

# Cooperation and Collaboration in Natural Command Language Dialogues

J. Gabriel Amores and José F. Quesada

University of Seville

jgabriel@us.es jquesada@cica.es

## Abstract

This paper discusses cooperative and collaborative behaviour in Natural Command Language Dialogues (NCLDs). We first introduce Natural Command Language Dialogues and then briefly compare them with other types of dialogue. Cooperation and collaboration in NCLDs is then analyzed. Finally, a typology of conflicts in NCLDs is proposed, for which a solution has been implemented in two spoken dialogue prototypes. Sample dialogues are taken from research carried out under the Siridus and D’Homme European projects.

## 1 Natural Command Language Dialogues

A Natural Command Language (NCL) is a command language expressed through the medium of natural language. We take NCLs as the set of input and output natural language expressions which are acceptable in a given application domain. This domain is semantically defined by the functions (commands) known by the user and the system, and the natural language vocabulary which may be used to express those commands. In addition, NCLs should contain metalinguistic patterns and expressions typical of human-like interaction. Natural Command Language Dialogues are artificially constructed models of action-oriented dialogues (including knowledge representation and reasoning) able to guide the interaction between the different parts involved in a dialogue based on a NCL. A NCLD should allow the following kinds of phenomena:

- **Multiple Task NCL:** In contrast to Task-Oriented Dialogue models, NCLDs must be able to manage different tasks. Thus, one of the main functions of NCLD

systems will be task detection, as will be explained below.

- **Context Dependency:** Only at the dialogue level is it possible to understand anaphora, ellipsis and other context dependent constructions. From the dialogue system design perspective, the treatment of these discourse phenomena will imply the representation and storage of the whole dialogue history. For an illustration, see (Quesada and Amores, 2002) in this volume.
- **Man-Machine Interaction:** An adequate level of naturalness and flexibility in the flow of interactions (restricted to the linguistic limitations of the underlying NCL), relevance and adequacy of the outputs, and consistency (order of arguments in commands) should be accomplished.
- **Interface with External Functional Components:** NCLs are aimed at the specification of commands belonging to a command language. The user’s goal is to execute the command(s). Therefore, the dialogue level must not only understand the naturally expressed commands, but also execute them. In fact, this implies the definition and use of another Command Language between the NCLD System and the external functional components in charge of the execution of the commands.

This conceptualization of NCLs has been applied in two spoken dialogue systems under the Siridus and D’Homme European Projects.

### 1.1 Siridus and D’Homme

In Siridus (Siridus, 1999), we are building an Automatic Telephone Operator System in Span-

ish jointly with Telefónica I+D. The user should be able to naturally issue commands in order to perform the following tasks: call an extension by name or office; redial a number; transfer incoming calls to another extension or office; cancel transference; set up a conference call; look up an office in the company's directory; and look up an e-mail address in the company's directory.

The benefits of a natural command language system in this domain is evident. For example, functions such as transferring incoming calls, placing conference calls, transferring ongoing calls etc, are typically performed by dialling several codes in a pre-defined sequence. However, since these codes are difficult to remember, users tend not to use them. Automatic dialogue systems seem to be very appropriate for these circumstances since they could allow users to perform some or all of these functions using their voice and natural language. Similar systems have been built by Bell Labs, AT&T and Wildfire Communications. For a comparison with our system, and a discussion about the benefits of automatic telephone systems, see (Torre and others, 2001)

In the D'Homme project (DHomme, 2001), the implemented system was able to perform the following functions in a home environment: switch on/off a device or set of devices; dim or bright a device or set of devices; and consult the state or location of a device or set of devices. A spoken interface offers several benefits to users, especially the elderly and disabled: we can query multiple devices with a single command, we can control devices, we can also use language to program devices to interact with each other, etc. Remote control via the telephone is a natural extension to home deployed systems. We can phone up the house to see if we forgot to turn off some device, or we could ask our house to turn on the heating on our way home after some days of absence.

In addition to these domains, there are some other dialogue systems implementing some version of a command language, such as the WITAS system (Doherty and others, 2000).

## 2 Comparing NCLDs with other types of dialogue

Before we analyze the cooperative and collaborative behaviour, it is worth pointing out some characteristics of NCLDs.

As a consequence of their own nature, NCLDs involve just two participants: the user and the machine. One of the first decisions concerns whether we should model the participants' internal beliefs, or more external aspects of the dialogue. Since the goal of this type of dialogues is that the user have control over the execution of one or more commands by the machine, most dialogues exhibit a marked functional or operational tendency, i.e. are action-oriented dialogues geared toward the execution of some non-communicative action. So, it seems reasonable to focus our model on the external aspects of dialogue. That is, it should be based more on what was said than what was in the minds of the participants when their interactions were produced.

NCLDs are different from other types of dialogue such as Information Seeking and Negotiative Dialogues. In Information Seeking dialogues, one participant requests information from the other. In Negotiative Dialogues, the goal of both participants is to come to an agreement about some conflict of interests.

Another aspect which differentiates NCLDs from Information Seeking Dialogues is the dynamic nature of the knowledge bases involved. In an information seeking dialogue, in which the system as a whole is viewed by the user as a repository of knowledge, knowledge bases have a clear static character from the point of view of the user. That is, the data in these resources may be updated, but not by the user during its interaction with the system. On the contrary, one of the main features of NCLDs is the presence of a command execution which is capable of dynamically modifying the contents of external resources to the dialogue, such as the knowledge bases associated to the domain.

### 2.1 Functional Embedding

An important aspect of NCLDs is that they usually exhibit *functional embedding* (Walton and Krabbe, 1995; Reed and Long, 1997). Functional embeddings occur when the goal of a sub-dialogue shifts to another dialogue type.

As pointed out above, task identification sub-dialogues are possible in NCLDs. As opposed to other dialogue types in which the overall task of the dialogue is predefined (seeking information about flights in a travel agency, etc.), the system in a NCLD does not know *a priori* which function the user desires to perform. The first task, therefore, involves identifying the speaker's intentions. Once the task has been identified, the corresponding plan may be unfolded. Task identification may be considered a sort of *deliberation* subdialogue in the sense of Walton and Krabbe (1995). In deliberation dialogues, the participants jointly aim to reach a decision or form a plan of action. Deliberation is goal-directed, in contrast to the more abstract reasoning characteristic of negotiation.

Clarification subdialogues are also common in NCLDs, when the name of the destination of a call or some other parameter required for the successful completion of a command was misunderstood or missing. If some portion of the desired action was understood, it will be used as indirect feedback to the user:

- **U(1)**: Please, transfer my calls to Meeting room E-30  
(function was understood, but not the destination)
- **S(1)**: Could you repeat where I should transfer your calls?

Clarifications may be viewed as negotiation stages within the overall dialogue. Another instance of negotiation in NCLDs occurs when the system proposes alternatives in cases of conflict. This will be discussed in more detail below.

Information seeking subdialogues are common in NCLDs when the user wishes to know the state of specific devices, or consult the details (e-mail, office number) of another user, as pointed out above.

Figure 1 below shows a possible cascade of functional embedding in NCLDs.

As has become apparent in this section, NCLDs exhibit much more complexity than one could originally envisage given the apparent simplicity of the task at hand.

### 3 Cooperative behaviour in NCLDs

Let us now turn to the question of cooperative behaviour.

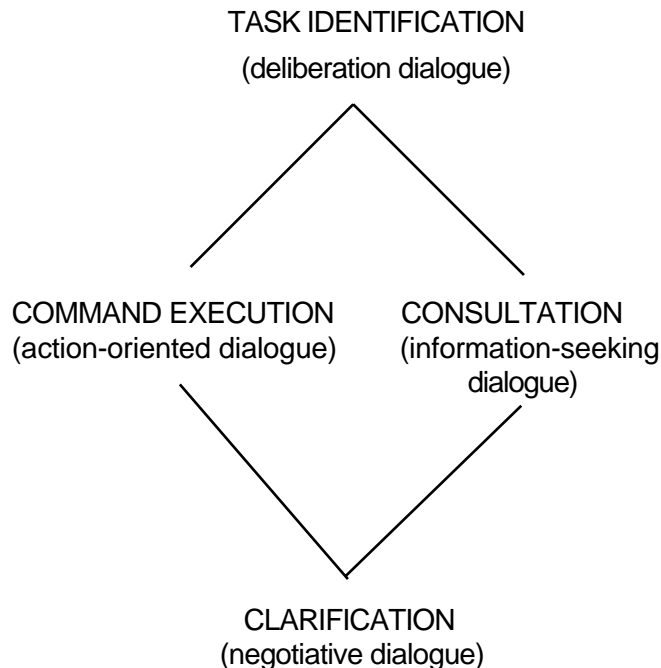


Figure 1: Cascade of functional embeddings in NCLDs

The linguistic definition of cooperation was first proposed by Grice (1975). Reed and Long (1997) propose a notion of cooperation which is similar to that of Grice, but acts at a higher level. Gricean cooperation is speaker-centered, whereas in their view, cooperation is more objectively discourse-centered. In their opinion all dialogue is inherently cooperative since any instance of true dialogue involves the participants accepting a common goal and working towards that goal within a given set of rules. This concept is also similar to the *Conversational Contract* of Fraser (1990).

Another relevant aspect regarding cooperation is that proposed by Allwood (1976). According to Allwood, two or more parties interact cooperatively to the extent that they:

1. take each other into cognitive consideration,
2. have a joint purpose,
3. take each other into ethical consideration,
4. trust each other to act in accordance with 1-3.

Of these, the most relevant requirement as regards NCLDs is 4 since, in fact, one of the

agents (the machine) in NCLDs is just *pretending* to act according to the principles. This idea reinforces our previous analysis of NCLDs as an inherently cooperative activity since, as long as the agents trust each other to act according to a set of principles, communication will be cooperative.

However, there is a simple fact about NCLDs which may change our perspective. Namely, the system in NCLD applications is usually *dispensable*. Its role is to make our lives easier. If it fails to achieve that goal, the user may just ignore it and proceed to perform the desired function in the ordinary way (by pressing a sequence of digits in the telephone pad, using light switches, etc.).

In this kind of scenario the system should be as unobtrusive as possible, trying to avoid a situation of frustration on the other participant which would indeed lead to interrupting any communication between them. So, what is usually modeled in NCLD systems is a *collaborative* behaviour in which the participants are in some sense working together to reach the desired outcome. The presence or absence of conflict will determine the level of collaboration in a dialogue. The next section describes the types of conflict which may arise in NCLDs and the expected level of collaboration which should be incorporated.

#### 4 Sources of conflict in NCLDs

Conflict arises from different sources in NCLDs.

First, the user may just not know exactly what functionality is available, or the parameters which they require. Providing help is one way of solving this conflict. Two *Dialogue Moves* take care of providing help in our systems. The first one provides general help on *what* commands are available. The second instructs the user about *how* to carry out a specific command, i.e. what parameters are required.

Second, given the dynamic nature of the domains in which we are working, one of the agents (the user) may just ignore the exact state of the world at that precise moment. Requesting information is then a collaborative activity which the system should model. Several dialogue moves have also been incorporated to take care of this kind of interaction.

A third type of conflict arises when one of

the agents wrongly believes that a specific goal is possible, but the state of the world does not permit its accomplishment. Solving this conflict involves proposing alternatives, as the following dialogue illustrates:

- **U(1)**: Call Mr Fernández
- **S(1)**: This extension does not respond. Do you want to leave a message or shall I try his mobile?

A more cooperative scenario is proposed in the following example:

- **U(1)**: Call Mr Fernández at extension 311 (Extension and name do not match in the database)
- **S(1a)**: Mr Fernández does not have this extension; shall I put you through his current extension?  
(or)
- **S(1a)**: Extension 311 belongs to Mr González; do you want him instead?

In some other contexts, the alternatives proposed by the system may be based on what the user *usually* does, that is, her most frequent destinations or actions in the past. Behaving cooperatively in this context occurs, for example, when the user underspecifies a destination of a call by a last name, and that last name turns to be ambiguous in the telephone directory. In such an event, the system may consult the most common destinations (chosen in the past by the user) and propose one who matches the last name provided.

- **U(1)**: Call Fernández  
(after spotting a frequent Pedro Fernández in the user's profile)
- **S(1)**: Do you mean Pedro Fernández?
- **U(2)**: Yes, please

This form of user adaptation is related to advanced modes of cooperation to be discussed in section 5 below.

*Reaccommodation* may be seen as a kind of cooperative behaviour as well since it involves solving an elliptical utterance. Question/Issue Reaccommodation is defined by Larsson (2002,

162) as “adjustments of common ground required to understand an utterance addressing an issue which has been resolved and does not influence any other resolved issue, influences another resolved issue or concerns grounding of a previous utterance.” Two instances of reaccommodation have been implemented in our system. The first example exemplifies a case in which there is a previous history in the dialogue, and the user issues an incomplete command.

- **U(1):** Switch on the light in the kitchen (the light in the kitchen goes on)
- **U(2):** Now the one in the living room

A more complex type of reaccommodation occurs when, after a function has been executed, the user retracts and wishes to undo the previous action:

- **U(1):** Switch on all indoor lights (all indoor lights go on)
- **U(2):** No, the one in the garage (the system must undo the previous action first)

The last (and more productive) source of conflict arises from misunderstandings stemming from misrecognition. In our systems, some degree of cooperative behaviour has been implemented, depending on the consequences of adopting an aggressive attitude or whether explicit confirmation was seen as necessary.

- **U(1):** I want to call 1 2 3 4 collect
- **S(1):** Do you really want to call 1 2 3 5 collect?
- **U(2):** no, 1 2 3 4.
- **S(2):** OK, shall I call 1 2 3 4 collect?
- **U(3):** yes, please.

The table below summarizes the types of conflict which may arise in NCLDs, the related Dialogue Move and the associated cooperative/collaborative action proposed for each case.

## 5 Advanced Cooperation

This section outlines some modes of advanced cooperation in NCLDs. Some of them may be considered cases of advanced flexibility or even robustness, which, in some sense, may be viewed as other aspects of cooperative behaviour. This functionality has not been fully implemented in our systems yet, partly because it relies on the technical limitations of the specific hardware being used.

- **User profiling:** In the automatic telephone domain, the command *Call my wife at her mobile* is only possible if a personal directory for each person making use of the system is made available, in addition to the general one in which all the personnel is stored.

In addition, the system might allow the user to set *modes* of behaviour. For example, in the home domain, it could be possible to issue the command *set night mode*, and automatically the system would perform a series of tasks such as setting some external lights on, all indoor lights off, switching the alarm on, etc.

Similarly, in the telephone domain, the user might want to set a *non-disturb* mode, and the system should transfer all incoming calls to the answering machine or to the secretary.

- **Default reasoning** may be considered a form of cooperative behaviour, at least in the telephone scenario. In our system a default action has been specified, whereby if no other action has been recorded in the dialogue history, a *PhoneCall* action will be executed. This is useful in the (frequent) cases in which the user just picks up the telephone and utters,

– **U(1):** Carlos García, please

- **Proactive Behaviour** is the most extreme case of cooperativeness. In the home environment, proactive behaviour has already been proposed for cases of fire, gas or water alarm, but they rely more on the technical capabilities of the specific setting than on dialogue capabilities.

In the telephone scenario, however, a proactive cooperative behaviour stemming

Type of Conflict	Related Dialogue Move	Proposed Action
user ignores overall functionality	askHelp	provide general help
user ignores how to carry out a specific command	askHelp	provide specific help
user ignores current state of the world	requestQuantity request Exist requestLocation requestDevice requestState	provide requested information
desired command cannot be accomplished	informExecution	Propose alternatives
elliptical command	errorRecovery	Try reaccommodation
Command and/or parameter misunderstanding	askRepeat askConfirmation	Clarification subdialogue

Table 1: Types of conflict in NCLDs and cooperative action proposed

from the dialogue occurs when the destination is busy, and the system proposes to trigger a call-back when she is done:

- **S(1)**: Calling Juan Fernández ...
- **S(2)**: Mr. Fernández is busy. Shall I have him call you back when he is done?
- **U(1)**: Yes, please.

## 6 DISC Guidelines

Finally, let us briefly examine to what extent our systems comply with the DISC guidelines for cooperative dialogue (Consortium, 1999). In particular, we will focus on the *specific guidelines* proposed.

- **SG1** *Summarising feedback*: is achieved through direct or indirect confirmation *Shall I transfer your calls to 0 1 2 3?*
- **SG2** *Provide immediate feedback*: in some tasks, such as conference calls, it is better to confirm one destination party at a time, given the ambiguities that would result from the combinations of first names, last names, second last names, etc.
- **SG3** *Ensure uniformity*: does not apply in the home environment since no linguistic feedback is generated for most command.

In the telephone scenario this is achieved through ‘canned expressions’ in the synthesizer.

- **SG4** *State your capabilities*: is achieved through general help. A non recognized command will only turn into *I cannot do that* or *that’s beyond my current capabilities* if the expression was understood as a command, and this command is not in the set of those supported by the current system. Otherwise, misinterpretation will arise.
- **SG5** *State how to interact*: is achieved through specific help.
- **SG6** *Be aware of user inferences*: is achieved through profiling behaviour.
- **SG7** *Adapt to target group, novice/expert users*: some version of this behaviour may be achieved in the telephone scenario since the recognizer may take input even before the system finished to provide instructions.
- **SG8** *Cover the domain*: some degree of flexibility is achieved through advanced cooperative functions such as user profiling.
- **SG9** *Enable system repair*: yes.
- **SG10** *Enable inconsistency clarifications*: yes, for example in the home domain, as in *The light in the bathroom is already on.*

- **SG11** *Enable ambiguity clarification*: yes, through anaphora resolution, default reasoning and user profiling.

## 7 Conclusion

This paper has analyzed the complexity of NCLDs, and the different perspectives from which such dialogues may be cooperative and collaborative. In particular, several types of conflict have been identified, for which an adequate solution has been implemented in two spoken dialogue prototype systems under the D'Homme and Siridus projects. As we have shown, the systems comply with most of the DISC guidelines for cooperative dialogue.

## References

- Jens Allwood. 1976. Linguistic communication as action and cooperation. Gothenburg Monographs in Linguistics 2, Department of Linguistics, University of Göteborg.
- The DISC Consortium. 1999. Disc dialogue engineering model. Technical report, DISC, <http://www.disc2.dk/slids/>.
- DHomme. 2001. <http://www.ling.gu.se/projekt/dhomme/>.
- Patrick Doherty et al. 2000. The witas unmanned aerial vehicle project. In W. Horn, editor, *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000)*, pages 747–755.
- Bruce Fraser. 1990. Perspectives on politeness. *Journal of Pragmatics*, 14(2):219–236.
- Herbert Paul Grice. 1975. Logic and conversation. In P. Cole and J. L. Morgan, editors, *Speech Acts*, volume 3 of *Syntax and Semantics*, pages 41–58. Seminar Press, New York.
- Staffan Larsson. 2002. *Issue-based Dialogue Management*. Ph.D. thesis, Department of Linguistics, Göteborg University, Sweden.
- José F. Quesada and J. Gabriel Amores. 2002. Knowledge-based reference resolution for dialogue management in a home domain environment. In *Proceedings of Edilog*, Edinburgh.
- Chris A. Reed and Derek P. Long. 1997. Collaboration, cooperation and dialogue classification. In K. Jokinen, editor, *Working Notes of the IJCAI97 Workshop on Collaboration, Cooperation and Conflict in Dialogue Systems, IJCAI 97*, pages 73–78, Nagoya, Japan.
- Siridus. 1999. <http://www.ling.gu.se/projekt/siridus/>.
- Doroteo Torre et al. 2001. User requirements on a natural command language dialogue system. Deliverable 3.1, Siridus Project.
- Douglas N. Walton and Erik C. W. Krabbe. 1995. *Commitment in Dialogue*. State University of New York, New York.