialogue style required by the system is constantly reinforced, so that the user is in no doubt about what form his/her part in that dialogue should take.

6. Conclusion

Our developments and tests to date with the ISDIP system and the CALE system illustrate a point which is fundamental to system design. Although it is important to ensure that any interactive system is efficient and potential errors are preferably avoided, this alone is not sufficient to ensure usability/user-friendliness. An essential factor in user-friendliness must be to ensure that the human-computer dialogues are clear and unambiguous for the human participant, and to provide strategies for feedback and repair of possible errors which fit with the user's expectations of how dialogic discourse can be progressed toward the transactional goal of the encounter.

When the dialogue for repairs can be so designed as to be, in a sense, unobtrusive - that is, the user is unaware that an error has been made and must be corrected - then the user satisfaction (and user-friendliness) is greatly enhanced. This means that the system is properly usable, and the human user has full access to the system's functionality and is free to concentrate primarily on the transactional goal of the encounter - completing the task in hand. The negotiations required for monitoring and management of the ongoing discourse do not then add to the cognitive load of the user, but can be performed 'naturally' - that is, in the same way as in human-human dialogue.

---

[11] It is possible to turn off this facility, but in practice we have found that this is only suitable for very specialised kinds of use. For normal, day-to-day operation, it is more efficient to retain the menu.

References

Brown, G., and G. Yule (1983) *Teaching the spoken language.* Cambridge: Cambridge University Press.

Cheepen, C. (1988) *The predictability of informal conversation.* Pinter Publishers.

Cheepen, C., and J. Monaghan (1990) *Spoken English: A practical guide.* Pinter Publishers.

Goffman, E. (1971) *Relations in public.* Allen Lane: Penguin Press.

Jefferson, G. (1972) Side sequences. In D.Sudnow (ed.), *Studies in social interaction.* New York: Free Press, 294-338.

Jefferson, G., and J. Schenkein (1978) Some sequential negotiations in conversation: Unexpanded and expanded versions of project action sequences. In J.Schenkein (ed.), *Studies in the organisation of conversational interaction.* New York: Academic Press, 155-172.

Monaghan, J., C. Cheepen, J. Hewitt, and J. Hobson (1990) Simulation in the evolution of an intelligent speech interface. In M.A. Life, C. Narborough-Hall, and W. Hamilton (eds.), *Simulation and the user interface.* Taylor & Francis.

Monaghan, J., C. Cheepen, and J. Hewitt (1991) Dialogue design for speech interfaces in the office context. In *Proceedings of HCI International '91.* Stuttgart.

Zajicek, M., and J. Hewitt (1990) An investigation into the use of error recovery dialogues in a user management system for speech recognition. *Interact '90.* North-Holland.