The Interaction of Discourse Modality and User Expectations in Human-Computer Dialog

Jussi Karlgren
Department of Computer and Systems Sciences
The Royal Institute of Technology and Stockholm University
Sweden
jussi@sics.se
May 1992
Licentiate Thesis, Stockholm University

Abstract

This study discusses the behavior of people towards natural language interfaces. It draws parallels to the behavior of people towards other people, and discusses how far these parallels can be stretched. A small experimental study of users performing tasks using a natural language interface to a database is presented, and the results related to the discussion.

The main points made are

1. that new modalities like the one used in typical human computer interaction - written interactive communication - are problematic for new users, from lack of conventions;

2. and that users' attitudes towards computers and of the system's linguistic and other competence shape much of the interaction, and that these attitudes change, and that thus the important factor to take into account in system design is not what the initial attitudes are but rather what the process of changing them is and how to utilize the process of change to teach the user the system language and interaction modality.

Acknowledgements

There are several people without whom this thesis would not have been possible, who have all commented, discussed, derided, encouraged, and helped formulate the ideas in this text. I want to thank them all for their constructive suggestions and comments: my advisors Carl Gustaf Jansson, Gunnel Källgren, and Yvonne Wærn; my father Hans and my brother Klas; the natural language processing group at IBM Nordic Laboratories: Ivan Bretan, Per Kristiansson, Bertus van de Poll, and Mohammad Sanamrad; my friends at SICS: Kia Höök, Björn Gambäck, Christer Samuelsson, and Carl Brown; and the anonymous subjects who donated of their time to participate.

Contents

1 Introduction

There are numerous factors that determine the properties of dialog between people: many of them are relevant to the study of human-computer interaction as well. One is the specificity of the counterpart. We model our linguistic behavior to what we perceive the counterpart to be like: a machine is a special case, and the attitudes of the user community towards the computer, and how these attitudes change, will determine part of the dialog's properties. Another factor is composed of the characteristics of the communication situation: we revise our picture of the counterpart on the basis of the counterpart's behavior, which is strongly determined by the mode and channel of the interaction.

In the special case of dialog investigated here, one of the parties in a discourse is a computer system. The general cooperative principles of human-human dialog hold for this special case as well, or should hold: systems should aid users to learn the language and conversational conventions that the system handles as fast as possible by giving users large amounts of sophisticated feedback thus training users to converge on the language and mode of interaction the system uses and prefers faster. It is important that users are presented linguistic cues that are designed to influence their language towards smoother interaction, and that they are given help functions to aid them in learning the linguistic capabilities of the system: people need and expect linguistic guidance to pose queries to natural language interfaces. Users will not know what language a natural language system handles, and will try out different constructions until they find a working syntax; until this happens they are disoriented.

The factors discussed in this thesis will influence the implementation of cooperative and conversationally competent systems. What the user community expects of machines as interlocutors is important, and is likely to change wildly and unpredictably over the next few years, depending on the certain wider dispersal of linguistically relatively impoverished systems of yesterday, and on the possible wider dispersal of the linguistically more competent systems of today and tomorrow.

1.1 Summary of Contribution

This study examines the specific qualities of human-computer dialog, and its effects on the users' expectations on the system. The characteristics of the dialog situation that users will encounter with interactive systems will form much of the attitudes of the users: the dialog situation should be designed with this in mind. This study shows that initial expectations of users are not as interesting as the mechanisms whereby they are changed, and that new modalities are lacking in conventions, and may be difficult for users to handle. The qualities of a modality may be difficult to display just as the competence of an interactive system may be difficult to display: they are interconnected in the sense that the qualities of a modality encourage users to hold certain expectations on the system competence or discourage a user from holding certain expectations.

1.2 Organization of this Thesis

Chapter 2 discusses human-human interaction and studies thereof, inasmuch they are relevant to human-computer dialogs; chapter 3 begins with a discussion relating chapter 2 to human-computer interaction and concludes with the formulation of a series of principles of human-computer interaction; chapter 4 describes an existing system and its features related the principles formulated in chapter 3; chapter 5

describes a small experimental study made on the system described in chapter 4 and relates the results to the principles from chapter 3; and finally, chapter 6 concludes the thesis with a discussion on the consequences of the principles proposed for future systems design and evaluation of systems.

2 Background

2.1 The Point Of Having Natural Language Interfaces

The point of building natural language systems to replace formal language systems is to make interaction easier. The assumption is that natural language interfaces lower the threshold for new users to start using systems, and enable users to utilize their everyday communication skills when interacting with the system. Natural language interfaces allow new users to use a new system with less bother. We may in the future assume large numbers of users using large numbers of interactive systems. Most users will most probably use each system they use infrequently, and will not want to relearn interaction languages for every system, and thus, natural language systems allow users to use several systems with less marginal effort than formal language systems would.

Users are meant to have less[1] learning to worry about when using a natural language interfaced system than they have to when using formal language interfaced systems. They are meant to shift the work burden from clause construction by the user to clause parsing by the interface system. Shifting work from user to system does not have to do with the formal properties of language, but with which language it is. Natural language is not easier, but it is there.

Users of natural language interfaced systems can not be expected to be familiar with computer systems, whatever systems they will be using: databases, advice-giving systems, and others. Typical users can not be expected to have any training on how to pose queries, and users should be assumed to be too impatient to bother with any formal guidance on how to work. On the other hand, users of natural language interfaced systems do have recourse to their natural behavior, including learning and adaptation. This natural cooperative behavior is what gives natural language technology a reason to continue work.

2.2 What is a Natural Language?

When is a natural language natural? Natural is usually used in opposition to artificial - but in this case the distinction may be drawn in the wrong place. There are several good examples of artificial languages that seem very natural to all of us: it does seem to be natural to add 89,234,134 to 1,772,988 using plus and minus; using the English language would definitely not be more natural, in any case, than would using arithmetic. And arithmetic is a prime example of an artificial language.

Further, the languages we speak and write are artificial too. Extreme examples of artificial spoken languages are languages like Esperanto, Volapük, and Interlingua, but there is no need to look that far. Estonian, Finnish, and Nynorsk are three languages spoken and written in a perfectly natural manner by large numbers of people. These languages are artificial, in every sense of the word. Their respective lexica - vocabulary and spelling - and grammars - phonology, morphology, and syntax - were all designed. And further, most languages of Europe, Swedish, for instance, undergo frequent revisions and formal changes, decided by boards and academies, in many cases in an artificial manner[2]. It has long been the contention

[1] Or, according to some early optimistic system designers, no learning at all to do, as referenced by [33] Pylyshyn.

[2] As I understand, the academics wear tails.

of linguists that all written language to begin with is artificial: it follows artificial norms and uses an artificial vocabulary. The German linguist Hermann Paul, for instance calls the spoken languages of Germany natural, or "natürliche" and the written norm artificial, or "künstlich". The influence of written language on spoken language is well established by linguists.

After establishing that the intuitive naturalness of the language spoken and written in a linguistic community is debatable the question is what a natural language is. Since arguably artificial languages are arguably natural, what gives a natural language its specific natural properties? A criterion for naturalness has been formulated [35] by artificial intelligence researcher Elaine Rich. She writes that naturalness never is a property of language in isolation, but rather a property of the relation between the language and the set of things it will be used to express. The point of the question is that some tasks are unsuitable for human natural language interaction due to their analog nature: word processing, designing, and other such tasks. With tasks of this type natural human verbal languages are less natural than, for instance, a point and click direct manipulation language. Nils Dahlbäck discusses [10] the same point in his doctoral thesis, where he makes a distinction between language that is natural in a specific situation, for a specific conversational setup with a specific counterpart as opposed to the view that natural language interaction should have computers play at being human.

The point is that "natural" in "natural language interface" should not be taken in the same sense as in the opposition between nature and nurture. A formal artificial language can well be natural for a given task, and it is in this sense that "natural" will be used below. "Natural Languages" in natural language systems today are not natural in either sense of the word, but may become natural in the sense that the users will feel more comfortable working with them than with any other language. Also, naming a language a natural language in virtue of its being natural for a task, learnable, and comprehensible may seem pleasing rhetorically, but will not really conform to the everyday perception of natural languages. Pressing a button to activate a doorbell could well be considered a natural language by a strict application of the naturalness criterion sketched above. In the following natural languages that resemble human interaction languages will be considered.

Rich asks: what is the appropriate role for English as an interface language? Can it ever be as natural as an artificial language? A specific problem for interactive linguistic tool design is that given a reasonable task, set of tasks, or task domain, that have been considered appropriate for verbal interaction, the linguistic technology of today [3] provides us with a patchy coverage of human verbal language at best. So, even with natural language interfaces, users will have to learn a specific register or sublanguage[4] to perform tasks in. The question is how to make this process easy for the user. Rich' question is: how can an appropriate register be chosen so that users can learn it efficiently?

Figure 2.1 Adapted version of Rich' figure on relationships between languages

[3]Irrespective of what our expectations on future generations of tools, technology, and developments in research are.

[4]The notion of a sublanguage which has been given a precise mathematical significance by Zellig Harris [17] as a subset of a well defined set is reasonable to work with as long as the differences between users are minimal as compared to the difference between their language(s) and the language the interface system handles. For other cases, for instance if the competence of the system approaches that of the users, the notion of a sublanguage will instil a vague mathematical uneasiness in most mathematically-minded readers: sublanguages imply the existence of superlanguages, which existence of course is not evident in the case of naturally occuring usage of human language. In most of the text the mathematically less presumptious term "register" defined in sociolinguistic literature will be used where interaction researchers might expect "sublanguage" to be abused. Register is defined as a variety of language according to use, rather than according to speaker or geographical location; a precise discussion of the term is carried out [16] by M. A. K. Halliday.

and tasks.

Rich draws figure 2.1, on relationships between languages and tasks: the circles on the left represent the size and expressiveness of the language used; the circles on the right the semantic complexity of the task domain at hand. The problem she points out is that there are cases where there is a mismatch between language complexity and task complexity: the first relation depicted is an example of such a mismatch, where the language is weaker than the domain it is designed to describe, the second shows the ideal case where a language matches its task domain, and the third shows the case where the language is larger than the task - either by being redundant, or allowing expressions that have no correspondence in the task domain. This is both wasteful and irritating to users, and as Rich points out, this is the case in many current applications of natural language, which may lead to the not uncommon generally negative view of natural languages as interface languages: "overkill and inefficient".

A complementary question to Rich' on learnability is the following: once a register has been chosen, given the technology of today, how make the user learn it efficiently? The complementary question, while not as respectful of users as might be expected to, addresses the fact that given that we assume that human verbal communication - written or spoken - is a useful model to emulate for interaction, we can only provide partial coverage, due to the state of the art in language technology. Given an interaction language with the appearance of a natural language, users will learn it, if given the correct cues. If they do not, the language in no way at all deserves being called natural.

Ivan Bretan proposes [3] that some tasks hitherto considered typically suitable for natural language interaction[5] may not always be that. He shows examples of database queries which users may have difficulty formulating in whatever language their mothers taught them. The examples in figure 2.2 are his: the first is the type of query that users may find easier to pose in natural language than in formal languages, whereas the second is cumbersome both to formulate and to comprehend. It strains the limits of what human performance of English can cope with. A well designed iconic direct manipulation interaction language may be more natural for complex queries of the type Q2 exemplifies.

Q1 Who exports more oil than Norway?

Q2 Show me the area and population of all the countries that are members of NATO and which do not import any Volvo cars.

Figure 2.2 A query, easy to pose in English, and one that is not. (from Bretan)

So, human verbal languages will not solve all interaction problems with computers, and not even all linguistic interaction problems. Sometimes interaction simply is difficult. An example similar to Bretan's above ones are problems clients may have constructing useful queries or descriptions of their troubles when consulting doctors or lawyers. This type of problem can be partly domain related, due to lack of expertise, but will arguably largely be linguistic.

In summary, a practical question can be distilled from the discussion: given a complex task domain, and an imperfect interaction language, how can we make the language natural, i.e. easily and naturally learnable, and thus aid the user in concentrating on solving the task?

2.3 When the Interaction Breaks Down

In the process of learning an interaction language we can foresee a number of misunderstandings and unsuccessful tries to interaction[6]. Jean Véronis proposes [40]

[5] By natural language interaction researchers.

[6] Which may be surprising to consider acceptable. However, no evidence indicates that human users would object to a few failed tries in interaction, if the repair is graceful.

a typology of errors[7]. He first distinguishes between errors by the system and the user. Thereafter he utilizes Noam Chomsky's distinction [8] between linguistic competence and linguistic performance: on the one hand the knowledge of language that a language user has, and on the other the actual use and instantiation of the same knowledge. Language as it is spoken, written, or understood is not only formed by the abstract rules and regularities that constitute the language user's knowledge of it: the situation, the user's personal characteristica, both psychological and physiological, and other cognitive aspects enter into defining the actual performance. Errors of competence are errors that stem from a mistaken view of the language in use, whereas errors of performance are slips and mistakes pertaining to the situation - the keyboard, noise, concentration and other local phenomena. On a spelling level he exemplifies a competence error with a user typing "hipotanoose" for "hypotenuse" whereas a performance error could be typing "hypotenudr" [8] .Véronis further distinguishes between errors on different levels in the linguistic description of language: words and characters - the lexical level, clauses and phrases - the syntactic level, and concepts and meanings - the semantic level.

Figure 2.3 Véronis' typology of errors in natural language interaction between machines and human users

Figure 2.3 - taken from his article - shows a rather fine-grained typology with typical errors placed in the respective cells. The typology is useful, but not entirely practical: as Véronis himself points out, it may be difficult to classify an error according to it. Problems have to do with both the errors themselves and with error detection. Some errors are intrinsically difficult to classify correctly: his example is a user typing "ommit" in place of "omit", which may either be a keyboard slip or a spelling error. Some errors may be detected on a level where they are not best explained: e.g. an error where a user substitutes "their" for "there" might be - depending on what types of processing algorithms are used - discovered on the syntactic or even semantic level, whereas it might best be explained as a spelling problem.

My point is that the errors that are labeled as competence errors - both from the side of the user and the system - are not competence errors in an isolated sense, per se, but errors in the sense that they are discrepancies between user and system competence in a situation where the system reserves the right not to adapt, and has a lower competence level than the user. The main focus for a helpful system in this kind of a situation should be teaching the user its competence, and not the correcting of performance errors. Véronis stresses the point that the language or register that the system is competent in must be friendly: it must be transparent to the user. He discusses the problems of teaching a user a language: a user will not want to be encumbered with learning grammatical notation to be able to use a natural language interface; the language must be easily understandable. Véronis specifies some heuristics for designing friendly registers, with respect to their formal properties: they must be closed under common and necessary linguistic transformations, meaning that all observable linguistic rules in the register must be generalizable to the whole register. The language must be coherent, and its boundaries easy to detect by trial and error.

So in the following, the theme will be how to design the system competence such that the user learns it with a minimum of fuss. The question here is how to identify discrepancies between the user and system competence, and how to have the user and the system converge on a common competence using the natural mechanisms of human linguistic behavior, i.e., using performance, in Chomsky's - and Véronis'

[7]Defined by him to be any situation that is unexpected by the system and in which communication fails. He stresses that the use of the word "error" does not imply that communication failure always is to be blamed on the user.

[8]Examine a qwerty keyboard for an explanation.

- terms, to teach competence.

2.4 The Audience Design Principle

There are some noncontroversial assumptions and observations made by discourse theorists that lie as a basis for most work in the area: people have some knowledge of their counterparts in discourse; people tend to act following this knowledge; people will monitor their counterparts' behavior, to ensure smoothly functioning communication; people will dynamically revise their knowledge of their counterparts in accordance with the observations they have made during a discourse.

As a special case of this adaptive process a hypothesis is that people tend to adapt their linguistic behavior to that of their counterparts in discourse. An observation to corroborate the hypothesis will be familiar to anyone who has spent some time in an environment where people speak a language or dialect similar to but different from their own language: a Stockholmer visiting Helsinki or a tourist with reasonable command of the English language on a tour of China will return to their home with their language changed. And anyone speaking and listening for any length of time to a person with a characteristic dialect will recognize the slight feeling of anxiety, worrying whether the counterpart will notice the transfer, and take exception to it. "Will they hear I take after them? Will they believe I am making fun of them?" There appear to be mechanisms that make people adapt to their linguistic environment. Naturally these mechanisms are highly individual as to the degree they effect interaction, but the direction in which they influence a normal cooperative situation is the same.

Several authors and thinkers have pointed out that discourse essentially is a cooperative endeavour that based on of shared understandings. From the more general observations of George Mead, for instance, who stresses [30] the aspect in conversations of taking the perspective of the other, to the more specific and elaborately constructed shared knowledge systems underlying most of today's work on language understanding, whose generally cooperative view bases its algorithms on a cooperative view, for instance as Grice has formulated [14] it in his articles on logic and conversation. The notion of mutual or shared knowledge is not unproblematic, and there is a debate on the subject in the literature on pragmatics of today [38]; for the purposes of this expose, however, the theoretical and formal qualities and properties of mutual knowledge need not concern us: what is obvious is that people do tailor their interaction to the perceived knowledge and capabilities of the counterpart - the so called Audience Design Principle. The fact is, after all, that communication usually works more or less well most of the time: and that there is a scale from completely dysfunctional communication to perfect rapport. Two things seem to influence the function of a particular communicatory situation: how similar the counterparts assume themselves to be, and how well they know each other. The one seems to offset the other: a different type of person may be easy to communicate with, provided that the person (or that type of person) is familiar; an unknown person may be easy to communicate with if the person is similar in terms of mutual understanding, experiences, knowledge and so on.

There are several reports on this sort of effects. Robert Krauss and Susan Fussell have reported [13], [27] empirical findings that indicate the practical effects of this tendency. In a number of interesting studies they have examined the characteristics and usefulness of messages that describe nonfigurative diagrams given by subjects to people they know on the one hand and to people they do not know on the other. It turns out, not very surprisingly, that subjects produce more useful descriptions of nonsense figures when they know the addressee, in the sense that the accuracy of picking out the figures on the basis of the descriptions was higher.

Given the natural tendency by a communicator to design a message for the

intended or perceived audience, it is natural for an interlocutor to assume that the counterpart in the conversation is doing its best for the conversation to proceed as well as possible. This where it feedback can be assumed to play an important role, to tune the two communicators' perceptions of each other. It can safely be assumed that this will have effects on the overall subject and referential expressions of the communication, and that they can be assumed to converge. The question here is if these effects are evident in the linguistic behavior of dialog parties as well: do people pick up linguistic behavior and language from a human counterpart? There are numerous studies that show that people do: one of the results are by Ray and Webb, in a study [34] on Kennedy news conferences. Apparently, at press conferences with president John F Kennedy, there was a positive correlation between question and answer length, not across question-answer pairs, but between conferences, and that this correlation increased over successive press conferences. This should be understood in the following way: press conferences varied between verbose and terse style, where either reporters asked long questions and got long answers, or reporters asked short questions and received short answers. This variation became more marked throughout the series of conferences, presumably because Kennedy and the reporters got more attuned to each others' styles. Another result is from a series of experiments [28] by Levelt and Kelter, where they found a strong positive correlation for question-answer pairs like "At what time do you close?" - "At five o'clock." on one hand and question-answer pairs like "What time do you close?" - "Five o'clock." on the other. In their examples, where the existence or non-existence of a preposition has little semantic significance, the persistence of the effect shows that there is a definite tendency to recycle material from a preceding speaker. On the level of naming and reference this is also corroborated by a study [20] made by Ellen A. Isaacs and Herbert H. Clark: subjects that were pairwise given a task of picking out and arranging postcards by talking about them quickly established referential expressions for the cards. The subjects also showed behavior for assessing each others' expertise in the subject matter - views of New York City, in this case, and of transferring expertise: teaching and learning about the postcard views. One of the questions that will concern us in the following is if these results are generalizable to conversations where the counterpart is a machine, that is, if human users of systems that interact in language resembling natural language will take after the linguistic behavior of the system.

2.5 Can We Learn from Human Behavior?

The starting point for this study is that people will carry over part of their normal behavior from human-human interaction to the computer console. At the barest minimum those of us who have a belief that natural language interface technology may have something worthwhile to offer us hope so: this is the whole point of building systems for human language processing.

Several authors have drawn parallels between human-human interaction on one hand and human-machine interaction on the other. There are risks in using methods found useful for the description of people in conversation when designing machines to interact with people: the description of human-human interaction is one thing, and the prescription of human-machine, or rather machine-human interaction, is another. One obvious risk is the risk of losing interesting and explanatory features of cognitive descriptions if they are not found implementable or useful. The other risk is losing interesting and useful mechanisms for human-machine interaction because they are not considered linguistically or cognitively plausible. This text has an engineering bias. The aim is to understand human behavior in as much it is interesting for system design, and to incorporate the most useful parts of this system design into the system. The study of people qua people is a different and much

more complex matter: this study makes few claims in that respect. The difference here is between the aims of computational linguistics and cognitive science on one hand: what human characteristics can we model computationally, and those of system design on the other: what human characteristics should we take into account when building a system, and of those, are there any of them we can make use of in the process of building the system?

Aravind Joshi discusses some differences between human-human interaction and human-machine interaction in an article [23] on Question-Answer systems. He proposes that some conventions in human-human conversation may have arisen as reflexes of processing constraints in humans, much as some syntactical properties of human languages are assumed to have a base in processing: we act the way we do and say the things we do because we know that our counterparts are human and think similarly to ourselves, including all our limitations. In human-machine interactions we may expect the machine to process certain aspects of the conversation more efficiently than we do - Joshi's example concerns inferencing - and others less efficiently than we do - pronominal resolution, for instance. These attitudes will arguably influence the conversational conventions that are followed in the course of a dialog.

In summary:

- Natural languages are not unproblematic notions.
- Languages are natural with respect to a task.
- Learnable languages are natural.
- People try to learn the language of their counterpart.
- People try to teach their language to their counterpart.

3 Adaptability in Discourse

3.1 How Is Human-Computer Interaction Different?

As outlined in the preceding sections, the whole point with natural language systems is that the interaction is performed in "the language of the user". This assumption seems to imply that there is little difference between the language a user uses with computers compared with the language used with people; on reflection, it is well known that people use very different linguistic methods depending on who they are communicating with. This is a well known fact from the study of language in use. Actually it would be odd, to say the least, if people were to use the same language with a computer as with, say, their elderly mother. It can safely be assumed as a null hypothesis that there will be special features on the language that people use when communicating with computers. This has been shown to be true, on several levels, by several studies and several authors.

Lexical variation will of course be dependent on the task, and the level of generality of the tool in use. Syntactic variation is limited, when users interact with computer systems. This is implicitly noted [37] by Christer Samuelsson and Manny Rayner, who succeed in crystallizing a shortcut through grammars for natural language interfaces using example-based techniques: users only use part of the linguistic freedom of expression available to them. Their results seem to indicate that this always is true, even in communication situations between human interlocutors; the results they present are collected from natural language interfaces to databases. Nils Dahlbäck also notes that syntax is simpler than needed in his Wizard of Oz simulation studies [10] as have several other studies [15], [25], [29], [39]. One of his

results indicate that users seem to use fewer pronouns when communicating with a natural language interface system than when communicating with a human over the same channel. Brennan, in contrast with Dahlbäck's results, indicates [2] no difference between two such conditions as regards pronoun use. This can probably be related to the interactivity of the situation, which will be discussed further below.

She shows that users start out using different language for computer and human counterparts, but that the difference disappears during the course of a session. Bozena Thompson has studied [39] human human interaction in two different modes compared to human computer interaction and shows clear syntactic differences between the different conditions. She says "In real life applications of computers the language is natural in a very specific sense, since it is constrained by the linguistic and situational context and subject to the inevitable restrictions of the computational grammar and the general requirements of this type of interaction. However, if computational interaction is to be natural, forms of language which are natural in normal dialog as well as those particularly suited to the application should be available to the user." In her study she finds clear differences, as well as similarities between the different conditions. She notes that "The dominance of simple sentences is striking. The reason is certainly not the lack of availability of complex sentences. ... On the whole, one is forced to conclude that monotony of structure is the rule rather then the exception in human computer interaction".

The output users provide to computer-generated queries show the same pattern. Kennedy, Wilkes, Elder, and Murray studied [25] answers given by test subjects to questions given over a teletype terminal. Some subjects believed the questions originated from a computer system, others that there was a person at the other end of the line, when in fact a computer system asked the questions in both cases. The style of language used by the subjects in the two groups differed markedly, where the subjects in the computer condition consistently output simple answers with very few embellishments, whereas the subjects who believed themselves to be communicating with a human used a much more varied language.

On a discourse level, Dahlbäck shows [10] that an extremely simple initiative-response model of discourse organization suffices to cover almost all of the interaction in his experiments. It is not to be expected that a simple model like this would describe interaction between human interlocutors, even in the same sort of task domain. Dahlbäck also notes the absence of politeness forms like indirect speech acts in the dialogs, and the presence of abrupt changes in dialog direction. This last observation is noteworthy, since semanticians, pragmaticians, and natural language interface engineers have all expended some effort into devising formal methods to compute the meaning of indirect speech acts. Jarke et al report [21] that users do not communicate with computers as they would with a human. They seem to be more careful than they would otherwise, avoiding small errors that they would not expect the computer to cope with, but presumably would allow to pass for human interlocutors. Guindon et al report [15] the same, but still label 31% of the user utterances ungrammatical: on closer scrutiny this turns out to mean that the users have made use of extensive ellipsis and other fragmentary constructions - the grammar Guindon et al used to make judge the grammaticality probably was not constructed with the specific situation in mind. They unsurprisingly report similarities between human-computer dialog and spoken speech, if contrasted with written formal technical language - except with regards to referring expressions.

3.2 User Attitude Dependent Effects on Interaction

Several of the features of human-computer interaction described in section 3.1 are strongly dependent on the expectations that users have on the linguistic competence of computer systems and the perceptions users have of the personal presence

of computer systems. Some of these attitudes are unlikely to change, while most probably will, with the advent of more and more competent systems, and the dispersal of competent and incompetent systems alike to larger and larger groups of less and less technically aware users.

Part of the effects can be related to the normal and natural mechanism that leads users to have different expectations on the linguistic competence of different counterparts in dialog: these expectations vary continuously from individual to individual to computer system. All dialog counterparts you are likely to encounter are different, and computers are just another special case among others. It is always the case that some tuning must be made before dialog runs smoothly, and this is true for computer systems as well as for people.

Another part may be related to the fact as Dahlbäck puts [10] it: "Computers are not people". For instance, the fact shown by him and others, that people do not seem to feel the need to express themselves politely to interactive systems. This type of attitudes where machines are treated as machines and not like people are probably more resistant to change, as long as it is obvious to users what physical entity they actually are communicating with[9]. The conceptions of counterpart processing constraints may be the foundation of many of our conversational conventions, as was argued [23] by Joshi; when we know our counterpart to be a natural language system, we may use this knowledge to express ourselves differently. Wachtel uses [41] the term natural natural language as opposed to natural language in an article describing a system. Wachtel argues that in building interactive and pragmatically sensitive systems, we need to model - in view of modeling behavior for future implementation - not a typically natural open human conversation but a restricted type of conversation[10] that also occurs between humans in certain well-circumscribed contexts: his example is the conversation between the ticket clerk and a would-be passenger at a railway station. He specifically mentions that this is a consequence of the user realizing that the counterpart in conversation is a machine.

An example: as long as users believe that they have near total control over the dialog situation, as they will believe with a natural language interface to computer systems that they are comfortable using, they are unlikely to abandon sudden shifts in topic and unlikely to start using politeness forms. On the other hand, when users are more aware of the linguistic capabilities of systems they may start using a wider range of syntactic constructions. When users are more aware of the context kept and actively present in the computer system, they are likely to start using pronouns and other anaphoric constructions, and to likely to start referring to earlier segments of the dialog. For instance, Guindon et al relate [15] the differences they found between human-computer dialog and spoken speech with regards to referring expressions to the users believing the shared context was poor.

Attitudes and user models of system competence are difficult, if not impossible to study in themselves. They can only be studied inasmuch as they have effects on the interaction: while the attitudes themselves are invisible the symptoms of them are visible. This is also what interests us in terms of system design: the attitudes and models are not interesting: neither useful nor harmful, if they do not have effects on the interaction. The interest of a system designer is to have the user obtain the correct and functional models to ensure that the user uses a functional way of interacting: this should be done by carefully inspecting the discourse behavior of the user.

[9] Which obviously does not always have to be the case.
[10] Which, by implication, is unnatural natural interaction!

3.3 Modality Dependent Effects on Interaction

Much of what makes interaction with computers different from interaction with people is the modality of interaction rather than the nature of the counterpart. That computers are different from people is one thing, which in itself probably will influence users in their interaction with them; the most obvious way they are different in is that they communicate in a new modality. The modality, or the channel the interaction is performed through, will make a difference: speaking is different from writing, and writing on a keyboard is different from writing by hand. This comes as no great surprise to anyone, and has been known since the very beginnings of linguistics, since when linguists have - at times heatedly - discussed which form of language to study: Hermann Paul, for instance, points [31] out that: "In [die nationalen Gemeinsprachen] stehen eine schriftsprachliche und eine umgangsprachlice Norm neben einander."[11] Otto Jespersen[12] also makes [22] the distinction very clear. That spoken and written language differ on several levels, that they essentially have different norms and different rules for form, content, and use, and that they are not two different ways of describing the same language is no news to us and this has been shown in great detail by a number of studies since. The differences between written language compared to spoken has been described by various linguists both through experimental data and through observation: Gustaf Cederschiöld[13], for instance, describes [4] written language as calm, more logical, more exact, more varied in choice of expression, compared to spoken language which is more free flowing, and more able to conform to the personality of the recipient, and so forth.

In more recent and often quoted studies[6],[7] Alphonse Chapanis et al characterize, on empirical grounds, speech as faster, more verbose, less planned, more repetitive and dysfluent, as containing a less varied vocabulary, more pronouns, as using less complex syntax, and so forth. Chapanis et al studied the difference between several different modalities - face-to-face, voice only, handwritten communication, typed communication - in solving tasks, and found that oral modes are faster for solving certain types of task, but wordier. Their findings indicate that a dichotomy of modalities fails to capture some points of interest: there are interesting differences that can be related to the level of interactivity between face-to-face and voice only on one hand, and hand- and typewritten on the other. Wallace Chafe has shown[5] syntactic differences when comparing corpora of written language to spoken language: nominalizations, conjoined phrases, relative clauses, and other constructions which package more information into a syntactic unit are all more common in written language than in spoken. Chafe relates the difference to the larger amount of time typically available to writers and readers compared to speakers and listeners. He goes on to discuss the difference in interactivity between typically spoken and typically written language: he shows that some of the characteristics of written language show up in oral literature, that seems to be more like written language in certain respects, than spoken.

This difference between written and spoken language has always been obvious: we have had written language around for a few thousand years and spoken language presumably three to four orders of magnitude of time longer. The difference between writing and speaking has been greater in the past. Lately written language has begun showing up in interactive communication: today there are situations where the choice of modality, the choice of written or spoken language is freer than before:

[11] In paragraph 289.

[12] In the first chapter, on "living grammar". His point is that spoken language is primary and that written is secondary, and that this has consequences for linguistics. This view need not concern us here.

[13] Throughout his work on the characteristics of Swedish as a written language.

Call or use e-mail? Or use telex? Fax? The availability of many choices leads to the realization that the sharp distinctions between the two traditional channels and between the two traditionally conceptualized levels of interactivity are not as sharp as could be imagined. There is a continuous space of different modes of interaction, and we can choose to position ourselves in different points in the space according to the situation, available channel, mode, and other factors.

Figure 3.1 Some communication channels in a modality space

Figure 3.1 shows some communications channels positioned in a modality space. Naturally both interactivity and modality are multidimensional concepts: the axes in figure 3.1 are composites of several in themselves interesting factors that come into play in discourse. The horizontal axis is intended to represent the temporal qualities of dialog, the tightness of discourse, the factors that determine the difference between e-mail and any other type-written letter. The vertical axis is intended to capture the bandwidth and conversational broadness qualities: the factors that determine the difference between a hand-written note and a typewritten letter, or between face-to-face dialog and telephone dialog. There are numerous other factors that have been left out: number of recipients, the degree of bidirectionality - in this study the two composite dimensions indicated in figure 3.1 will be used.

Differences between modalities can be studied with other factors held constant. This has been done in several different studies: Kerstin Severinson Eklundh has studied [12] human-human communication in an electronic conference system, and determined several characteristic features of dialog: a specific feature pertaining to interactivity and that can be related to the dimensions defined above is that certain types of feedback tend not to appear. One of the examples her study focuses on is the turn-taking form of information-seeking exchanges that would be expected to be three-part under a spoken channel with a query, or request for information by a user, and an answer, and ended by a confirmation, or feedback turn from the requester, tend to be two-part under the electronic conference system channel as exemplified in figure 3.2.

	Spoken	Computer-mediated
Query	What is the title of your report?	What is the title of your report?
Answer	"How Users Adapt to Interfaces"	"How Users Adapt to Interfaces"
Confirmation	"How Users Adapt to Interfaces"? Thanks.	—

Figure 3.2 Three-part information seeking exchange compared with two-part The last conversational move present in the spoken case, an independent feedback move, tends to be missing in the computer-mediated case. Philip R. Cohen has studied [9] differences in vocal and keyboard instruction giving between two human parties, and found differences on the pragmatic level. When giving instructions, the subjects under the spoken condition tended to refer more elaborately to components in a simple assembly task than the subjects under the teletype condition. "Print may remove inhibitions, as may talking to a machine" he notes, and concludes: "Keyboard interaction, with its emphasis on optimal packaging of information into the smallest linguistic space appears to be a mode that alters the normal organisation of discourse."

In conclusion it is obvious that modalities, or channels matter. The modality of discourse will impose constraints on the interaction: we express ourselves differently to the same recipient over different modes. What is important to note that in the case of new situations with interactive modalities, it is not certain that the user will have knowledge of where in the modality space it is possible to position the dialog in question. The initial attitudes and images of the counterpart in question will help the user to determine at what point to start, but when users have little experience, they will be on the lookout for clues in which direction to move, and they will not be certain that they are in the correct position. A hypothesis is that they will start

out with a situation they are familiar with: in the case of natural language interface users today, most are familiar with traditional formal language query interfaces, or at least have some conception of how they work. Most users today are familiar enough with computers not to antropomorphize them, and unfamiliar enough with natural language query systems to expect them to be similar to formal language systems. This may change in the future: where in the diagram users will expect the communication with computers to begin will vary from user to user, depending on previous experiences. As the diagram in figure 3.1 is constructed each counterpart and channel may cover areas in it, interacting in diverse modes: fax machines can be used in various ways. The important thing for someone wishing to communicate is to find the optimal mode, or rather, a satisfactory mode. The question is how, or in what way, and what consequences this searching procedure will have for system design.

3.4 Do Users Adapt?

As has been described in the last few sections there are differences between human-human interaction and human-computer interaction, and there are differences between written and spoken interaction, and there are differences between more and less interactive interaction. The question this study aims to address is how users adapt to the various conditions under which they interact with systems: given the various starting points of users with respect to their attitudes and models of system competence, and the space of modalities with its various available modes of dialog, how do they ensure an interaction will function? The hypothesis of this study is that they try to adapt to the language the system seems to prefer and they try to gain clues as to what type of interaction the system supports. One visible effect, is the contention of this thesis, will be that they pick up and recycle language from the counterpart, as has been shown to be the case in human-human interaction, and as will be seen below, has been shown for human-computer interaction as well.

Lars Ahrenberg, Nils Dahlbäck, and Arne Jönsson, in simulated natural language interaction studies [1], have shown influence on users' language over sessions on a lexical level. On the other hand, in the study [25] mentioned in the previous section, Kennedy et al note that subjects show very little adaptation to the language of the counterpart, whether the counterpart is perceived as a human or a computer. These results, which at first sight seem incompatible, can probably be explained with modality differences; there is a certain type of protective conservativeness in a new and unfamiliar situation - like a new modality will be - "if the interaction works, don't fix it". Susan E. Brennan shows [2] differences in syntax: at the beginning of sessions users who believe themselves communicating with a computer system use a complete sentence only half the time, compared with all the time for users who believe themselves communicating with another human. She ran the subjects under different response conditions: verbose and terse, and found that towards the end of the sessions there was no difference across different counterpart conditions, but across response style: the response style had had the users adapt to the language given back as feedback, and the initial attitude had lost its effect. Elizabeth Zoltan-Ford shows [42] the same effect in a study where she compares several conditions: she shows that the user tends to model the length of the response - what is termed the "speech duration effect", discussed in the previous section on adaptation between human interlocutors. She also shows that a terse style is more likely to catch on than a verbose style, and that users are more likely to adapt if their queries are not understood when they do not. What is most interesting is that she shows no differences in adaptation across channel of communication: half of the test subjects interacted in spoken mode, half in written or typed. Initial attitudes thus seem to have little to do with what attitudes the user ends up with: the important factor is

adaptation.

3.5 What To Study and How To Do It

There are problems when formulating a study about underlying factors of adaptation in interaction: the study can observe symptoms - surface features of interaction. How to explain them where they come from may be unclear. In the case of modality, and the specificity of the counterpart in human-computer discourse discussed above, there are mutual influences: modality-dependent features will affect the picture that the user has of a system counterpart; the picture the user has of a counterpart will affect the use the user makes of the modality. Thus, the two notions are not independent of each other, but neither is the distinction a false one: just as we vary our register in speaking to different humans, we vary the register and style we communicate in with a certain person over different modes and channels. The study presented in this thesis will examine the behavior of users, and determine how the users made use of the subspace of the entire modality space made available to them by the system.

The fact that users may have preconceptions about system competence will influence the starting point in the modality space. We can say little of attitudes and mental models: we can only observe behavior and draw inferences from it. And as has been said above, in section 3.2, this is what interests us in terms of system design: in this study, the attitudes and models are not interesting in themselves if they do not have effects on the interaction.

The questions to study are if the effects evident in human-human interaction are strong enough to be useful and noticeable in human-machine interaction. In other words:

- Where in the modality space will the user start an interaction: what are the initial expectations?
- How does the user learn where in the modality space the dialog with the system can be performed, and how should the system make the user aware of a satisfactory position?
- What difficulties are inherent in the mode that typically is used for human-computer dialog?

There is no scarcity of experimental data. Most of the results reported in discussions above are from experiments made with simulated natural language systems - so called Wizard of Oz studies - where it is easy to manipulate the different conditions tested: easier than it would be to build a real natural language system. Simulation studies have been described [11] for instance by Nils Dahlbäck and Arne Jönsson. Testing a real system that in fact is being used at industrial sites across the world gives data of a different sort: as has been argued in previous sections, the modality and interactivity of the dialog situation may give rise to specific effects from system to system: this may explain the partly inconsistent results gathered from simualation studies. Ideally, the study should be from real field studies. There are practical problems with gathering field data, though: most real commercial users will feel that researchers have no business poking about their database. In the study described in the next sections, a real commercial system is used with a demonstration database and test subjects chosen to participate in a study. Using a real system proved practical: building and manning a simulation system is infinitely more work than using a real and well functioning product built and tested by large teams of software designers.

In summary:

- Human-computer interaction is different from human-human interaction.
- Users have attitudes about their counterparts
- Computers are a special case of counterparts
- Attitudes change behavior
- Attitudes change on the basis of behavior
- Behavior is analyzed, among other things, in terms of adherence to communicative conventions which are partly channel- and mode-specific.

4 The System

This section describes the IBM SAA LangugeAccess system, a commercial natural language query interface to relational databases, available as a program product. The system has been developed at the IBM Nordic Laboratories which is also where the study described in the following section was performed.

4.1 System Architecture

IBM SAA LanguageAccess is built around a natural language component, a so called natural language engine, which uses a grammar and a conceptual schema to produce database queries from natural language input as indicated in figure 4.1 below. The system also produces paraphrases of the queries to enable the user to check the system's interpretation of the query. The system can be connected to any relational database under the appropriate computer configurations, and is in principle portable from language to language, given a well enough established grammar.

Figure 4.1 A simplified overview of the IBM SAA LanguageAccess system with an added on help prototype

A conceptual schema provides the link between the coded language on one hand - the lexical items and syntactic structures - and the relations of the database. Figure 4.2 below indicates what a conceptual schema may look like. A schema of this type must be defined for each specific database or application that is to be accessed. The schema contains most of the lexical competence of the system, and as will be shown below, the design of the schema is important for the usefulness of the interface, and for its customers, IBM provides a advanced tool - the Customization Tool - to build and manage a conceptual schema. The example databases used in the study were adminstrative databases of programming projects and companies, with fairly simple conceptual schemata defined for them.

```
Terms:
manager, consultant, salary, expense, cost

Relations:
employees have salaries
consultants have expenses
consultants cost expenses
```

Figure 4.2 Conceptual Schema Excerpt

The system is described in full detail in its different aspects by in several IBM publications [19] as well as in a recent article [36] by Mohammad A. Sanamrad and Ivan Bretan. The grammatical formalisms employed are described in a report [26] by Erik Knudsen.

4.2 Interaction

The interface allows users to enter database queries in what is described as standard English. The queries are first passed through a parser which performs a syntactic analysis of the string. This analysis is in the form of both a syntactic and a semantic tree representation, constructed simultaneously by the parser.

The semantic trees are used to construct a series of internal representations which are then in turn used to generate paraphrases and formal database queries in SQL form.

Query:

```
'List managers with levels.'
```

Semantic Tree:

```
command(
 tns(
  acc(
   npindef(
           attpp(
                 prep(
                      npindef(nomen(t8)),
                      pp,
                      prepos(value(48)) ),
                 nomen(t3),
                 48 )),
    verb(list) ),
   [ca = 1, cmd = 1, es = 1,
    imp = 1, syst = 1,typ = az]))
```

Concept-Oriented Logical Form:

```
query(report,
set(y8,
   set(y7,
      instance(manager,y8) \&
      instance(sempl\_level,y7) \&
      possesses(sempl,sempl\_level,y8,y7))))
```

Paraphrase:

```
Find employees that are managers and have levels.
```

SQL:

```
SELECT X1.SERIAL,X1.EMPNAME,X1.‘‘LEVEL’’
 FROM SLV.SEMPL X1
 WHERE X1.SERIAL
 IN (SELECT X2.MANAGER
  FROM SLV.SDEP X2
  WHERE X2.MANAGER IS NOT NULL)
```

Figure 4.4 Query, internal representations, paraphrase, and SQL

The user enters a query and receives a paraphrase, or - when the system cannot interpret the query unambiguously - a list of paraphrases. The user is requested to approve one of the paraphrases for database processing. After approval the SQL form of the query is sent to the database query facility.

The paraphrases have a distinctive syntax. The paraphraser is disjoint from the analysis component: both the algorithms and the grammars are different. They

have different aims. The paraphraser does not need as much elegance in its expressive form, but it needs to phrase itself unambiguously. One of the points of the paraphraser is that it provides, and makes explicit, different readings of the query in order to aid the user in disambiguating between them. The paraphrases are designed to be unambiguous: the language used is formal and verbose, and one of the assumptions of this study was that during the course of the session it would transfer to the users' language: it is one of the only sources of natural language available to users.

Query: 'show me all managers'

Paraphrase: Find managers.

Answer: ID FNAME LNAME —— ——— ————— 1 DEBORAH JONES 2 ERIK JONSSON 3 FRITZ JOCHEN 4 GERTRUDE JOFFE

Figure 4.5 Typical Interaction

The answers displayed by the system are not given in natural language: the answers are produced by the database management system and the report or the chart that is displayed on the screen is generated by the query interface and by the interface management products it avails itself of. Typically the answer is in the form of a table, as in figure 4.5, but different types of charts are also available. The user can specify in the query how the answer is displayed. Thus the interaction with the system is a highly specific type of dialog situation: the user has full initiative, and system volunteers nothing: the only feedback given to a user is the paraphrase and the answer table, or an error message in cases where the system is not able to interpret the query. The users always have recourse to the immediately preceding query: the new query is entered in the same edit window as the previous query was, and when the edit window is used, the preceding query is still displayed in it for editing. When a word or term is not defined in the schema, it is highlighted in the query window to indicate that the system will take it as a name instead. This may lead to more editing and less new typing than might be the case otherwise. The users may also have all previous queries available, displayed in a separate log window. This window was not used by any of the subjects, however.

4.3 Help Prototype

The standard on-line help provided by SAA LanguageAccess - in addition to the normal documentation and User's Guides - consists of a series of help screens, with a fixed text describing SAA LanguageAccess and listing some examples of queries for different database systems: no specific help is given, for any given conceptual schema. The examples indicate what types of syntactic structures are available.

List nouns ... manager ... List relations for "manager" ... employees report to managers ...

Figure 4.6 Help prototype access

Some subjects were given access to a help prototype - which is not a part of SAA LanguageAccess - that given a conceptual schema and a grammar, will produce phrases and queries about terms defined in the schema. These model queries and example phrases were generated using the conceptual schema defined for the database used for the experiment. The help prototype also provided functions to inspect the vocabulary of the conceptual schema and the lexicon of the natural language engine. The help prototype will be described in a forthcoming [32] report by Sanja Petrovic.

Type a number to get the function you want

1 Exit Help 2 List vocabulary by categories 3 List relations for a term 4 Generate sample queries for a term

Figure 4.7 Screen layout with Main Help Panel

A subject is able to scan through the entire vocabulary sorted by part of speech, and for each term defined in the conceptual schema the user may request a list of phrases to be displayed to indicate what types of relations the conceptual schema has coded for that lexical item. An example is shown in figure 4.6. The access to the help prototype was through a menu driven interface which was somewhat unreliable and not altogether easy to use. All subjects that had access to the prototype did not make use of it, and most only used part of the functions available.

4.4 Qualities of the System in Terms of Modality What are the system's qualities in terms of its modality? The system takes written input only, and in a strictly user initiated manner: the user types queries, and the system responds either by answering or indicating that it did not understand the query. The user types queries into a query window with editing possibilities and when the query is composed to the user's satisfaction it is sent off with the Enter key. When a query has been processed and answered or rejected, the query window will be topmost on the screen, and the user will in the default setup have the previous query in it, to be edited or discarded for the next query. If the user has requested a log of queries, all successfully parsed queries are stored in a log window, where they can be retrieved into the edit window from.

How interactive is the system? In terms of a simple information-seeking query, the system answers when prodded to: it provides the answer asked for or requests the user to rephrase the query; the user is given full responsiblity for the dialog, and little possibility to influence and study the system: the system takes no initiative, and does not indicate what contexts it has or what information is available to it. The model for how a dialog is run is simple and easy to understand: always simple two-part interactions. However, in spite of its extensive grammatical coverage, the linguistic capabilities of the system are not obvious, and the system provides no feedback designed to make the user aware of what language it understands and prefers, excepting error messages in case of syntactic failure.

In terms of the modality space and the diagram defined and sketched in section 3, the users initially will have no idea where they are located. It is not to be expected that users can figure out for themselves what the system can do: at least not without extensive training. The users' preconceptions and previous expectations probably will place them somewhere in the vicinity of an existing mode of communication: in figure 4.8 some possible prototypical locations in the modality space are marked out. If the user has had experiences with formal language query database systems, for instance, it is likely that the user will expect something similar from the new system; if the user has high expectations of what the system competence is, the user may have face-to-face dialog as a model, or dialogs over electronic mail systems. It is in the obvious interest of the system to quickly position the user in such a mode which the system can interact in competently: none of the models are good, in the sense that they do not exploit the advantages of the system in an efficient way.

Figure 4.8 Modality space with system types

Natural language query systems are an improvement over formal language query systems in the direction of face-to-face dialog, and with a system like the help prototype in this study the level of interactivity is heightened dramatically. The user types in a term, and receives as a response all the information about that term's usage, expressed in the language that the system itself understands. This will heighten the complexity of the dialog, and will increase the number of turns taken by users for each query, if used, which may not seem as functional in terms of most statistical evaluations measures used today. However, it will enable users to clarify the terminology of a task area before attempting to solve the problem in it.

This mode of interaction can be positioned in the diagram, upwards and hopefully to the right, where each turn takes less time to effect, of the typical natural

language query interface. The system can now interact in several different modes, in a larger area of the diagram. As has been discussed in section 3.5 the qualities of the modality that the interaction proceeds in may cause the interaction to have certain features which will then cause the user to have a certain attitude about the counterpart; a certain attitude about the counterpart may lead the user never to use a certain available feature in the communication mode. It is now in the interest of the system to ensure that the user is aware of the size of the area covered. The question is what surface characteristics dialog in the different modes has, and how to influence the user to move the dialog to the most efficient mode available.

4.4 What Does The Dialog Look Like?

The second major influence that was discussed in the previous section was the specificity of the counterpart. As was discussed earlier, the attitudes and expectations of users can vary unpredictably over users and over time, and the interesting factor to investigate is not users' attitudes and expectations in themselves, but how they change, and how they should be made to change to keep the dialog as functional as possible. In this respect, the transparency of the system in terms of displaying its competence and dialog qualities is important to help the user gauge how the dialog can be handled. All the more so as the system in this specific case takes no initiative, as has been described above, and the user has full responsibility of running the dialog.

The user receives feedback from the system in the form of paraphrases, answers to queries, answers to help requests, and error messages. If the user receives a paraphrase, the system has made an interpretation of the query. In some cases the query may have been formulated so that the system interprets it in an unexpected way: queries may be unintentionally ambiguous, or the system may not expect a certain term to be used in a certain way. In these cases the paraphrases are useful to determine inappropriate interpretations. When the query is processed successfully and the paraphrase accepted by the user, the user receives an answer, and can then, depending on the type of the query, validate the answer, i.e. check to see if the answer is relevant to the query posed. On this level, the system gives the user several possibilities to pick up clues about system competence. The user can also ask meta-level queries about system competence, to find out about system competence explicitly: "What terms do you know?", for instance. If the user has recourse to the help system, which heightens interactivity of the dialog as has been described above, this process naturally becomes easier. If the help system is used, the user can converse with the system about the terms: "what do you know about X", and thus get a picture of the system's competence in a more explicit way. If the query is not processed in expected ways, the user receives an error message. In spite of the informative character of the majority of error messages it may still be difficult to understand what type of failure the system made in interpreting the query.

However, the user has no possibility to inspect what context the system currently works with. The system will attempt to handle anaphora, correlating pronouns with references in previous queries, but since the users will have no expectations on this type of functionality, and the systems does not indicate its high competence, the users do not know enough to exploit the mechanism, and this aspect of the system competence may not be used as much as it deserves, unless some mechanism to exhibit the competence to the user is conceived of.

5 The Study

14 subjects were asked to solve tasks using the IBM SAA LangugeAccess system, a commercial natural language query interface to relational databases, available as a program product, and described in the previous section. The study [24] presented here is small, and with no more than 13 sessions the study cannot be expected to be statistically reliable data on human behavior. This should be taken to mean that the results will not state facts about how people behave in general; the results are observations about features of both this specific interface and this specific interaction situation; these results are then discussed in terms of the distinctions made in section 3. The results do not tell us that users tend to do X rather than Y Z% of the time: they are observations on the usefulness of types of behavior. A certain behavior X is observed, and a certain other behavior Y is observed, and after noting that to accomplish a typical task, a behavior X is more useful than a behavior Y, the real result to be found in the study is to use this fact to point out how to build systems that naturally help users to behave in a more functional way.

5.1 Subjects

The subjects were chosen with a number of criteria in mind. Subjects had to be fluent in English, be unbiased with respect to the goals of the study, have no previous experience with natural language interfaces, and be sufficiently experienced computer users for the test to proceed without too many interruptions: both with respect to databases and the system itself.

It is not difficult to find reasonbably fluent English-speakers in Sweden. Most of the subjects used in this experiment were familiar with SQL and databases, and some had seen a demonstration of SAA LanguageAccess, while none had ever actually used the system. All were reasonably accomplished typists. As noted [18] by Patrick A. Holleran, in a methodological article, the motivation of subjects will influence the results of the study. Subjects may try to make the interface work, if they suspect that the tester is part of the design team. In general, subjects in experimental studies tend to put more effort into a task than they would in a non-experimental situation, both because of curiosity and the novelty of the situation and to perform well in a situation where they are observed, especially if they identify with the aims of the project. In this case, in virtue of their past experience of databases, most subjects naturally were more interested in trying out new software than the typical user would be.

In summary, the subjects will have to be considered atypically cooperative, atypically experienced, and atypically persistent. Whatever trouble these subjects had with the interface, less interested and less experienced subjects would have had more.

5.2 Tasks

The tasks given to the subjects were to determine answers to three questions with data from a database. The tasks were expressed in general terms and given in spoken language, and if the subject knew the language well enough, given in Swedish. The idea was to have a well defined task, but with a minimal amount of linguistic information useful for task solving in the task formulation itself.

Employees report to managers Employees work for managers Managers are employees Managers have departments Departments have managers Employees work at departments

Figure 3.8 Conceptual model of the application

The database contained information about a programming project. The way the data was organized can be seen in figure 3.8, where a conceptual model over the database is shown. The parts of the database that was used in the experiment concerned employees and managers, and consultants, and programming languages.

The subjects were told that the database was a managerial database over a programming project. They were told that the database contained information about "managers, employees, consultants, programmers, programs, departments, locations, and that sort of thing" and this seemed sufficient for them. "Determine which manager is the top manager." The way to solve this task is to list employees, ask for their manager, and then asking for the manager that the manager in question reports to, and to thus chain one's way up through the hierarchy. The distance from manager to the employees lowest in the hierarchy is five, and the number of employees in the database is 123. Figure 3.9 shows the part of the conceptual model that applies to this task.

Employees report to managers Employees work for managers Managers are employees Managers have departments Departments have managers Employees work at departments

Figure 3.9 Conceptual model of the application: managers The organization modeled in the database is strictly hierarchical. Every employee reports to a manager. Every manager reports to a manager as well. Every employee reports to one manager only. In the database, an employee may have several managers, reflecting the fact that managers have at least a manager to report to, and may have managers reporting or belonging to them.

Another way to get the answer in one single query is to ask for a manager with no manager. This method was used by some subjects. This is a natural way to obtain the answer, but it is worth noting that the question "Which manager has no managers" is not likely to be heard outside this application. The naturalness of the construction is debatable.

Subtasks to solve here include

a) find out about the hierarchial organizational structure b) find term "manager" c) realize that there is no separate term defined for "top manager" d) realize that the department the top manager works in has not special term attached to it e) find term "report to" relating employees to their managers or investigate the construction "manager of NN" which carries two meanings: the manager NN reports to and the managers that report to NN. "Determine which consultant is the tallest and what he or she is paid." The database contains four consultants who work for departments. The consultants have heights and expenses.

Consultants cost expenses Consultants have lengths Consultants have consultant lengths

Figure 3.10 Conceptual model of the application: consultants

In this task the main problem is that consultants, according to the conceptual model, are not paid and do not earn money and are not given salaries: consultants cost money, or have expenses. Some of the first subjects were asked to determine which consultant is paid the most, but the task was reformulated for the later sessions. Subtasks to solve for this query include

a) find term "consultant" b) find term "consultant height" or "tall" or "height" c) realize that consulants do not have salaries but costs d) find term "cost" or "expense" e) chain the two subqueries into one query

A way to obtain the answer would be to ask "List consultants with height and expenses." "Determine which programming language has the most code written in it." The database contains a list of programs. For each program, the database contains information about its size and the programming language used in it. It turned out to be difficult to add the sizes of programs together, so a sufficient answer

to this query was to find the language with the largest number of programs written in it.

Programmers are employees Programmers work in languages Programmers work on modules

Modules are written in languages Modules have sizes

Figure 3.10 Conceptual model of the application: languages and modules

Subtasks include:

a) find term "program" or "module" b) find term "language" c) list number modules for each language

A query that would produce a useful answer is "List number of programs per language."

5.3 Experiment Setup

The experiments took place at IBM Nordic Laboratories on Lidingö. After the first two sessions, which were less formal and took place in a normal office environment, with the test conductor and subjects in the same room, the remainder of sessions were conducted in a test laboratory specially equipped for usability testing on the IBM premises. The subjects were seated at a computer terminal and given one of the tasks to solve vocally. No written instructions were provided, and if the subject knew Swedish well enough, the task was given in Swedish; otherwise, the task was given in English. In both cases, the task was given in informal terms. When the subject indicated satisfaction with how the task was solved, or decided that the task was too difficult, another task was given. Some subjects worked through all three tasks, but most only worked on two during the hour scheduled for the sessions.

The sessions were logged by means of videotape recordings and keyboard log programs, and observed through a one way mirror and a slave monitor from a room adjacent to the experiment terminal. The subjects were encouraged to think aloud while solving the task, so that any comments they would have would be registered on the videotape. Most did not, however, and in order not to distract their attention from the task they were not reminded to do so.

The terminal was set up with a database interface in one screen window, and a help prototype session in another, with switching between the windows by mouse clicks.

The subjects were not informed of what was being tested, but most believed that the study concerned grammar coverage or a help system prototype, or that the study was a general usability study. If the subjects asked questions about the study, they were told that the test concerned grammar coverage. They were led to believe that the object of the study was to solve the tasks rapidly, and to complete all of them. After the sessions, the subjects were given more information on the focus of the study, and were asked for comments. In some cases, specific features of their session were discussed.

5.4 Results

The study shows some related things. Firstly, users use a restricted language. Too restricted in fact: their expectations are lower than reasonable. Some use a formally oriented syntax - "computerese", as shown in figure 5.2 - some use strange but functional language, and most tend towards safe and conservative choices. Secondly, users adapt their language to what they believe the system prefers. Thirdly, when given direct evidence of the system competence, they willingly use it, and adapt their language to it.

The two influential factors on interaction discussed in sections 3 and 4 are of course difficult to separate, as has been discussed previously. In the next few sections

some results and typical interactions are reported. The results are analyzed in terms of modalities and counterpart attitudes in the last two subsections of this section.

The formal evaluation of results from a study this small poses some problems of generalizability, as has been discussed above; another, more general problem in analyzing this type of behavior is that it is difficult to claim things about the success of interaction. If one user poses twelve queries to receive an answer, and another achieves the same result with one query, after having failed twice, the first user has a 100% success rate, while the other user has 33% - but arguably the second user has been more successful in the interaction. Below, in figure 5.1, the sessions are tabulated for numbers of attempted and successful queries, and for help access. Successful is here taken to mean as having produced a database table result: the attempts that did not succeed may have been canceled by the user as well as not being understood by the system.

```
Session Number of  Number of  Number of  Number  Help access
successful  attempted  failed  of tasks Vocabulary Phrase
queries queries  tries solved lookup generation

1 40 45 5 2 -
2 27 49 22 2 -
3 11 55 44 2 12
4 28 46 18 0 -
5 14 77 63 2 -
6 21 55 34 0 -
7 18 47 29 2 3  2
8 14 21 7 3 2
9 9 24 15 3 8  7
10 22 27 5 2 1
11 16 39 23 1 -
12 5 23 18 2 4  5
13 9 26 17 2 -

 234 534 300
```

Table 5.1 Number of queries, tries, and help accesses

The help prototype provided for the test subjects was a panel with commands to inspect the vocabulary of the system, and to generate relation phrases and sample queries for terms in the user vocabulary, i.e. the vocabulary defined in the conceptual model. All subjects for whom it was made available did not feel the need to use the help prototype, and most did not use all functions available. Figure 5.1 shows how many of the users had access to the system and how they used it. Those subjects that discovered the sample query and the relational phrase generators did use the system enthusiastically and used the help system at the beginning of each task, in an explicit learning phase, instead of trying to solve the problem without help first.

5.5 User's Initial Expectations on the System

Users' expectations on system competence are most likely colored by their previous experiences with computers. Some users used a telegraphic syntax when interacting with the system, and some tried to use a SQL-like syntax. Some examples of syntax that has been judged specific for human-computer interaction are shown in figure 5.2[14]; most users did not, however, use this type of constructions. Of the thirteen

[14]In this as in following figures "Try" stands for a query that was not interpreted correctly or canceled by the subject; "Qu" stands for a query that was processed successfully, i.e. produced

sessions, six contained queries that had a telegraphic quality. Of all the 534 queries, about 5% were of this type. Some other unusual and unidiomatic constructions were used: however, they have to do with the unusual tasks and unusual interaction modes: there are few ways of posing a query like the ones shown in the latter half of figure 5.2 in an idiomatic way. The way the queries are posed has to do with the task and the mode of interaction, where the subject expects that the whole query has to be output in one turn, without possibility of elaboration, and also expects that the system can cope with the type of complex constructions that result.

```
Examples of telegraphic syntax used:

Try How many employees
Try Which is hq
Qu list managers departments location
Qu location
Qu managers that have managers
Try length of consultants
Try salary of anders andersson
Qu managers with no manager
Examples of other unusual constructions used:
Try select manager where manager does not have manager
Try Is there a manager of all departments
Try which manager is the manager of the manager named eva cruse
```

Figure 5.2 Unusual syntax used by subjects

As has been argued in previous sections, the initial attitudes of users are not as interesting as how they change. As an extreme example, the first queries of some test subjects are tabulated in figures 5.3 and 5.4. The queries in figure 5.3 produced successful answers, while the ones in figure 5.4 did not.

```
Qu managers?
Qu list departments
Qu who are the consultants?
Qu list manager with highest level
Qu Give me all managers
```

Figure 5.3 Successful first queries:

These first queries vary, both in vocabulary and in syntax. The ones that were not understood by the system used unexpected syntax in the first case, or, in the others, assumed terms not defined. The ones that succeeded varied as well: they have a varying syntax, in some cases not compatible with the written norm for English. The point here is that the first expectations of a group of computer literate users are low: the syntax is uniformly rather simple, a feature which will persist through the session, and none of the subjects professed any frustration with the fact that their first few queries were not understood by the system. Users will probably accept a failed query or two, initially, if they can be understood as part of a learning process.

```
Try How many employees
Try top manager
Try I want the chief of department software
Try who is project manager
Try Who is the top manager?
```

an answer. “Par” stands for a system generated paraphrase.

Figure 5.4 Examples of unsuccessful first queries:

Is there something to learn here? Probably not. That computer professionals have low expectations of computers' flexibility is probably no surprise: the results would have been very different if a different group of subjects had been used. The expectations of the user community will also vary wildly depending on what computer systems they have encountered previously. A user who never has used computers before, but heard all kinds of stories about their capacity might expect more, as will a user who has used several well functioning natural language interfaces elsewhere, which probably will be the case in the not too distant future. A study on first queries will give interesting results about the attitudes towards technology, and should probably be done on its sociopsychological merits alone, but will probably give little useful information for grammar and dialog designers.

Later in the study, encouraged by success, occasionally the users wished to utilize a construction which was not supported by the version of the system used in the experiment. In figure 5.5, the subject unsuccessfully tried to use a universal quantifier fronted in a clause.

```
Try  For each language get the sum of sizes of modules
Error msg Could not analyze the question.
```

Figure 5.5 A complex construction

This was an example of a construction not covered by the system, and even here, the subject was wary of it, and asked the test conductor[15] if the construction would work before trying it. In general the users were careful of using complex constructions, and as the first query examples showed, choice of vocabulary was typically the main cause for interaction breakdown.

The inverse question is to investigate which of the system's capabilities are such that the users do not expect them, and that thus run a risk of being underutilized or even completely left unused.One example are constructions like the one in figure 5.6. The system allows users to refer to entities mentioned in previous queries by pronouns, to facilitate subsequent ones, a mechanism which is useful even for reasonably simple queries. As has been reported by several other studies and discussed in the previous sections, users tend not to use pronouns in interaction with the system. This system allows users to refer to entities in previous queries as exemplified in figure 5.6.

```
Qu 1 List managers.
Qu 2 What departments do they belong to?
```

Figure 5.6 A type of complex construction not commonly used by subjects

This type of construction was only used in two sessions, in both cases after the subjects had been given a hint that it was possible. In the example shown in 5.7 - taken from the first session, in the experiment where the test subject and the test conductor both were sitting in the same room - the subject is trying to find which salesperson sells the most and what that person sells. Only after having been given a hint by the test conductor does the user try referring to the previous query.

```
Qu who has the highest commission
Par Find employees that have the maximum commission
that belongs to employees.

A name is displayed on the screen.
Subject asks test conductor:
```

[15]The subjects had a possibility to communicate with the test conductor over a microphone link.

```
- Do I have to remember his name?
Test conductor replies:
- Try a pronoun.

Qu what does he sell
Par Find products that are sold by employees that are
salesreps and commissions that are the maximum
  commission that belongs to employees belong to.

Subject: - Wow. How did it do that?
```

Figure 5.7 Using pronouns takes some coaxing

In summary, the users were careful not to exceed the capabilities of the system and their initial expectations of what syntax the system coped with was low. However, the interesting question is not what users expect from a system at the start of a session - after all, this is a new type of system, and users cannot be expected to know beforehand what the system handles - but what they learn to expect from it during the course of a session.

5.6 Change in User's Expectations

The subjects in this study were conscious of the fact that they were learning a language during the session, and commented on it after the sessions - one subject said: "After a while you would get into the dialect." This learning process, whether conscious or unconscious, manifested itself in obvious ways in the interactions. The natural language feedback available to users from the system was the paraphrase list and the language the users requested that the system generate in the help system. The most obvious natural language feedback is the paraphrase list: the paraphrases appear for each query, and seem to be the system's own language.In fact, this is not what the paraphrases are designed to be: they are designed to display unambiguous readings for the query, not to teach the user the system language.

```
Qu  list manager with highest level
Par Find employees that are managers ...
Qu  list departments
Par Find departments.
Qu  list managers
Par Find managers.
Qu  list level
The subject changes the query.

Qu find level
Par Find levels.
```

Figure 5.8 Lexical transfer from paraphrase

Paraphrases clearly do influence users. Some subjects commented on the paraphrases and explicitly mentioned their influence: "This is the kind of language the system likes, right?". On one level, the paraphrases influenced the lexical choice of the user. For instance: users made use of all kinds of command verbs: "find", "show me", "display", "list", etc in the beginning of the sessions, but by the end, most users had switched to "find", which is what is used in paraphrases. This effect, of system vocabulary leaking into the interaction has been reported in other studies as well, as was discussed in section 3. In figure 5.8, taken from the very beginning of a session, the subject first asks three questions using "list" and for each one receives "find" paraphrases. For the fourth question, he starts by typing "list level", but

then, before pressing "Enter" to send off the query, goes back and changes to "list" to "find". He then stays with "find" for the duration of the session. After the session, when asked about the verb choice, the subject reported having the feeling that the system preferred "find" to "list".

```
Qu Find managers reporting to Mike Jones
Par Find managers that report to managers named mike jones.
Qu find managers
Par Find managers.
Qu  Find managers reporting to Eva Cruse
Par Find managers that report to managers named eva cruse.
Qu Who works for Mike Jones
Par Find employees that work for managers named mike jones.
Qu Find managers that report to Mike Jones
Par Find managers that report to managers named mike jones.
```

Figure 5.9 Syntactical transfer from paraphrase

On a syntactic level, the effects are less obvious, but there are some examples in the material. In the case shown in figure 5.9, about twenty minutes into the interaction, the subject asks a series of questions about managers that report to other managers. The latter constructions use the same syntax as the paraphrase. Later in the session the subject reverts to the first construction again, however.

5.7 Users Search for Guidance When the Interaction Does Not Work

In the preceding section it was shown that users absorb behavior from paraphrases. If the query that the user enters is not understood by the system, either for syntactic or lexical reasons, no paraphrases are shown. If there is no help available and the error messages are no help, the only method the user has to succeed in getting a query through is to try and try again. If there are examples of language available, the users will try to emulate that language. In figure 5.10, the subject was trying to establish the name of the consultant with the most income. The conceptual schema[16] defined for this database does not accept that consultants earn money or have incomes: the way to obtain a useful answer would be to ask for a list of consultants sorted by expenses. Under the conditions in this session, the help prototype was not available: the only help is a fixed help panel with a series of example queries that are not specifically designed for the task and the current database. In the end, the subject finally asks a query which would not be directly relevant to solving the task as it had originally been formulated.

```
Qu who are the consultants? Produces a successful answer.

Try who are the consultants and what do they earn
Try who are the consultants and what is their wage
Try who are the consultants and what is their sallary
Try who are the consultants and what is their salary
Try who are the consultants and how mush does each earn
Try who are the consultants and how much does each earn
Try who are the consultants and how much is each paid
Try how much is each consultant paid
```

[16]The database used in the example is a very small example database, and the schema an even smaller representation of the knowledge in it. Naturally, in a practical situation, a schema would be defined with more care.

```
Try how much does each consultant earn
Try how much is each consultant's salery
Try how much is each consultant's salary
```

The subject consults the list of example queries. They include "What is the average salary of a manager?"

```
Try what is the salary for the consultants
Try what is the wage for the consultants
Try what wage does each consultant receive
Try what salary does each consultant receive
```

The subject again consults list of example queries. They include "Show the salesreps with the three highest commissions."

```
Try show the consultants with the highest salary
Try show the consultants with the highest wage
Try show the consultants with the highest income
```

The subject again consults list of example queries.

```
Try what is the salary of each consultant
Try what is the income of each consultant
```

The subject again consults list of example queries. As before, they include "What is the average salary of a manager?"

```
Try what is the average salary of a consultant
Try what is the average salary of the consultants
```

Figure 5.10 Transfer from examples when the conceptual schema is rudimentary

5.8 Paraphrases as Feedback

If the syntax of a query is syntactically and lexically acceptable to the system a paraphrase is produced. In some cases the system tries to interpret the query even when it is faulty or misinterpreted in some way. In these cases the paraphrase helps users determine if the query was interpreted correctly.

```
Try whhich are the managers
Par Are there any managers named whhich?
```

Figure 5.14 Typing error discovered by paraphrase inspection

In the two cases in figures 5.14 and 5.15 the typographical and spelling errors that the subjects entered were interpreted as names by the system, and by inspecting the paraphrase, the subjects were able to correct the queries.

```
Try How much is Anders Andersson payed
Par Find the sum of the total sizes that belong to modules named anders andersson payed.
```

Figure 5.15 Spelling error discovered by paraphrase inspection

In figure 5.16, the subject used a word not present in the conceptual schema. The system again tried to interpret the word as a proper name [17], and when shown the paraphrases the subject canceled the query and rephrased the question.

[17] The word "charge" was highlighted in the query window, to indicate that the system interpreted it as a proper name.

```
Try  Who is in charge?
Par Find modules in languages named charge.
Find employees in departments named charge.
Find departments in locations named charge.
Find modules in load modules named charge.
Find employees in locations named charge.
```

Figure 5.16 A paraphrase as a good reason to cancel a query

In some cases the qualities of the paraphrase grammar make this feedback method difficult to use. In figure 5.17 the subject asks for the language of each module, and when the paraphrase in the figure was shown he canceled the query, not expecting it to produce useful results. In fact, the query would have given him the result he sought.

```
Qu Show me the language of all modules
Par Find modules that have languages.
```

Figure 5.17 A paraphrase as a bad reason to cancel a query

Users paid attention to paraphrases because of their status as the sole conveyor of information - albeit idiosyncratic - of the system's linguistic competence and of the system processes.

5.9 How To Use Feedback - The Help Prototype

Most subjects used the prototype help system for dictionary lookup. In figure 5.18 the subject is trying to identify the tallest consultant. She first tries once with no success and decides to consult the help prototype. She inspects the vocabulary. The vocabulary displayed includes "height", "maximum", and "length". She tries the terms, but receives unexpected error messages, and returns to the help prototype. She finds the term "consultant length" in the user noun list, and searches the user verb list for "earn" which she does not find. After a couple of more tries she receives an answer. In this example the vocabulary lookups help the subject to formulate the query.

```
Try Give me consultant with max lengt
Error msg Could not analyze the question.

Help Inspect vocabulary:
Nouns
All words

Try Give me consultant with maximum height
Try Give me all consultants and their lengths
Try Get all consultants and their lengths
Error msg Cannot work out what the pronoun their refers to.

Help Inspect vocabulary:
Pronouns

Try Get the name and length and salary of all consultants

Help Inspect vocabulary:
Nouns
Verbs
```

```
Try Get the name and maximum consultant length of all
consultants
Try Get the name and maximum consultant length
Qu Get the name and consultant length of all consultants

Answer
CONSID  FNAME      LNAME            LENGTH
------  ---------  ---------------  ------
1       MAUD       GREEN            172
2       MONA       GLAZIN           178
3       PETER      BROWN            176
4       ANDERS     ANDERSSON        186
```

Figure 5.18 Vocabulary lookups [18]

Several subjects used the help system to generate relational phrases from the conceptual model, to examine what relations a word enters into. In figure 5.19, the subject is trying to find out which consultant has the highest expenses, and after trying once with "salary" takes recourse to the help system to receive an answer. As has been described above, the conceptual model does not recognize consultants as receiving salaries: consultants have expenses and costs associated with them.

```
Try give consultant with higher salary

Help Generate phrases for ‘‘consultant’’

List includes:
Consultants have expenses

Try give consultant with higher expenses
Try give consultant have higher expenses
Qu give expenses of consultants
```

Figure 5.19 Using relational phrase generation

Some subjects tried generating sample queries. In figure 5.20, the user is trying[19] to find out the expenses associated with the tallest consultant, and has already tried using "earn", and generating sample queries for "salary" without success, and is now, after finding "expensive" in the base vocabulary list, using questions

```
Try how expensive is the tallest consultant
Try Who is the tallest consultant and how expensive is he

Help Vocabulary search
Includes
expensive
Phrase generation (for various terms)
Includes
What was costed by consultants
Tell me who costs expenses
More vocabulary search
```

[18] Since these examples were collected the system coverage has been extended, and the third try would produce the answer table directly.

[19] Since these examples were collected the system coverage has been extended, and the help prototype would produce phrases with a standard syntax.

```
Try what was costed by the consultant with highest length
Try which consultant costed most
Try which consultant cost most expenses
Qu what was costed by consultants
```

Figure 5.20 Using sample query generation

with "expensive" with no success. He then consults the help system, and after seeing the consultant sample query list which includes: "what was costed by consultants" and "tell me who costs expenses" he starts using constructions with "costed"[20], eventually using a direct quote from the sample query list and then, later, finding out separately who the tallest consultant was.

The subjects who used the help system most started each task by requesting help, rather than trying to figure out by trial and error. In figure 5.21 the subject has just completed a task successfully, using the relational phrase generation function from the help system. She has been given a new task, to find out which programming language has the most code written in it. She begins by consulting the help system and gets off to a flying start in the interaction. This way of using the help system not for help but for learning is good indication of the usefulness of the system, not only in objective terms, which, as has been reasoned elsewhere in this study, can be misleading, but in perceived usefulness by users

```
Help Generate phrases for ‘‘language’’
Generate phrases for ‘‘code’’
Generate phrases for ‘‘program’’

Try Get name and size and language all programs
Qu Get name and size and language of all modules
```

Figure 5.21 Beginning by help

5.10 Summary of Results

We can now address the questions posed in section 3.5 using the results obtained from the study and the modality space defined in section 3.3:

- Where in the modality space will the user start an interaction: what are the initial expectations?

The study showed us that users tend to use language, as shown in 5.4.1, which mirrors their expectations on the system competence; for most of the subjects and most of the time, these expectations were too low.

- How does the user learn where in the modality space the dialog with the system can be performed and how should the system make the user aware of a satisfactory position?

Users adapt their language to what they believe the system prefers, as shown in section 5.4.1: when given direct evidence of the system competence, they willingly use it, and adapt their language to it, as shown in section 5.6. In the current system, the most visible natural language feedback is the paraphrase list: the paraphrases appear to the user as if they were the system's own language. As shown in section 5.4.5, users paid attention to paraphrases to learn what the system was doing, and at times picked up linguistic items from the language used in them, as shown in

[20] "Costed" can hardly be assumed to be in most users' language to begin with. This belies the analysis of interaction languages as sublanguages of users' language.

section 5.4.2. This is not always functional: the paraphrases were not designed with this in mind. When the users had access to the help prototype, they used it to learn what the conceptual model contained, as shown in 5.4.6, thus broadening the interaction. The users tried hard to make the interaction as interactive as possible. This tendency should be made use of, in that the users watch the system for clues of how to act; the help prototype, which raised the level of interactivity, showed, as in the examples in section 5.4.6, that this is a useful strategy. Even when the users simply used it for meta-queries, like in section 5.4.6.1 on the vocabulary, it helped them formulate the query faster than they would have been able to otherwise; when used for higher-level inspection, the subjects learnt to rely on the system very rapidly, as shown in 5.4.6.2 and 5.4.6.3.

- What difficulties are inherent in the mode that typically is used for human-computer dialog?

The structure of the dialog in the sessions was extremely simple. The subjects input a query, and received an answer: practically all exchanges were two-part. This restriction was not necessary: as indicated in section 5.4.1 the system is capable of intepreting follow-up questions, but this capability was never utilized by the subjects. The unnecessarily low interactivity of the modality gave the users no support for acquiring a complete picture of the system competence.

The main problem with using written language at a terminal, however interactive, is that users may not expect the dialog to be conversational. If the user does not expect this, there is no way to lead the user into a conversational type of interaction except by explicitly pointing out that it is possible to refer to earlier queries and their answers. There is no natural way to broaden the dialog, if the user believes it is as broad as it can get.

6 Concluding Remarks

6.1 A New Modality

The most obvious point in connection with modalities is that learning a new one, a new standard for communication, is not an easy task: we spend the first three or so years of our individual lives bootstrapping the spoken interactive modality and the rest of our lives perfecting it, and collectively we have spent millions of years developing the system. We spend at least twelve years of schooling - blood, sweat, and tears - to learn written non-interactive communication, with varying degrees of success: some people never learn, and prefer not to write if they do not absolutely have to. Today, with computers entering the stage, completely new modalities, or new positions in the space of many others, different from the ones we know, have to be dealt with, notably the one described in this study: written channel, but in a somewhat interactive mode. So far, one of the most specific features of written modality is that it is a relatively non-interactive mode of communication: this is changing, and not only because of natural language interfaces, but because of other technological advances as well.

Communicating over a new modality is difficult, and indeed frustrating for many people. Some people have trouble using telephones, and several otherwise communicative people refuse to use telephone answering machines: spoken non-interactive communication is too difficult or strange for them to feel comfortable leaving messages on a tape which gives no feedback. Empirical studies aiming at investigating properties of dialogs with computerized telephone directory information services find that a large percentage of callers simply hang up when they discover a machine at the other end.

Kerstin Severinson Eklundh notes[12] in reporting a study on computer mediated human-human communication that it is a new mode of communication, lacking in conventions, and that the type of communication is perceived as speech-like, in a similar mid-position that Chafe observed[5] for oral literature - between spoken and written language. The choice in language should not be seen as something between spoken interactive and written non-interactive, but between a wide range of available options. New technology will provide more options, but there have always been more than just simply written and spoken: different genres and different registers of language use have different degrees of interactivity. One measure of this might be pronoun usage. As has been pointed out[10] by Dahlbäck, pronouns are comparatively scarce in human-computer dialogs. This can partly be related to the specific features of computers, but as the number of pronouns varied across various dialog types - advice giving dialogs have more pronouns than simpler database queries - something in the dialog itself can be assumed to influence the number of pronouns. It may be the degree of interactivity: in highly interactive spoken dialogs, the number of pronouns is high when compared to less interactive dialogs.

The consequences for system design are two. First, that users should not be expected to jump to written interactive communication without initial problems: new modes of communication are hard to get used to, especially if the mode lacks well established conventions; and second, users will need more guidance than might be expected. In terms of the notion of modality space and the diagrams sprinkled through the text, the user needs help to position the dialog in the correct position, or in one of the correct positions within the space that the system can communicate in. Severinson Eklundh notes[12] that new users on the COM conference system may behave as if the communication is more interactive than more experienced users do, and Dahlbäck notes[10] that users had some trouble ending dialogs with a system: with no logoff command, the way to close an interaction was unclear to inexperienced users. It takes some time to catch up with the conventions of interactivity that evolve in a community of users. The important point here, as with the conceptual, syntactical, and lexical properties of language used, is that the model must be learnable and understandable to the user. Secondly that the modalities involved will not necessarily be similar: the degree of interactivity, and the combination of textual and diagram elements in the dialog will probably lead to a specific dialog model necessary for every type of interaction.

6.2 Specificity of Machines as Counterparts

So what do users expect? For instance, as has been discussed in 3.1 and 3.3, it has been shown that users try to be more careful about how they formulate their queries and instructions when communicating with computers than when communicating with humans: they expect computers to be picky. This is an explicit prejudice. An implicit prejudice, a prejudice on a different level, is that computers only cope with simple dialog structures and that users have full control over the conversation: a user expects the machine to answer identically to two questions posed in succession if the questions are identical. The model the user has seems to be that the machine is a retrieval system, and that an input by the user retrieves an output by the machine. This is in stark contrast to what users expect from people: a librarian asked the same question twice, would be expected to produce two different answers.

As long as systems are question answering systems this is fine. The problem is that a too simple dialog structure will leave interesting and useful capabilities of the interface system unused as, which has been showed above in section 5.4.1, was the case of pronoun usage. Again, as has been discussed above, the initial attitudes are not as important as how they are modified: the problem here is not the level and method of description of dialogs, or the fact that users seem to model computers

in a certain way, but the fact that users have no way of learning, save by trial and error, that this specific counterpart will be able to handle a certain mechanism. This is a question of learnability. For instance, how will users learn what is current and referrable in the current context of the machine? Today, most users in studies seem to assume that the machine has no context but the one the user has explicitly formulated in the last query - this is already an unjust prejudice, as in the case of SAA LanguageAccess; how users could be shown the anaphoric competence of a system is a difficult question.

6.3 Lowering Precision to Obtain Better Understanding

What consequences for system design should be extracted from this study? The first main point is that the channel and mode of interaction should be chosen and designed in such a manner that the user will be able to both use natural linguistic mechanisms, and rapidly learn conventions for use. This is important because of the second main point: the system should make sure that the user will easily be able to form a model of how and what the system understands, on the basis of feedback the system produces.

```
Try how many employees has the company
Error msg Could not analyze the question. ‘‘company’’ is highlighted
Qu How many employees are there
Par Find the number of employees. Produces answer
```

Figure 6.2 Failed interaction?

Exactly how this should be done is of course a matter for further studies and careful engineering considerations, but one important general principle is that the interaction should be widened and dispersed, to ensure that the interaction is not just a query and an answer, but rather a combination of marginal or introductory text to discuss and clarify concepts, and follow up questions. This way the informational content of each successful query is less and the interaction will be more successful as a whole. An example may clarify this principle. If a user poses a query to a system, receives an error message, reformulates the query, and receives the answer looked for, as in figure 6.2, it might be construed as a 50% successful interaction. Compare this to the example conversations in figures 6.3 and 6.4: do they contain "failed" turns? In actuality, even though the answers took two turns to produce, arguably every turn in the interaction was useful.

```
In a delicatessen:
Customer I would like a drumstick, please.
Shopkeeper What do you mean, ‘‘drumstick’’?
Customer The leg of the chicken. That part.
Shopkeeper Oh. Here you go. Two forty, please.
```

Figure 6.3 Failed interaction?

```
In a delicatessen:
Customer:  The leg of the chicken. That part.
Shopkeeper One drumstick, here you go. Two forty please.
Customer Oh, wait. Make it two drumsticks.
Shopkeeper Sure. Here you are. That’ll be four eighty.
```

Figure 6.4 Failed interaction?

```
- What is the telephone number of Anders Andersson in Nacka?
- Anders Andersson in Nacka ... one moment. There are two of them,
```

```
  or he may have moved recently. Address?
- Ugglev\"agen.
- Ugglev\"agen. The number is 718 90 00.
```

Figure 6.5 Successful interaction

```
- See the game last night?
- Yeah! They really showed them. Way to go!
- Yup. Can you lend me twenty until tomorrow?
- Sure.
```

Figure 6.6 A tuning process

By widening the dialog, as some subjects in the study did by starting an interaction by using the help system to generate phrases as described in section 5.4.6.3, they lessened the informational content of each turn taken, but they heightened the efficiency of the interaction by gaining more material on the system's way of thinking and formulating the area. In human-human conversation the denseness of information in turns is lower than might be expected at first thought: the interactions in figures 6.4[21] and 6.5 exemplify this, by diluting the information density of the dialog while at the same time showing the functionality in this tuning process. As was discussed in the previous section, it is difficult to display context and competence: the easiest way to do this is to use the natural human conversational mechanisms for tuning and learning.

References

[1] Lars Ahrenberg, Nils Dahlbäck, and Arne Jönsson, "Samtalsanalys av (och för) människa-dator-interaktion" in Samtal och språkundervisning, Studier till Lennart Gustavssons minne, Ulrika Nettelblad & Gisela Håkansson (ed.), Linköping Studies in Arts and Science 60, Linköping, 1990

[2] Susan E. Brennan, "Conversation with and through Computers", User Modeling and User-Adapted Interaction 1 (1991) 67-86

[3] Ivan Bretan, "Enhancing the Usability of Natural Language Interfaces with Direct Manipulation", Master's Thesis at the department of Computational Linguistics, University of Gothenburg, Gothenburg 1990

[4] Gustaf Cederschiöld, Om svenskan som skriftspråk, C. W. K. Gleerups förlag, Lund, 2:a upplagan 1902

[5] Wallace L. Chafe, "Integration and Involvement In Speaking, Writing, and Oral Literature", in Deborah Tannen (ed.) Spoken and Written Language: Exploring Orality and Literacy, Ablex, Norwood, 1982

[6] Alphonse Chapanis, Robert B. Ochsman, Robert N. Parrish, and Gerald D.Weeks, "Studies in Interactive Communication: I. The Effects of Four Communication Modes on The Behavior of Teams During Cooperative Problem Solving", Human Factors 14:6 (1972) 487-509

[7] Alphonse Chapanis, Robert N. Parrish, Robert B. Ochsman, and Gerald D.Weeks, "Studies in Interactive Communication: II. The Effects of Four Communication Modes on The Behavior of Teams During Cooperative Problem Solving", Human Factors 19:2 (1977) 101-126

[21] Conversation with the Stockholm phone enquiry, on May 3, 1992, 21:25. The name and number of the household in the example have been changed to protect the integrity of those involved.

[8] Noam Chomsky, Aspects of Theory of Syntax, The MIT Press, Cambridge, Massachusetts, 1965

[9] Philip R. Cohen, "The Pragmatics of Referring and the Modality of Communication", Computational Linguistics 10:2 (1984) 97-147

[10] Nils Dahlbäck, Representations of Discourse - Cognitive and Computational Aspects, Linköping Studies in Arts and Science 71, Linköping Studies in Science and Technology 264, Doctoral thesis at Linköping University, 1991

[11] Nils Dahlbäck and Arne Jönsson, " A System for Studying Human Computer Dialogues in Natural Language", Research Report, Departement of Computer and Information Science, Linköping University, LiTH-IDA-R-86-42, 1986

[12] Kerstin Severinson Eklundh, Dialogue Processes in Computer-Mediated Communication, Linköping Studies in Arts and Science 6, Doctoral thesis at Linköping University, 1986

[13] Susan R. Fussell and Robert M. Krauss, "Coordination of Knowledge in Communication: Effects of Speakers' Assumptions about Others' Knowledge", manuscript from Columbia University, Department of Psychology, New York, 1990

[14] H. P. Grice, "Logic and Conversation", in P. Cole and J. Morgan (eds), Syntax and Semantics 3:Speech Acts, Academic Press, London 1975 41 - 58 (Manuscript of lecture delivered 1967)

[15] Raymonde Guindon, Kelly Shuldberg, and Joyce Conner, "Grammatical and Ungrammatical Structures in User-Adviser Dialogues: Evidence for the Sufficiency of Restricted Languages in Natural Language Interfaces to Advisory Systems", in Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics, Stanford, 1987

[16] M. A. K. Halliday, Language as social semiotic, Edward Arnold Ltd, London, 1978

[17] Zellig Harris, Mathematical Structures of Language, John Wiley & Sons, New York, 1968

[18] Patrick A. Holleran, "A methodological note on pitfalls in usability testing", Behaviour & Information Technology 10:5 (1991) 345-377

[19] International Business Machines Corporation, IBM SAA LanguageAccess General Information, Publication Number SH19-6680, 1990

[20] Ellen A. Isaacs and Herbert H. Clark, "References in Conversation Between Experts and Novices", Journal of Experimental Psychology: General 116:1 (1987) 26-37

[21] M. Jarke, J. Krause, Y. Vassiliou, E. Stohr., J Turner, and N. White, "Evaluation and Assessment of a Domain-Independent Natural Language Query System", A quarterly bulletin of the IEEE computer society technical committee on Database Engineering 8:3 (1985) 34-44

[22] Otto Jespersen, The Philosophy of Grammar, George Allen & Unwin Ltd, Woking, 1924

[23] Aravind K. Joshi, "Mutual Beliefs in Question-Answering Systems", in N. V. Smith (ed), Mutual Knowledge, Academic Press, London 1982

[24] Jussi Karlgren, "Report on a Study of Natural Language Interfaces", IBM Nordic Laboratories, Lidingö 1991

[25] Alan Kennedy, Alan Wilkes, Leona Elder, and Wayne S. Murray, "Dialogue with Machines", Cognition 30:1 (1988) 37-72

[26] Erik Knudsen, Grammars, Parsing, and Logic Programming, SYSLAB Report 63, University of Stockholm, Sweden

[27] Robert M. Krauss and Susan R. Fussell, "Perspective-taking in Communication: Representation of others' knowledge in reference", in press, Social Cognition

[28] Willem J. Levelt and Stephanie Kelter, "Surface Form and Memory in Question Answering", Cognitive Psychology 14:1 (1982) 78-106

[29] Ashok Malhotra, "Knowledge-based English Language Systems for Management Support: An Analysis of Requirements", System", in proceedings of International Joint Conference on Artificial Intelligence, 1975, 842-847

[30] George Mead, Mind, Self, and Society, University of Chicago, Chicago, 1934

[31] Hermann Paul, Prinzipien der Sprachgeschichte, Verlag von Max Niemeyer, Halle, 4th edition 1909

[32] Sanja Petrović, "Providing Help in a Natural Language Query Interface to Relational Databases", *Yugoslav Journal of Operations Research*, 2:2 (1992), pp 207-218

[33] Zenon W. Pylyshyn and Richard I. Kittredge, "Databases and Natural Language Processing", A quarterly bulletin of the IEEE computer society technical committee on Database Engineering 8:3 (1985) 2-9

[34] Michael L. Ray and Eugene J. Webb, "Speech Duration Effects in the Kennedy News Conferences", Science 153 (1966), 899-901

[35] Elaine Rich, "Natural Language Understanding: How Natural Can It Be?" , in proceedings of The Second Conference on Artificial Intelligence Applications, 372-377

[36] Mohammad A. Sanamrad and Ivan Bretan, "IBM SAA LanguageAccess: A Large-Scale Commercial Product Implemented in Prolog", to appear in the proceedings of the 1st International Conference on the Practical Application of Prolog, 1992

[37] Christer Samuelsson and Manny Rayner, "Quantitative Evaluation of Explanation-Based Learning as an Optimization Tool for a Large Scale Natural Language System", in proceedings of International Joint Conference on Artificial Intelligence, Sydney, 1991

[38] N. V. Smith (ed), Mutual Knowledge, Academic Press, London 1982

[39] Bozena Thompson, "Linguistic Analysis of Natural Language Communication with Computers", in proceedings of COLING 1980 190-201

[40] Jean Veronis, "Error in natural language dialogue between man and machine", International Journal of Man-Machine Studies 35 (1991) 187-217

[41] Tom Wachtel, "Pragmatic Sensitivity in Natural Language Interfaces and The Structure of Conversation", in proceedings of COLING 1986, 35-41

[42] Elizabeth Zoltan-Ford, “How to get people to say and type what computers can understand”, International Journal of Man-Machine Studies 34 (1991) 527-547