WIZARD OF OZ STUDIES — WHY AND HOW

Nils Dahlbäck, Arne Jönsson, Lars Ahrenberg

Natural Language Processing Laboratory
Department of Computer and Information Science
S-581 83 LINKÖPING, SWEDEN
Phone: +46 13 28 10 00
nda@ida.liu.se, arj@ida.liu.se, lah@ida.liu.se

ABSTRACT

We discuss current approaches to the development of natural language dialogue systems, and claim that they do not sufficiently consider the unique qualities of man-machine interaction as distinct from general human discourse. We conclude that empirical studies of this unique communication situation is required for the development of user-friendly interactive systems. One way of achieving this is through the use of so-called Wizard of Oz studies. We describe our work in this area. The focus is on the practical execution of the studies and the methodological conclusions that we have drawn on the basis of our experience. While the focus is on natural language interfaces, the methods used and the conclusions drawn from the results obtained are of relevance also to other kinds of intelligent interfaces.

1 THE NEED FOR WIZARD OF OZ STUDIES

Dialogue has been an active research area for quite some time in natural language processing. It is fair to say that researchers studying dialogue and discourse have developed their theories through detailed analysis of empirical data from many diverse dialogue situations. In their recent review of the field, Grosz, Pollack and Sidner [12] mentions work on task-oriented dialogues, descriptions of complex objects, narratives, informal and formal arguments, negotiations and explanations. One thing which these studies have shown is that human dialogue is a very complex activity, leading to a corresponding complexity of the theories proposed. In particular it is evident that participants must rely on knowledge and reasoning capabilities of many different kinds to know what is going on in a dialogue.

When it comes to using data and theories to the design of natural language interfaces it has often been argued that human dialogues should be regarded as a norm and a starting-point, i.e. that a natural dialogue between a person and a computer should resemble a dialogue between humans as much as possible. But different kinds of dialogues differ in complexity, and simple service encounters of the type that are likely applications for natural language interfaces presumably do not exhibit all complexities seen in other forms of dialogues. Furthermore, a computer is not a person, and some of the differences are such that they can be expected to have a major influence on the dialogue, thus making data from human interaction an unreliable source of information for some important aspects of design, in particular the style and complexity of interaction.

First let us look at some of the differences between the two dialogue situations that are likely to play a significant role. We know that language is influenced by interpersonal factors. To take one example, it has been suggested by R. Lakoff [20] and others that the use of so-called indirect speech acts is motivated by a need to follow “rules of politeness” (1. don’t impose, 2. give options). But will a user feel a need to be polite to a computer? And if not, will users of NLIs use indirect requests in the search of information from a database? If not, do we need a component in our NLI for handling indirect requests? This is obviously an empirical question that can be answered only by studying the language used in such situations.

Indirect utterances are of course something more than just ways of being polite. There are other forms, such as omitting obvious steps in an argument — relying on the listener’s background knowledge, and giving answers, not to the question, but by supplying information relevant to an inferred higher goal. But also the use of these will presumably vary with the assessed characteristics of one’s dialogue partner.

In the case of keyboard input another important factor is that the communication channel is different form ordinary human dialogues. The fact that the dialogue will be written instead of spoken will obviously affect the language used. As pointed out by Cohen [3] (p. 123) “Keyboard interaction, with its emphasis on optimal packaging of information into the smallest linguistic “space”, appears to be a mode that alters the normal organization of discourse.”

Much of our language behaviour, on all levels, from pronunciation to choice of words and sentences, can be seen as a result of our attempts to find the optimal compromise between two needs, the need to make ourselves understood, and the need to reach this goal with as little effort as possible. It is a well established fact in linguistic research that we as speakers adapt to the perceived characteristics of our interlocutors. The ability to modify the language to the needs of the hearer seems to be present already at the age of four [24]. Language directed to children is different from language directed to adults, as is the case when talking to foreigners, brain-injured people etc. There are good reasons to believe that similar adjustments can and will be made when we are faced with the task of interacting with a computer in natural language. One important consequence of this is that goals in some dialogue research in computational linguistics such as ’Getting computers to talk like you and me’ [23] or developing interfaces that will “allow the user to forget that he is questioning a machine”[10], are not only difficult to reach. They are misconceived.

Given these differences between the two types of dialogue and the well-founded assumption that they will affect the linguistic behaviour of the human interlocutors, it follows that the language samples used for providing the empirical ground should come from relevant settings and domains. In other words, the development of NLI-software should be based on an analysis of the language and interaction style used when communicating with NLIs. Since, as we just observed, users adapt to the language of their interlocutors, analysis of the language used when communicating with existing NLIs is of limited value in the development of the next generation systems. This is what motivates data collection by means of Wizard of Oz techniques, i.e. studies where subjects are told that they are interacting with a computer system through a natural-language interface, though in fact they are not. Instead the interaction is mediated by a human operator, the wizard, with the consequence that the subject can be given more freedom of expression, or be constrained in more systematic ways, than is the case for existing NLIs. (Some well-known studies based on a more or less ’pure’ Wizard of Oz technique are those of Cohen [3], Grosz [11], Guindon [13], and Kennedy *et al.* [19]. For a review and discussion of these and other studies, see [5, 18]. Fraser & Gilbert [9] provides a review focused on speech systems).

There are more than one reason for wanting to conduct Wizard of Oz-experiments. In our own work we have primarily been interested in characterizing the genre of Natural Language interface interaction. Another reason is to provide the empirical basis for the development of the software for a particular application. The importance of this has been stressed by Ogden [21] (p. 296), who claims that “The performance of the system will depend on the availability of representative users prior to actual use, and it will depend on the abilities of the installer to collect and integrate the relevant information”. Of course you cannot expect to gather all the data you need for the design of a given application system by means of Wizard of Oz studies, e.g. as regards vocabulary and syntactic constructions related to the domain. But for finding out what the application-specific linguistic characteristics are it is a valuable technique [17]. For gathering data as a basis for theories of the specific genre of human-computer interaction in natural language, the Wizard of Oz-technique seems to us to be the best available alternative.

The rest of this paper is concerned with a description of our work in the area of Wizard of Oz simulation studies. The focus is on the practical execution of the studies and the methodological conclusions that we have drawn on the basis of our experience. Some results on the characteristics of human-computer-interaction in natural language have been reported elsewhere [4, 5, 6, 7, 8, 18] and further work is currently in progress. Some of the major results obtained thus far is that man-machine dialogues exhibit a simpler structure than human dialogues, making it possible to use simpler but computationally more tractable dialogue models; that the system relies on a conceptual model specific for the domain *but* common to all users, i.e. uses mutual knowledge based on community membership [2]. These results in turn suggest less need for dynamic user modelling but a larger need for dynamic focus management than has hitherto been assumed in the HCI/NLP communities.

2 SOME DESIGN ISSUES

To circumvent the risk of drawing general conclusions that in fact are only a reflection of the specific experimental setting used, we have striven to vary the type of background system, not only as regards the content or application domain, but also as regards the ’intelligence’ of the system and the types of possible actions that can be performed by the person using it. So far we have used nine different real or simulated background systems. Apart from the use of ’pure’ natural language, in one case the dialogues also contain tables displaying the contents of the INGRES-database, and in two cases a limited use of graphics is possible.

In our work on characterizing the genre of NLI-dialogues, our aim has been to simulate the interaction with a system in the case of an occasional or one-time user, i.e. a user who is unfamiliar with the system, but has some knowledge of the domain. We think that this is the most relevant user-category to study, as otherwise the user will be adapted to the system.

We have therefore tried to use background systems and tasks which follow these criteria. But it is not enough to have a reasonable background system and a good experimental environment to run a successful experiment. Great care should also be taken regarding the task given to the subjects. If we give them too simple a task to solve, we will not get much data to analyse, and if we give them too detailed instructions on which information they should seek from the system, there is a risk that what they will type is not their way of phrasing the questions but ours. Our approach has been to develop a so-called scenario, i.e. a task to solve whose solution requires the use of the system, but where there does not exist one single correct answer, and/or where there are more than one way to reach the goal. Fraser and Gilbert [9] in their simulations of speech-systems also propose the use of scenarios to achieve realistic interactions.

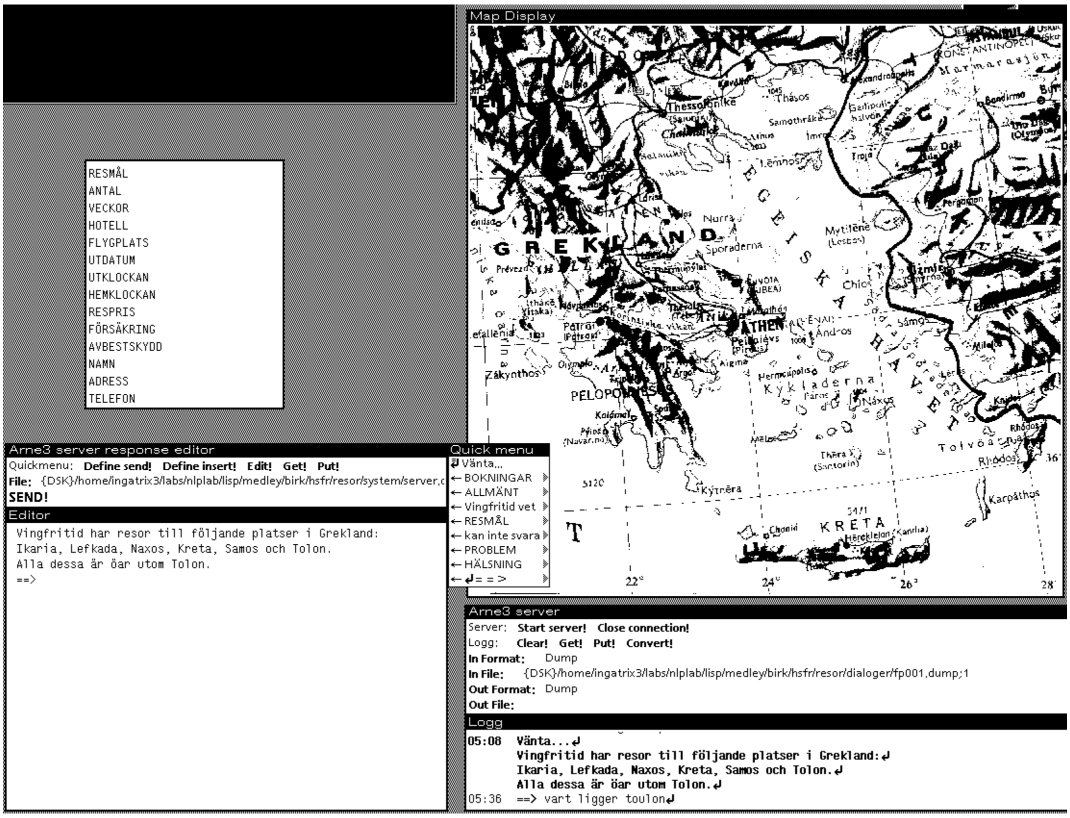
We have previously stressed some consequences of the fact that computers are different from people which has motivated the choice of Wizard of Oz simulations. But another consequence of this is that such simulations are very difficult to run. People are flexible, computers are rigid (or consistent); people are slow at typewriting, computer output is fast; computers never make small mistakes (e.g. occasional spelling errors), people make them all the time. The list could be made longer, but the conclusion is obvious. If we want our subjects to believe that they are communicating with a computer also after three exchanges, we cannot let the person simulating the computer just sit and slowly write the answers on the screen. Therefore, to make the output from the wizard resemble that of a computer as far as possible as regards timing and consistency, we have developed an environment for conducting the experiments. The background system can be a real system on the same or another computer, or it can be simulated too. The simulation environment will be the topic of the next section.

3 THE SIMULATION ENVIRONMENT *ARNE*

The simulation environment now exists in its third version, ARNE-3. Some of its main features are:

- response editor with canned texts and templates easily accessed through menus
- ability to access various background systems
- editor for creating queries to database systems
- interaction log with time stamps

The simulation environment is customized for each new application. An overview of the simulation environment is shown in figure 1, where the application is a Travel agency system holding information on holiday trips to the Greek archipelago. The environment in its base configuration consists of two parts, a log and a response editor, each accessed through its own window. The editor window can be seen in the lower left part of the screen, while the log window is found in the lower right part. Maps and other kinds of graphics can also be displayed.



File: {DSK}/home/ingatrix3/labs/nlplab/lisp/medley/birk/hsfr/resor/system/server.c

```
Vingfritid har resor till följande platser i Grekland:
Ikaria, Lefkada, Naxos, Kreta, Samos och Tolon.
Alla dessa är öar utom Tolon.
==>
```

In Format: Dump
In File: {DSK}/home/ingatrix3/labs/nlplab/lisp/medley/birk/hsfr/resor/dialoger/fp001.dump;1
Out Format: Dump
Out File:

```
05:08  Vänta...↲
       Vingfritid har resor till följande platser i Grekland:↲
       Ikaria, Lefkada, Naxos, Kreta, Samos och Tolon.↲
       Alla dessa är öar utom Tolon.↲
05:36  ==> vart ligger toulon↲
```

Figure 1 An overview of the simulation environment (The Wizards view.)

In one scenario for computerized travel agency system called Travel the subject can also order a holiday trip. The window in the upper left part of the screen is the template for the order. This is filled in by the wizard as the interaction proceeds. When the ordering window is completed, the subjects receive a confirmation in natural language of the ordered item. This is generated automatically by a Common Lisp function from the order template. This is in line with our general policy to automate as much as possible of the interaction.

The editor window is used to create responses to the subjects. When a response is ready it is sent to the subject and simultaneously logged in the log window. To speed up the response time the editor has a hierarchically organised set of canned texts which are easily reached through a set of menus, seen to the right in the editor window. Figure 2 shows the hierarchy of canned texts for the menu item Resmål (Eng. Resort). The wizard can stop at any point in the hierarchy and will thus provide more or less detailed information depending on how far the dialogue has developed. So, in the example of figure 2, if the wizard stops at Lefkada, on the second level in the hierarchy, the subject will be provided with general information on Lefkada, while if the wizard stops at Adani, general information about hotel Adani on Lefkada is provided. The total amount of canned text available in this fashion is 2300 rows, where a row is everything between a full line of text to a single word. This corresponds to approximately 40 A4-pages. The text from the menus is entered into the editor window, to allow the wizard to edit the information, if necessary, before sending it to the subject.

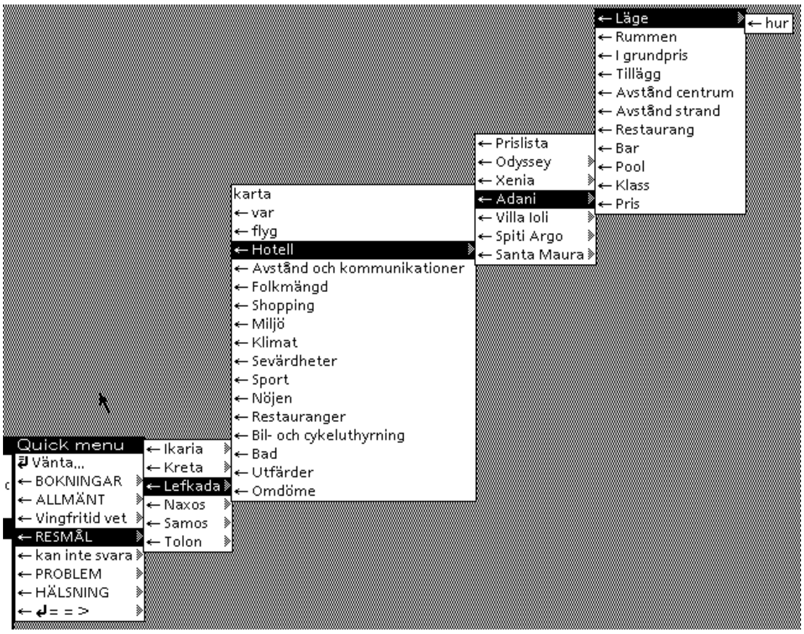
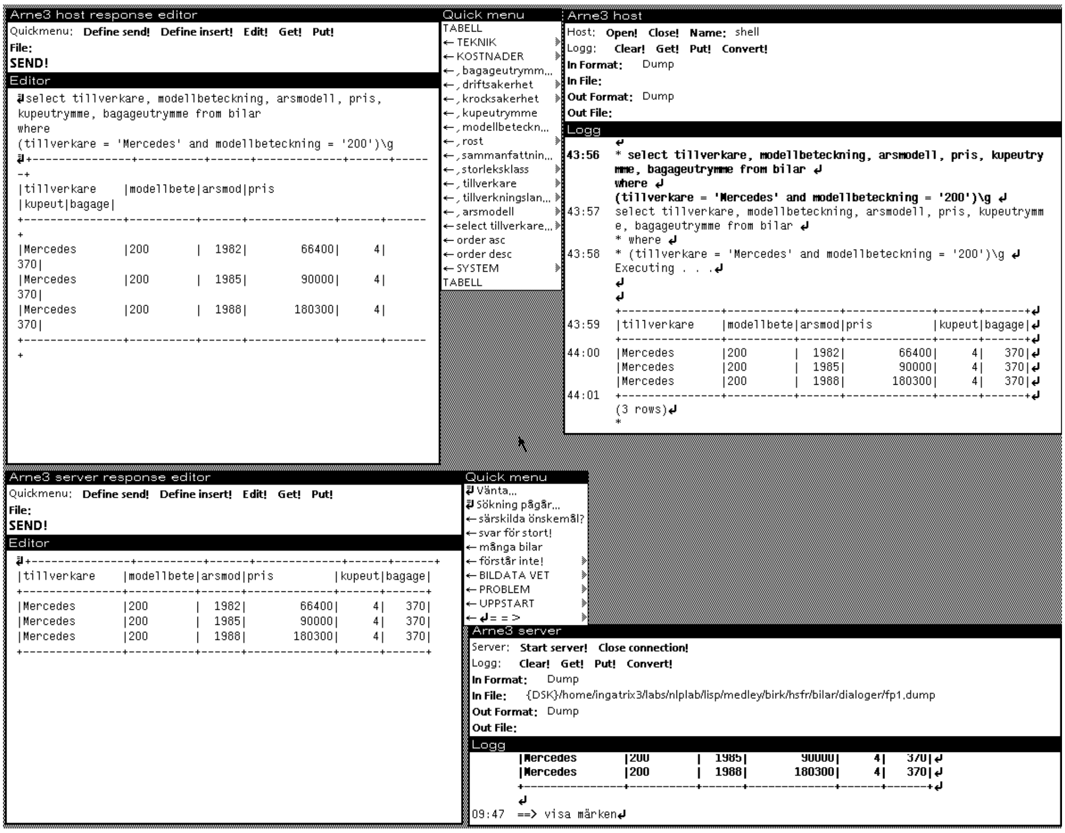


Figure 2 A menu hierarchy

Certain messages are so simple and also commonly used that they can be prompted directly to the subject without first passing the editor. These are also present in the quick menus. In the example there are two such quick responses, one is the prompt

==>, and the other is Vänta ... (Eng. ”Wait ...”). The latter ensures that the subject receives an acknowledgement as soon as the input is received by the system, while the wizard scans through the canned texts for an answer.

The simulation environment can also be connected to existing background systems. One example of this is the *Cars* simulations, where subjects could acquire information on properties of used cars from an Ingres database containing such information. The simulation environment in these simulations consisted of four different windows, as seen in figure 3.



```
↲select tillverkare, modellbeteckning, arsmodell, pris,
kupeutrymme, bagageutrymme from bilar
where
(tillverkare = 'Mercedes' and modellbeteckning = '200')\g
↲+---------------+----------+------+-------------+------+-----
-+
|tillverkare    |modellbete|arsmod|pris
|kupeut|bagage|
+---------------+----------+------+-------------+------+------
+
|Mercedes       |200       |  1982|        66400|     4|
370|
|Mercedes       |200       |  1985|        90000|     4|
370|
|Mercedes       |200       |  1988|       180300|     4|
370|
+---------------+----------+------+-------------+------+------
+
```

Name: shell
In Format: Dump
In File:
Out Format: Dump
Out File:

```
       ↲
43:56  * select tillverkare, modellbeteckning, arsmodell, pris, kupeutry
       mme, bagageutrymme from bilar ↲
       where ↲
       (tillverkare = 'Mercedes' and modellbeteckning = '200')\g ↲
43:57  select tillverkare, modellbeteckning, arsmodell, pris, kupeutrymm
       e, bagageutrymme from bilar ↲
       * where ↲
43:58  * (tillverkare = 'Mercedes' and modellbeteckning = '200')\g ↲
       Executing . . .↲
       ↲
       ↲
       +---------------+----------+------+-------------+------+------+↲
43:59  |tillverkare    |modellbete|arsmod|pris         |kupeut|bagage|↲
       +---------------+----------+------+-------------+------+------+↲
44:00  |Mercedes       |200       |  1982|        66400|     4|   370|↲
       |Mercedes       |200       |  1985|        90000|     4|   370|↲
       |Mercedes       |200       |  1988|       180300|     4|   370|↲
44:01  +---------------+----------+------+-------------+------+------+↲
       (3 rows)↲
       *
```

```
↲+---------------+----------+------+-------------+------+------+
 |tillverkare    |modellbete|arsmod|pris         |kupeut|bagage|
 +---------------+----------+------+-------------+------+------+
 |Mercedes       |200       |  1982|        66400|     4|   370|
 |Mercedes       |200       |  1985|        90000|     4|   370|
 |Mercedes       |200       |  1988|       180300|     4|   370|
 +---------------+----------+------+-------------+------+------+
```

In Format: Dump
In File: {DSK}/home/ingatrix3/labs/nlplab/lisp/medley/birk/hsfr/bilar/dialoger/fp1,dump
Out Format: Dump
Out File:

```
       |Mercedes       |200       |  1985|        90000|     4|   370|↲
       |Mercedes       |200       |  1988|       180300|     4|   370|↲
       +---------------+----------+------+-------------+------+------+↲
       ↲
09:47  ==> visa märken↲
```

Figure 3 The simulation environment used for the Cars application

Here there are two windows added for database access, one is the actual database interface and the other is a query command editor. As forming a SQL-query can take quite some time we needed to speed up that process. This is done in a similar way as the response generation namely by using hierarchically organised menus. The menus contain information that can be used to fill in a SQL-template as shown in figure 4. In figure 3, the SQL-query is displayed in the command editor in the upper left window, while the top menu for forming queries is immediately to the right of that window.

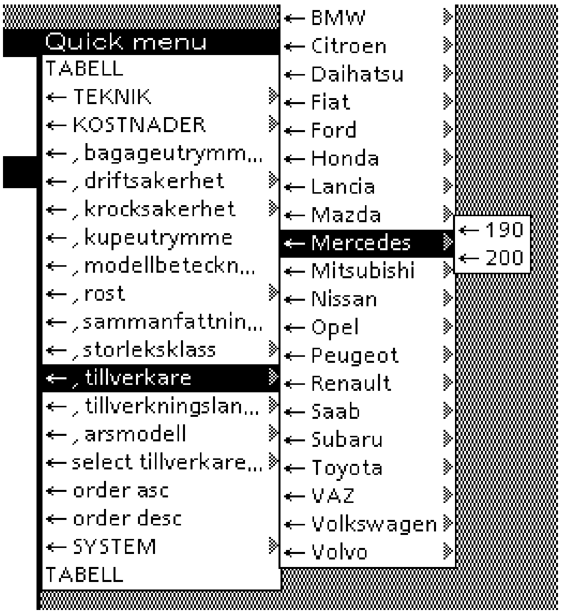


Figure 4 The editor used for database access

The editor used for this purpose is an instance of the same editor that was used for creating responses to the subject. Thus, the wizard need not learn a new system which again provides a more efficient interaction as the same commands and actions are used to carry out both tasks. The database access menus do not only contain SQL-query templates, but also entries for the objects and properties that are stored in the database. Thus the wizard can avoid misspelled words which would lead to a database access failure and a slow down of response time.

It is a time-consuming task to customize the simulation environment to a particular application. For some applications we have used some 20-40 pilot studies before being satisfied with the scenario and the performance of the simulation. But we believe that without such preparation, there is a large risk that the value of the results obtained is seriously diminished.

4 EXPERIMENTAL DATA

As mentioned previously, we have run Wizard of Oz-experiments both as part of a research on the general characteristics of NLI-dialogues, and as part of the development of a NLI for a specific application. The present size of our corpus is approximately 150 dialogues. It can be sub-divided into two corpora intensively analyzed and used in our empirical studies, called corpus 1and 2 below. The first contains 21 and the second 60 dialogues. For these studies we have also collected approximately 40-50 dialogues during the development of the simulation environment and the experimental situation. Furthermore, we have a set of dialogues collected while exploring background systems and experimental settings which we for various reasons have not pursued further. This set, which contains approximately 20 dialogues, and some of the experiences obtained during this work, is described in the section ’selecting systems’ below.

4.1 Corpus 1

Corpus 1 was collected using the first version of the simulation environment. This corpus contains dialogues with five real or simulated background systems.

The first system, PUB, was a library database then in use at our department, containing information on which books the department owned, and on which researchers room they were kept. Common bibliographic information was also obtainable. Four dialogues with this system are included in this corpus.

Another similar system, called C-line, was a simulated database containing information about the computer science curriculum at Linköping University. The scenario for the subjects was that they should imagine themselves working as study counsellors, their task being to answer a letter from a student with questions about the Master's program in computer science. Five dialogues were run in this condition.

The third system, called HiFi, is of a different kind. Here the user can order high quality HiFi equipment after having queried a simulated database containing information about the available equipment. The system can also answer some questions from about which pieces can suitably be combined, so in a sense it is not a database but an expert system. Five dialogues were run in this condition.

The fourth system in this corpus is the first version of the automated travel agency encountered previously in the description of the simulation environment. In this version there was no graphics facilities. This system is similar to the HiFi system in that the user can perform more than one task, but the travel system never gives any advice but only supplies information. In this corpus there are three dialogues run in this condition.

The fifth system in corpus 1 is a simulated wine selection advisory system. It is capable of suggesting suitable wines for different dishes, if necessary within a specified price range. It could also tell whether two wines could be combined in the same meal. The task of the subjects was to select wines for a dinner party where the menu and the amount of money available were determined. To be able to get out of a situation where the wizard did not know what to do, the simulation system also included a simulated system breakdown and restart. Four dialogues from this system are included in corpus 1.

A general overview of this corpus is presented in Jönsson & Dahlbäck [18] and Dahlbäck & Jönsson [7]. Dahlbäck [4] and Dahlbäck & Jönsson [8] report on dialogue structure while Dahlbäck [6] presents an analysis of pronouns distribution and function. Ahrenberg, Jönsson & Dahlbäck [1] gives an overview of the NLI-project for which the analysis was used. Dahlbäck [5] presents the most detailed analysis of both the dialogue structure and the pronoun patterns and also analyses the use of definite descriptions.

4.2 Corpus 2

The second corpus was collected using the refined Wizard of Oz-simulation environment presented here and a new set of scenarios.This corpus consists of totally 60 dialogues using two different background systems, the *Cars* database of used car models and a considerably revised and enlarged version of the travel system used in corpus 1. In this corpus half of the subjects could only obtain information from the system, whereas the other half of them also could order the trip as was the case in corpus 1. Dialogues where collected under two different conditions: one where the subjects knew that they were interacting with a person and one which was a real Wizard of Oz-simulation. We thus have 10 subjects in each cell as seen in figure 5. The analysis of this corpus is presently under way. Some results are used in [8]. Jönsson [15, 16, 17] presents a dialogue manager for NLIs based on an analysis of this corpus.

The same simulation environment is also used in a current project concerned with an empirical comparison of dialogue grammar based and plan-based dialogue models.

		Background system		
		Database		Database + ordering
Condition	Computer	Cars (10)	Travel (10)	Travel (10)
	Human	Cars (10)	Travel (10)	Travel (10)

Figure 5 Corpus 2

5 SELECTING SYSTEMS

We have found the simulation of database-dialogues fairly straightforward, as is the case with the simulation of systems where the user can perform more tasks, such as ordering equipment after having obtained information about the stock. But for some other kinds of systems we have encountered different kinds of problems, in some cases leading us to abandon the project of collecting the dialogues for a particular system, in some cases providing us with less reliable data.We include a description of our experience, since we believe there is something to be learned from it, either when designing these kinds of experiments, or when considering the applicability of natural language interfaces for similar kinds of systems.

The first example is of an EMYCIN based expert system, advising on tax issues in connection with the transfer of real estate. There were many reasons for our believing that this was a suitable system for our purposes. The domain is one with which we thought most people had some familiarity. Another reason was that rule-based expert systems such as this are a large and growing area and is considered one possible application domain for natural language interfaces.

The basic reason for not being able to use this promising application was that the system was only a prototype system that was never completed. Not only did it contain some bugs, but there were “holes” in its knowledge, i.e. some sub-areas for which no rules were implemented. It turned out to be impossible to create a reasonable scenario which guaranteed that the subjects kept themselves within the system’s competence.

The lesson we learned from this was that if we shall use a real background system it must be well tested and functioning properly. Furthermore, the dialogue of EMYCIN-based expert systems is controlled by the system to an extent that it is difficult to simulate a more open dialogue where the user can take the initiative too.

With the development of bitmapped screens and mouses, it becomes interesting to study multi-modal interfaces where users can use both written input and direct manipulation. And if we make it possible for the user to use both modes, we can learn something about when the different interface methods are to be preferred. We therefore tried to use a computer-based calendar system developed at our department for this purpose. In the system you can book meetings with groups and individuals, deciding on the time and location. You can also ask the system about these meetings, and about the times when people or groups of people are not booked, for instance when planning a meeting with them. You can do this by writing in a calendar displayed in a window on the screen, but also using a limited natural language interface.

There were two major problems when designing this experiment. The first problem was to expand the ARNE environment so that it could handle graphics too. In the Calendar system we actually send the graphics on the net between the different work stations, which for obvious reasons gave long response times. This gave rise to some problems discussed below. In the later travel agency project we have therefore stored all the graphical information on the user's machine, and only send a signal from the wizard to this station tell which picture to display.

The second problem was deciding how to analyse the obtained data, and this we did not solve. If we lack well developed theories for dialogues in natural language, the case is even worse for this kind of multi-modal dialogues. The only thing we have been able to do thus far is to simply observe the users running the system. But even this simple data collection has given us one somewhat surprising observation concerning the effects of very slow response times on the dialogue structure. The interesting fact is that in spite of these long response times, the language used be the subjects still is coherent, with a number of anaphoric expressions, something which goes somewhat contrary to expectations, since one could assume that it is necessary for the user to have the dialogue 'on the top of his consciousness' to be able to use such linguistic devices. It is of course not possible here to give an explanation of this phenomenon, which in our opinion requires further investigation. But it is possible that the fact that both the dialogue and the calendar is displayed on the screen affects the dialogue structure.

Another system tried but not used was an advisory program for income tax return and tax planning that runs on IBM PCs. The reason for thinking that this was a suitable system for our experiments is of course the same as the one first one described above. One reason for not using it was that very little dialogue was necessary to use the program, apart from filling in the menus that correspond to various part of the tax forms. So it seems as if a natural language interface is not the preferred mode for such a system, but at most something which can supplement it. Another difficulty was with the scenario, as people are not willing to present their own income tax planning in an experiment and it is quite a complex task to learn a fictitious tax profile.

In another experiment an advisory system was simulated. But there are some problems with this too, the most important being that it is difficult for the Wizard to maintain a consistent performance and give the same answers to similar questions from different users, and even from the same user. To some extent these problems can be overcome, but it seems to require longer development phases than for other kinds of systems.

Advisory system thus seem to give us two kinds of problems if we want to use them in Wizard of Oz studies. On the one hand, the simulation of the system is difficult to do, and if one wants to use a real system developed on existing shells, at least in some cases the dialogue is system driven to an extent that there seem to be little that can be gained from such a study.

To summarize, we can identify three parameters that must be given careful consideration: the background system, the task given to subjects, and the wizard's guidelines and tools.

- The background system should be simulated or fully implemented. A shaky prototype will only reveal that system's limitations and will not provide useful data. Furthermore, the system should allow for a minimum of mixed-initiative dialogue. A system-directed background system will give a dialogue which is not varied enough. However, if the purpose is to collect data from the use of a particular application, or for the development of an interface for a particular system, then that application will determine the interaction.

- The task given to subjects must be reasonably open, i.e. have the form of a scenario. Retrieving information from a database system and putting it together for a specific purpose can be fruitful. But, if the domain is so complex that it requires extensive domain knowledge, or the task is of a more private nature, then it is likely that the subjects try to finish their task as quickly as possible and again not provide enough dialogue variation. The specification of the task must allow for varied outcomes. Many different outcomes must be considered "correct" and there should be many ways to explore the background system to achieve a reasonable result.
- Finally, we have the simulation environment and guidelines for the wizard. The simulation experiment must be studied in detail, from pilot experiments, before the real simulations are carried out. This information is used to provide knowledge to the wizard on how to act in various situations that may be encountered. Pilot experiments are necessary for every new application to reveal its distinctive character. Of course, this can not replace a careful examination of the application. The wizard needs to have full knowledge of the information that can be provided, how it is organised, and how it can be acquired. Moreover, he needs a variety of pre-stored responses covering typical situations. Otherwise, besides slowing down the simulation, the ensuing variation will provide results that are less generalizable.

6 DOES THE METHOD WORK?

We have conducted post-experimental interviews with all our subjects. The most important objective was of course to ascertain that they had not realized that the system had been simulated, and also to explain what we had done and why we had deceived them. We also explained that the collected dialogue should be destroyed if they so wished. In our first series of studies none of the subjects thought that it had been a simulation. Some said that they had been somewhat surprised in a positive way by the capabilities of the system used, but none to the extent that they thought it improbable that such a system could be built. In our most recent project comprising the 60 dialogues of Corpus 2, two subjects voiced suspicions, but this does not alter the conclusion that Wizard-of-Oz studies are possible to realize for a large number of applications.

We have also asked subjects which aspects of language they thought would be most difficult to make a computer handle. The most frequent answers were spelling correction and large vocabularies. No subject mentioned connected dialogue, anaphora, pronouns or the like. (This is corroborated by the collected dialogues, which contain few spelling errors, but a large number of utterances the interpretation of which requires knowledge of the preceding dialogue.) The answers given in these interviews give us some confidence in having succeeded deceiving the subjects, and that therefore the dialogues reflect the language that would be attempted when communicating with a computer. In spite of this, in some cases the structure of the dialogues gives the impression that the subjects changed task during the dialogue; from solving the task given in the instruction to trying to experiment with the capabilities of the system. This does not necessarily make the dialogues more unrealistic - experimenting with a new and interesting program is something that many computer interested people do. But these parts of the dialogues should perhaps be given a separate analysis.

7 FOR AND AGAINST THE CHOSEN METHOD

In a review of the Wizard of Oz method Fraser and Gilbert [9] argued, on ethical grounds, against deceiving subjects about the nature of their conversation partner. We do not want to deny that there are ethical problems here, but we think that they can be overcome, and that there are good reasons for trying to do so. As pointed out above it has been shown that there are differences in the language used when subjects think they communicate with a computer and when they think they communicate with a person. And, what is more important, the differences observed concern aspects of language over which subjects seem to have little conscious control, e.g. type and frequency of anaphoric expressions used. So at least if your interests concern syntax and discourse, we consider it important to make the subjects believe that they are communicating with a computer, simply because we do not think that subjects can role-play here and give you the data you need. And if, on the other hand, you find that subjects find it difficult to use the existing NLI, as for instance in [14], this amounts hardly to anything more than a demonstration of the limitations of existing technology.

So much for the need, but how about the ethics? We would claim that if one uses the practice developed within experimental social psychology of a debriefing session afterwards, explaining what you have done and why you found it necessary to do so, and furthermore that you tell the subjects that the data collected will be destroyed immediately if they so wish, you will encounter little problem. In our experiments we have done so, and we have so far only had a number of good laughs and interesting discussions with our subjects on their expectations of what a computer can and cannot do, but no one has criticized us for our ’lies’. Perhaps one reason for this is that none of the subjects felt that they had been put in an embarrassing situation. It is not exactly the same as Candid Camera.

Another possible critique is that one should study existing systems instead of simulated ones. But in this case we agree with Tennant's [25] conclusion that people can often adapt to the limitations of an existing system, and such an experiment does not therefore tell you what they ideally would need. It could also be argued that the human ability to adapt to the communicative capacity of the dialogue partner means that what we find is only the subjects adaptive responses to the wizard's conception of what an NLI should be able to do. But it is exactly for this reason that the wizards in our experiments have not been instructed to mimic any specific capacity limitations. At the present stage of development of NLI technology, we cannot say with any high degree of certainty what we will and will not be able to do in the future. Furthermore, it is extremely difficult to be consistent in role-playing an agent with limited linguistic or communicative ability, so, to make such an experiment you would need some way of making the restrictions automatically, for instance by filtering the input through a specific parser, and only understand those utterances that can be analysed by this parser. Furthermore, the fact that we have used different wizards for the different background systems guarantees at least that the language we find our subjects using is not the reflection of the idiosyncrasies of one single person's behaviour in such a situation.

The possible critique against the artificiality of the experimental setting can be levelled against another aspect of the method used, namely that the subjects are role-playing (cf. Ogden [21]). They are not real users, and their motivation for searching information or ordering equipment is really not theirs. This is an argument that should be taken seriously. It is, however, our belief that the fact that the subjects are role-playing affects different aspects of their behaviour differently. If the focus of interest is for instance the goals and plans of the users, and the way that is manifested in the dialogue, the use of role-playing subjects should be made with caution.

But if the focus is on aspects not under voluntary conscious control (cognitively impenetrable, to use Pylyshyn's [22], term), the prospect is better for obtaining ecologically valid data. To take one specific example; if a user is just pretending to buy a holiday trip to Greece, she might not probe the alternatives to the extent that she would if she were in fact to buy it, simply because the goal of finishing the task within a limited time takes precedence. But it does not seem likely that the latter fact will affect the use of pronouns in a specific utterance, or the knowledge about charter holidays and Greek geography that is implicitly used in interpreting and formulating specific utterances.

8 CONCLUDING REMARKS

The present paper makes two points, one theoretical and one methodological. On the theoretical side we argue that it is natural for any human engaging in a dialogue to adapt to the perceived characteristics of the dialogue partner. Since computers are different from people, a necessary corollary from this is that the development of interfaces for *natural* dialogues with a computer cannot take human dialogues as its sole starting point, but must be based on a knowledge of the unique characteristics of these kinds of dialogues. Our own work has been concerned with natural-language interfaces, but the argument is of relevance for all kinds of intelligent dialogue systems.

The methodological point is simply that to acquire the relevant knowledge, we need high quality empirical data. But if the point is simple, gathering such data is not quite that simple. One way of doing so is by simulating intelligent interfaces (and sometimes also systems) using so-called Wizard of Oz-studies, i.e. having a person simulate the interface (and system). But it is important to realize, that to acquire the required high-quality data a great deal of care and consideration need to be used in the design of such experiments. We have described our own simulation environment ARNE and some of our practical experiences, both positive and negative, to illustrate some of the points that we consider important if such a research program is to contribute to the development of theoretically and empirically sound user friendly intelligent interfaces.

9 REFERENCES

1. Ahrenberg, Lars, Arne Jönsson & Nils Dahlbäck ’Discourse Representation and Discourse Management for Natural Language Interfaces’ *Proceedings of the Second Nordic Conference on Text Comprehension in Man and Machine,* Täby, Stockholm (1990).

2. Clark, Herbert H. & Catherine Marshall ’Definite Reference and Mutual Knowledge’ . In Joshi, Aravind, Webber, Bonnie, and Sag, Ivan (eds.) *Elements of Discourse Understanding.* Cambridge, Mass.: Cambridge University Press (1981).

3. Cohen, Philip R. ’The pragmatics of referring and modality of communication’ *Computational Linguistics* **10** (1984) pp 97-146.

4. Dahlbäck, Nils ’Empirical Analysis of a Discourse Model for Natural Language Interfaces’ *Proceedings of the Thirteenth Annual Meeting of The Cognitive Science Society,* Chicago, Illinois (1991) pp 1-6.

5. Dahlbäck, Nils, *Representations of Discourse, Cognitive and Computational Aspects*, PhD-thesis, Linköping University (1991).

6. Dahlbäck, Nils ’Pronoun usage in NLI-dialogues: A Wizard of Oz study. To appear’ *Proceedings of the Third Nordic Conference on Text Comprehension in Man and Machine,* Linköping April, (1992).

7. Dahlbäck, Nils & Arne Jönsson ’Empirical Studies of Discourse Representations for Natural Language Interfaces’ *Proceedings of the Fourth Conference of the European Chapter of the ACL,* Manchester (1989) pp 291-298.

8. Dahlbäck, Nils & Arne Jönsson ’An empirically based computationally tractable dialogue model’ *Proceedings of the Fourteenth Annual Meeting of The Cognitive Science Society,* Bloomington, Indiana (1992) pp 785-790.

9. Fraser, Norman & Nigel S. Gilbert ’Simulating speech systems’ *Computer Speech and Language* **5** (1991) pp 81-99.

10. Gal, Annie *Cooperative responses in Deductive Databases.* PhD Thesis, Department of Computer Science, University of Maryland, College Park (1988).

11. Grosz, Barbara *The Representation and use of Focus in Dialogue Understanding*. Unpublished Ph.D thesis, University of California, Berkely (1977).

12. Grosz, Barbara J., Martha Pollack & Candace L. Sidner ’Discourse’ In: Posner M. I. (Ed.) *Foundations of Cognitive Science,* Cambridge, MA: The MIT Press (1989) pp 437-468.

13. Guindon, Raymonde ’A multidisciplinary perspective on dialogue structure in user-advisor dialogues’. In Guindon, Raymonde (ed.) *Cognitive Science and its Applications for Human-Computer Interaction*. Hillsdale, N.J.: Erlbaum (1988).

14. Jarke, M., Krause, J., Vassiliou, Y., Stohr, E., Turner, J. & White, N. ’Evaluation and assessment of domain-independent natural language query systems’ *IEEE quarterly bulletin on Database Engineering,* Vol. 8, No. 3, (1985) pp 34-44.

15. Jönsson, Arne ’A Dialogue Manager Using Initiative-Response Units and Dis-

tributed Control’ P*roceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics,* Berlin (1991) pp 233-238.

16. Jönsson, Arne ’A Method for Development of Dialogue Managers for Natural Language Interfaces’ *Proceedings of AAAI-93*, Washington DC (1993).

17. Jönsson, Arne *Dialogue Management for Natural Language Interfaces — An Empirical Approach* PhD-thesis, Linköping University (1993).

18. Jönsson, Arne & Nils Dahlbäck ’Talking to a Computer is not Like Talking to Your Best Friend’ *Proceedings of The first Scandinivian Conference on Artificial Intelligence*, Tromsø, Norway (1988) pp 297-307.

19. Kennedy, A., Wilkes, A., Elder, L. & Murray, W. ’Dialogue with machines’ *Cognition*, **30** (1988) pp 73-105.

20. Lakoff, R.T. ’The Logic of Politeness; or minding your p’s and q’s’ *Papers from the Ninth Regional Meeting, Chicago Linguistic Society*, (1973) pp 292-305.

21. Ogden, William C. ’ Using Natural Language Interfaces’ *Handbook of Human-Computer Interaction,* M. Helander (Ed.), Elsevier Science Publishers B. V. (North Holland) (1988).

22. Pylyshyn, Zenon *Computation and Cognition*, Cambridge MA: The MIT Press (1984).

23. Reichman, Rachel *Getting Computers to Talk Like You and Me*, MIT Press, Cambridge, MA (1985).

24. Shatz, M. & Gelman, R. ’ The development of communication skills: Modifications in the speech of young children as a function of listener’ *Monographs of*

the Society for research in child development. **38**, No 152.

25. Tennant, Harry *Evaluation of Natural Language Processors* Ph.D. Thesis. University of Illinois Urbana-Champaign (1981).