Natural Language Engineering **3** (2/3): 103–122. Printed in the United Kingdom

© 1997 Cambridge University Press

A model for habitable and efficient dialogue management for natural language interaction

ARNE JÖNSSON

Department of Computer and Information Science, Linköping University, S-581 83 Linköping, Sweden
e-mail: arnjo@ida.liu.se

(*Received 25 April 1997*)

Abstract

Natural language interfaces require dialogue models that allow for robust, habitable and efficient interaction. This paper presents such a model for dialogue management for natural language interfaces. The model is based on empirical studies of human computer interaction in various simple service applications. It is shown that for applications belonging to this class the dialogue can be handled using fairly simple means. The interaction can be modeled in a dialogue grammar with information on the functional role of an utterance as conveyed in the linguistic structure. Focusing is handled using dialogue objects recorded in a dialogue tree representing the constituents of the dialogue. The dialogue objects in the dialogue tree can be accessed by the various modules for interpretation, generation and background system access. Focused entities are modeled in entities pertaining to objects or sets of objects, and related domain concept information; properties of the domain objects. A simple copying principle, where a new dialogue object's focal parameters are instantiated with information from the preceding dialogue object, accounts for most context dependent utterances. The action to be carried out by the interface is determined on the basis of how the objects and related properties are specified. This in turn depends on information presented in the user utterance, context information from the dialogue tree and information in the domain model. The use of dialogue objects facilitates customization to the sublanguage utilized in a specific application. The framework has successfully been applied to various background systems and interaction modalities. In the paper results from the customization of the dialogue manager to three typed interaction applications are presented together with results from applying the model to two applications utilizing spoken interaction.

1 Introduction

This paper presents a computational model for dialogue management for natural language interaction. Theoretical motivations and empirical evidence for the model have been published elsewhere (Dahlbäck 1991*a*; Dahlbäck and Jönsson 1989, 1992; Jönsson and Dahlbäck 1988).

The problem of managing discourse can be divided into three parts (Grosz and Sidner 1986): linguistic structure, attentional state and intentional structure. The need for a component which records the objects, properties and relations that

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

are in the focus of attention, the attentional state, is not much debated, although the details of focusing need careful examination. However, the role that is given to the intentional state, i.e. the structure of the discourse purposes, and to the linguistic structure, i.e. the structure of the sequences of utterances in the discourse, provides two orthogonal (Reichman 1989) approaches to dialogue management. One approach utilizes the linguistic structure to identify the user's intentions and purpose of participating in the discourse. This is then modelled in plans describing the actions that may possibly be carried out in different situations. Proponents of this approach base their work on the model developed by Allen, Cohen and Perrault (Cohen and Perrault 1979; Allen and Perrault 1980), and include Carberry (1990), Litman (1995), Litman and Allen (1990), Ramshaw (1991) and Pollack (1990). In the other approach, utterances are interpreted from the linguistic structure on the basis of their functional relation to the previous interaction. This approach relies on the assumption that the structure of the conversational moves can be used to model the dialogue. The identification of the users' goals is still an important issue, but "Though it is true that conversational moves frequently reflect speakers' goals, it is important to stress that these moves can be identified and interpreted without reference to a speaker's underlying intent for an utterance." (Reichman 1985, p. 21). The idea is that the constraints on what can be uttered allow us to utilize speech act information only, modeled in a dialogue grammar. Examples of projects utilizing this approach are the Dynamic Discourse Model (Polanyi and Scha 1984), LINLIN (Ahrenberg *et al.* 1990), SUNDIAL (Bilange 1991; McGlashan *et al.* 1992), HCRC Map Task (Anderson *et al.* 1991), PLUS (Jokinen 1994), GALAXY (Zue 1994; Seneff *et al.* 1996; Verbmobil Alexandersson and Reithinger 1995; Reithinger and Maier 1995), and WHEELS (Meng *et al.* 1996).

There are two major problems with the approach utilizing plans describing the user's intentions (Jönsson 1996). One is the problem of identifying the primitives needed (Guindon 1988), and the other problem concerns performance of plan recognizers (Kautz 1991). If the goal is to mimic human language capabilities, the plan recognition approach might be necessary despite its problems. But a natural language interface should not try to mimic human interaction. On the contrary, such interfaces will not only be slow, they will also provide the user with an erroneous model of its capabilities. Instead research on dialogue management for natural language interfaces should focus on developing habitable and robust models that correctly and efficiently handle those phenomena that actually occur in human-computer interaction without having the user feel constrained or restricted when using the interface.

Furthermore, the kind of language employed by the user is dependent on the application, resulting in different sublanguages (Grishman and Kittredge 1986), i.e. subsets of natural language, including not only lexicon and grammar variations but also dialogue variations such as how the user and the system handle clarifications, who takes the initiative, etc.

The work presented in this paper is restricted to studying human-computer interaction in natural language, and natural language interfaces for different applications which belong to the domain called simple service systems (Hayes and Reddy 1983).

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

Simple service systems require in essence only that the user identify certain entities, parameters of the service, to the system providing the service, and once they are identified the service can be provided. This domain includes many important applications for natural language interaction, not only relational database applications but also for instance applications utilizing a hierarchically organized database.

More specifically, a dialogue manager is described, responsible for managing focus structure and dialogue structure. The focus structure represents entities mentioned in the discourse and provides the dialogue manager with information to allow a user to refer to them in the course of the interaction. The dialogue structure represents relations between segments in the dialogue and is used to control the dialogue manager.

The paper is organized as follows. Section 2 describes a dialogue manager based on empirical studies of human computer interaction. Section 3 presents details on how the dialogue manager can be customized to account for sublanguages carried out in typed interaction. In section 4 the dialogue manager's actions for task-related initiatives are presented. Section 5 applies the dialogue model to spoken interaction, and section 6 finally summarizes the paper.

2 The dialogue manager

The Dialogue Manager[1] was designed from an analysis of a corpus of 21 typed interaction dialogues, using five different background systems, collected in Wizard of Oz-simulations (Dahlbäck *et al.* 1993).

It can be viewed as a controller of resources for interpretation, background system access and generation. Actual interpretation and generation are carried out by other modules of the interface, including the ability to interpret sentence fragments, multi-sentential, extra-grammatical utterances and anaphora resolution. The Dialogue Manager receives input from the interpretation modules, inspects the result and accesses the background system with information conveyed in the user input. Eventually an answer is returned from the background system access module and the Dialogue Manager then calls the generation modules to generate an answer to the user. If clarification is needed from any of the resources, it is dealt with by the Dialogue Manager.

The Dialogue Manager controls the interaction and holds information needed by the modules in the interface, including the Dialogue Manager itself. The information is modelled in dialogue objects, which represent the constituents of the dialogue. A dialogue object has two components. The first component contains static information describing the properties and relations of the dialogue object. The second component is a process description of a prototypical use of the dialogue object, an action plan (Jönsson 1991).

Two types of static parameters are distinguished: situation parameters and content parameters. The situation parameters specify the situation in which the dialogue

[1] The Dialogue Manager was implemented in Common Lisp and runs on a Sun Sparc station

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

object occurs, such as Initiator, Responder and contextual information. The content parameters model focus structure and dialogue structure, as presented below.

2.1 The dialogue tree

During the course of interaction a dialogue tree is built up from instances of dialogue objects. The dialogue tree serves two purposes. First, it serves as a vehicle for monitoring the dialogue, to guide decisions on how to proceed in the dialogue and where a user move fits into the dialogue, if it is to be regarded as a new initiative, a clarification request, or a response to a system initiative.

The dialogue tree also records the focus parameters to be used by the referent resolving algorithms of the interpretation and generation modules. Local focus is maintained using a simple copying principle where each new dialogue object is instantiated with information from the focus parameters of the previous dialogue object (Ahrenberg *et al.* 1990; Seneff 1992). This forms the initial context for the dialogue object, and is updated with new information from the user initiative. The focus parameters are also updated with the values from the database access module, if provided. Global focus is maintained by recording the dialogue objects in the dialogue tree.

2.2 Focus structure parameters

Focus structure models the entities invoked during the course of interaction. In simple service systems interaction a common user initiative is a request for domain concept information from the database; users specify a database object, or a set of objects, and ask for the value of a property of that object or set of objects (Ahrenberg 1987). Database objects are modeled in the dialogue objects using the parameter Objects and the domain concept information in the parameter Properties. The values of these parameters depend on the background system, and the natural language interface needs to be customized to account for the demands from each application (Jönsson 1993*b*).

The structure of the Objects parameter reflects the structure of the background system. Its value can be explicitly provided as, for instance, it is in *show saab 900 of 1985 model*. However, this is not often the case. Instead, the user provides only partial information, e.g. *show mercedes*, or a new set of objects by specifying properties, e.g. *Show all medium size cars with a safety factor larger than 4*. It is also possible to describe new objects by way of other objects; for instance specifying hotels from the specification of a resort, as in utterance U27 in Figure 1[2].

The Properties parameter models the domain concept in a sub-parameter termed Aspect which can be specified in another sub-parameter termed Value. For instance, *How fast is a Mercedes?*, provides Aspect information on the domain concept, *speed*

[2] The examples are English translations of dialogue fragments from the original Swedish corpus.

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

U27: what hotels are there on Crete
S28: Wait...
The hotels in Platanias on Crete are:
Kronos, Agrimia, Village Suites and Villa Margarita.
U29: which one is the cheapest
S30: Wait...
Price Villa Margarita:

	4/7	11/7	18/7	25/7
1 week	2725	2725	2675	2625
2 weeks	3272	3275	3225	3175

U31: what is next cheapest
S32: Wait...
Price Village Suites:

	4/7	11/7	18/7	25/7
1 week	3150	3150	3100	3050
2 weeks	4025	4025	3975	3925

U33: what service can these hotels provide
S34: Wait...
The basic price at Villa Margarita includes shared apartment.
Cleaning 3 days/week, no dish-washing. No meals.
The basic price at Village Suites includes shared apartment.
Cleaning 3 days/week, no dish-washing. No meals.

Fig. 1. From the TRAVEL information and order corpus, dialogue 8.

which when specified by the database manager provides information on the Value of the Aspect.

2.3 Dialogue structure parameters

The model for dialogue management assumes that a dialogue is divided into three main levels on the basis of structural complexity. One level corresponds to the size of a dialogue (D), another level to the size of a discourse segment and a third level to the size of a single speech act, or dialogue move. Thus, a dialogue is structured in terms of discourse segments, and a discourse segment in terms of moves and embedded segments. Utterances are not analyzed as dialogue objects, but as linguistic objects which function as vehicles of one or more moves. An initiative-response (IR) structure is assumed where an initiative opens a segment by introducing a new goal and the response closes the segment (Dahlbäck 1991*b*).

There are various other proposals as to the number of categories needed. They differ mainly on the modelling of complex units that consist of sequences of discourse segments, but do not comprise the whole dialogue. For instance, LOKI (Wachtel 1986) and SUNDIAL (Bilange 1991) use an intermediate level characterized by having a common topic, i.e. an object whose properties are discussed over a sequence of exchanges. However, a sequence of segments may be connected in a number of different ways (Ahrenberg *et al.* 1990), e.g. by dealing with one object for which

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

different properties are at issue. But it may also be the other way around, so that the same property is topical, while different objects are talked about.

To specify the functional role of a move, two parameters, Type and Topic, are used. Type corresponds to the illocutionary type of the move. In simple service systems two sub-goals can be identified (Hayes and Reddy 1983, p. 266): (1) "specify a parameter to the system" and (2) "obtain the specification of a parameter". Initiatives are categorized accordingly as being of two different types: (1) update, *U*, where users provide information to the system and (2) question, *Q*, where users obtain information from the system. Responses are categorized as answer, *A*, for database answers from the system or answers to clarification requests. Other Type categories are *Greeting*, *Farewell* and *Discourse Continuation* (Dahlbäck 1991*b*), the latter being used for utterances from the system whose purpose is to keep the conversation going.

Topic describes which knowledge source to consult. In information retrieval applications three different topics are used: the database for solving a task, *T*, acquiring system-related information about the application, *S*, or finally, the ongoing dialogue, *D*. If the background system allows update, e.g. ordering of a specified item, a fourth knowledge source is needed to account for this. A similar knowledge source could contain background information such as current day etc.

2.4 Dynamic behaviour

The control of the dialogue is distributed to the nodes in the dialogue tree. Every node in the tree is responsible for its own correctness through the action plan which describes the actions to carry out in order to perform a dialogue task. This has the advantage that the action plans, as they only have local scope, are very simple. Furthermore, they are more generally applicable.

The action plan of a dialogue object builds the dialogue tree bottom-up with a top-down prediction from the context. Normally, for user initiated dialogues, an IR-node creates an instance of a user move-node. The move-node interprets the user initiative and integrates it into the tree. Information from the move-node is then available also at the IR-node from which it was initiated. If the database manager fails to access the database due to lack of information, it signals this to the IR-node which initiated the database request. This node creates an instance of an IR-unit for a clarification request and inserts it into the dialogue tree. The action plan in the new IR-node for clarification request generates a move explaining the missing information and creates a user move waiting for user input. This user move-node is interpreted and based on the values of its static parameters and the parameters of the initiating IR-node it is appropriately integrated into the dialogue tree. There are roughly three ways to do this:

- If it satisfies the expectations for an answer, it is connected as a response to the initiating IR-node.
- If it constitutes a new initiative, it will be connected to a new IR-node, which in turn will be connected to the dialogue tree with the values from the initiating

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

IR-node copied to the new IR-node. Interaction then proceeds from this new IR-node.

- If it initiates a clarification request to the system's clarification request, it will be connected to a new IR-node which in turn will be connected to the initiating IR-node. The treatment of multiple sequential clarifications follows the same pattern as that for one clarification subdialogue. There is, in principle, no limit on the number of parallel subdialogues allowed and no limit on the depth of subdialogues.

If we compare the use of dialogue objects to the execution of a plan scheme, the preconditions of a plan description can be seen as the initial static part of the dialogue object. Thus, when, for instance, a move dialogue object has been executed and is to be connected somewhere in the tree, the node in the tree that it is to be connected to is identified by looking at the values of the attributes in the static component of the nodes in the dialogue tree. When the correct object is identified, the rest of the body of that object is executed, i.e. the dynamic part of the dialogue object is carried out. Furthermore, the static part of the identified object can be given new values for some of its attributes as an effect of the execution of the body. Thus, the static part of the dialogue object at event time t resembles preconditions in a plan scheme, while the static part at event time $t+1$ resembles the effects. The dynamic part of the dialogue object resembles the body of the plan schemes. What differs in the two approaches is the content of the objects and how to identify the correct object to be used. In the plan-based approaches, the operators in the plan scheme are used to reason about the user's goals and intentions and their effects. Thus, plans describe actions which may possibly be carried out in different situations. The correct plan is found by searching through the various possible plan schemes in the domain, looking for plans which satisfy the users goals and intentions. In the approach proposed here, users also have specific goals in mind. However, a conversational move is interpreted without reference to the intention behind the goal. Information about what speech act is being carried out and its functional relation to the previous interaction and the specifications of the focal parameters is sufficient.

3 Applying the dialogue model to typed interaction

The Dialogue Manager has been customized to account for the sublanguage arising in three different typed interaction applications (Jönsson 1993*a*; Ahrenberg *et al.* 1996; Jönsson 1995). The CARS application (see Figure 2 for an example) is a relational database with information about second-hand cars. For this application a prototype natural language interface has been implemented which accesses an INGRES database system. The TRAVEL application (see figure 1) provided information about charter trips to the Greek archipelago. The TRAVEL application utilizes a hierarchically structured database and was customized for two different configurations, one allowing only information retrieval, TRAVEL1, and another allowing users to also order a specific trip, TRAVEL2. The customization was based on a corpus of

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

1750 utterances from 30 dialogues, 10 for each application, collected in Wizard of Oz-experiments.

Customization of the Dialogue Manager involves two major tasks: (1) Defining the focal parameters of the dialogue objects in more detail and customizing the heuristic principles for changing the values of these parameters; (2) Constructing a dialogue grammar for controlling the dialogue.

3.1 Customizing the focus structure

For the CARS application the focused objects are cars described by the sub-parameters (Manufacturer, Model, Year). The TRAVEL application utilizes a three-layered database: the Greek archipelago, the various resorts and finally the hotels at each resort. However, from the empirical investigations, it turns out that there is no need to explicitly represent the various levels in the hierarchy. Instead one single sub-parameter holding any of these object types is sufficient, i.e. an even simpler Objects parameter than the CARS application. To illustrate this, consider figure 1. After utterance U27 the value of the Objects parameter is the resort *Crete*. This will be changed to a set of hotels when the response from the background system is generated, S28.

The general heuristic focusing principles need to be slightly modified before they can be applied to the CARS and TRAVEL applications. For the CARS application the heuristic principles apply well to the Objects parameter. An intentionally specified object description provided in a user initiative will be replaced by the extensional specification provided by the module accessing the database, which means that erroneous objects will be removed, as they will not be part of the response from the database manager. For the TRAVEL application the principles for providing information to the Objects parameter are modified to allow hotels to be added if the resort remains the same.

Figure 1 illustrates an interesting phenomena found only in the application utilizing a hierarchically structured database, TRAVEL. In U27 the user picks out a set of hotels at a resort but is only interested in a subset of them. If we apply the heuristic principle that hotels are appended to the Objects parameter if the resort remains the same, the Objects parameter will hold the correct subset requested in U33. However, the database search in U31 should be restricted to the set specified in S28. This is accomplished by introducing a focus parameter termed Secondary Objects, to hold the subset from which individual objects are investigated Jönsson 1993*a*.

The heuristic principles for the Properties parameter for the CARS application need to be modified. The principle is that if the user requests information about a set of cars that is a subset of the cars already in Objects, then the properties provided in the new user initiative are added to the old set of properties. This is based on the observation that users often start with a rather large set, in this case a set of cars, and then gradually specify a smaller set by adding restrictions (cf. Kaplan 1983), for instance in CARS using utterances such as *remove all small size cars*. For the TRAVEL application the copy principle holds for the Properties parameter without exception.

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

3.2 Customizing the dialogue structure

The dialogue grammar is developed by constructing a minimal dialogue grammar from an analysis of dialogues from the application, or an application of the same type, e.g. information retrieval from a database. This grammar is generalized and extended, using general knowledge on human-computer natural language interaction, with new rules to cover "obvious" additions not found in the initial grammar. Restrictions can also be imposed, for instance limiting the recursive depth of subdialogues Jönsson 1993*b*. In the CARS dialogues extensions included, for instance, Greetings and Farewells, which did not appear in the analysis of the dialogues. In the TRAVEL system it involved, among other things, allowing for multiple clarification requests and clarification requests not answered by the user. Some extensions not found in any of the dialogues were also added, for instance, a rule for having the system prompt the user with a discourse continuation if it becomes uncertain who has the initiative. However, if a phenomenon requires sophisticated and complex mechanisms, it will be necessary to consider what will happen if the interface is used with a grammar without that addition. This also includes considering how probable it is that a certain phenomenon may occur.

3.3 An example

To illustrate the behaviour of the customized Dialogue Manager consider the dialogue fragment in figure 2. Initially, the Dialogue Manager creates an instance of a dialogue-node, to capture the whole dialogue, which in turn creates an IR-node, IR_1, and inserts it in the dialogue tree. IR_1 creates an instance of a move-node which interprets the first utterance and integrates it into the dialogue tree making the information available to IR_1. The first utterance, U1, is a task-related user initiative, a Q_T[3], which correctly partly specifies an object Objects=(Manufacturer: *Mercedes*, Model: , Year:). However, it does not specify domain concept information[4].

Thus, IR_1 initiates a clarification subdialogue by creating a new IR-node, IR_{11}, and connects this to IR_1. The action plan of IR_{11} creates an instance of a system move-node. From the lack of value to the Properties parameter, IR_{11} concludes that the user needs to provide domain concept information and S2 is generated. The system initiative S2 consists of one part providing system information and one part providing an explicit request for user information. The next action of IR_{11} is to create an instance of a user move-node which interprets the response from the user. IR_{11} is an IR-node of type Q_D/A_D which means that a discourse-related question should be followed by a discourse-related answer. The first part, the Q_D, was generated by the system. The second part, A_D, carries the information that only an answer is regarded as a valid response. The Aspect slot is used to hold the

[3] For brevity, when presenting the dialogue grammar, the Topic of a move is indicated as a subscript to the Type.

[4] One could argue that the user asks for all information. This would, however, be too much information to display and thus a similar clarification subdialogue would be initiated from the system.

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

U1: show mercedes
S2: Wait...
Cars can answer questions about a number of car models in regard to make, model designation, year, country of manufacture, susceptibility to rust, and size class, as well as questions about cost, driving and collision safety, interior (cabin) and baggage (boot) space, and technical data.
Any special requests?
U3: cost and space
S4: Wait...
Checking...

Manufacturer	Model	Year	price	cabin	boot
Mercedes	200	1982	66400	4	370
Mercedes	200	1985	90000	4	370
Mercedes	190	1988	149000	3	320
Mercedes	200	1988	180300	4	370

Fig. 2. From the CARS corpus, dialogue 1.

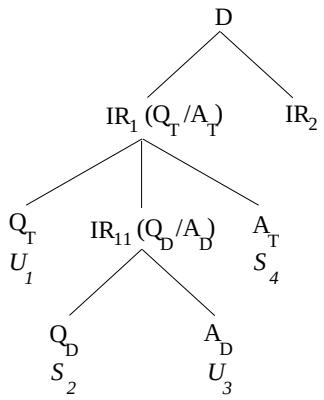


Fig. 3. The dialogue tree for the dialogue in figure 2.

parameter for which the system wants an answer and the Value slot is used for the user's answer. If the user answers correctly, the values for Properties in the initiating IR-unit are updated. A Q_D/A_D-unit is identified from the type information, i.e. the Type of the response from the user is A. Otherwise the user move is regarded as not being an answer to the system request. In this case, the user, in U3, provides an answer to the clarification request, an A_D. The user move-node is connected to IR_{11} and updates its Properties parameter with the values *cost* and *space*.

This information is made available to IR_1 which can submit a request to the database access module. The response from the database manager updates the Objects parameter of IR_1 to the four objects presented and also updates the Properties parameter. Thus, the Objects parameter now consists of a set of four objects, not one as in the initial request, U1. IR_1 creates an instance of a generate-move, an A_T to present the result to the user, S4. Finally, IR_1 signals to the top-level dialogue node, D, that it is done and the interaction proceeds from the D-node which again instantiates an IR-node, IR_2, ready to interpret a new user initiative.

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

Table 1. *User-initiatives classified according to context-dependence*

	CARS	TRAVEL1	TRAVEL2
Fully specified	52%	44%	59%
Local context needed	43%	50%	39%
Global context needed	5%	6%	2%

Table 2. *Types of dialogue segments and their relative frequency in three different applications*

	CARS	TRAVEL1	TRAVEL2
Number of rules	15	12	14
Q_T/A_T	60%	83%	70%
Q_D/A_D	12%	2%	2%
Q_S/A_S	9%	2%	2%
Q_T/A_D	7%	2%	3%
Q_T/A_S	5%	6%	4%
Others[a]	7%	5%	19%

[a] Others denote, for instance, Ordering rules, Greetings, and Farewells.

Initially IR_2 has values to its context parameters copied from IR_1 as described above. Figure 3 shows the resulting dialogue tree.

3.4 Empirical results

The usability of the dialogue management framework can be illustrated from a quantitative analysis of focus and dialogue structure complexity and variability.

Table 1 presents the results on context-dependence from the customization to the CARS, TRAVEL1 and TRAVEL2 applications. A user initiative can be:

- Fully specified, i.e. the utterance is complete without information from context,
- Local context information dependent, i.e. the utterance can be interpreted from information found in the current segment as copied from the previous segment or
- Global context information dependent, i.e. a search in the dialogue tree is needed to find the referent of the indexical utterance.

From the table we can see that the large majority of user inputs in the corpus can be handled by the fairly simple context model of copying information from the previous IR-node to the new one. In CARS 95% of the utterances could be interpreted without a search in the dialogue tree. For the TRAVEL application the corresponding figures are 94% without ordering and 98% if ordering was utilized.

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

The complexity of dialogue structure can be illustrated from statistics on the number of rules and the variability in the rules utilized. A summary of the statistics on dialogue structure is presented in Table 2[5]. It shows that the number of rules is limited; a maximum of 15 rules are needed. It turns out that a grammar of in total 21 rules is sufficient in order to model all three applications (Jönsson 1993*a*). Furthermore, we can see that the variability is limited. The most common segment consists of a task-related initiative followed by an answer from the database, Q_T/A_T, sometimes with an embedded clarification sequence, Q_D/A_D. In CARS 60% of the initiatives start segments of this type. For TRAVEL 83% of the initiatives in the non-ordering dialogues and 70% of the ordering dialogues are of this type. Consequently, the interfaces are most often used appropriately for getting information from the background system.

4 Actions for task-related initiatives

This section connects the two dialogue management information sources; focus structure and dialogue structure. Normally a natural language interface to database information retrieval applications is user-directed, i.e. the user initiates a request for information from the background system and the interface responds with the requested information. The interface in principle only takes the initiative to begin a clarification request under three circumstances[6]:

- A difficulty arises when interpreting an utterance, e.g. unknown words or questions outside the domain of the database.
- A difficulty arises when accessing the database, e.g. when the user needs to provide a parameter for correct access.
- A difficulty arises in the presentation of the result from the database access, e.g. the answer is too large to print on one screen.

The action to be carried out for task-related questions depends on the specification of the values passed to the focal parameters Objects and Properties. This specification in turn depends on information from the user initiative together with the information copied from the previous IR-unit and context information from the dialogue tree and the answer from the database system. This contrasts to other approaches based on dialogue acts, such as SUNDIAL (Bilange 1991) or the negotiating dialogues of Verbmobil (Smitz and Quantz 1995), where a combination of the illocutionary act and domain-specific characteristics determines the system's behaviour.

The description of an object or property can be either: correct, partly correct, incompatible, ambiguous, erroneous, or not provided. Erroneous means that the user has specified an object which is not in the database. Partly correct means that the description contains at least one correct object or property description, but also one or more erroneous descriptions. Incompatible descriptions utilize elements which do not belong together, e.g. Volvo Camry.

[5] Labels of IR-segments have the form of a pair of move labels separated by a slash (/).
[6] The system also takes the initiative when collecting information in a frame dialog (Hoeppner *et al.* 1983; Hoeppner *et al.* 1986), as will be discussed below.

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

Table 3. *A summary of the Dialogue Manager's actions to task-related initiatives*

Objects	Properties	Resulting action(s)
Correct Partly correct Not provided	Correct Partly correct Aspect	A_T
Correct Partly correct Not provided Incompatible	Erroneous Value Ambiguous Aspect	Q_D/A_D A_T
Correct	Not provided	Q_D/A_D A_T
Erroneous	–	A_S
–	Erroneous Aspect	A_S
Incompatible		A_S
(Too large to print)		Q_D/A_D A_T

The relation between the values to the Objects and Properties parameters and the resulting action described in terms of Type and Topic is summarized in table 3. In principle, any combination of Objects and Properties in a cell in a row results in the action shown in the right column. From the table we can identify three basic actions to task-related IR-units depending on the values of the parameters Objects and Properties: A_T, Q_D/A_D A_T and A_S.

4.1 Responses expressing task information

The most common action responding to a successful task-related user initiative, a Q_T, is a task-related system answer with information taken from the database, A_T. This requires correct, or partly correct, values for Objects and correct, or partly correct, Aspect values for the Properties parameter. The values for these parameters can be taken either from the preceding dialogue or they can be provided in the user input. What is important is that the initiative in context provides enough information so that it can be used to access the background system and that the answer from the background system is in some sense correct. A special case, also resulting in an A_T, is when no object description is provided but the Properties are fully specified and can be used to access the database, e.g. *show all medium class cars costing less than 70,000 crowns.*

4.2 Clarification requests

A clarification request, a Q_D/A_D A_T, is to be considered as a special case of the normal A_T-action as specified above. Such a clarification subdialogue is utilized by

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

U11: What is the shape of Ford Fiesta costing 26,800 crowns?
[Q_T, Correct Objects, Erroneous Properties]
S12: Wait...
Cars cannot answer questions concerning the shape of car models.
[A_S]
U13: Is it rusty?
[Q_T, Correct Objects, Correct Properties]
S14: Wait...
Checking...

Manufacturer	Model	Year	Rust
Ford	Fiesta	1982	2

[A_T]

Fig. 4. From the CARS corpus, dialogue 13.

the system to achieve more information from the user in order to get fully and correctly specified values for Objects or Properties.

It is initiated when no value for the Objects parameter is specified or when the values are correctly, or partly correctly, specified but the values of the Value slot in Properties are erroneous or under specified. For instance, in *remove all cars with low operational safety* the expression "low" is too vague, resulting in a clarification request to the user to be more specific. Another case is where no Aspect is provided or the provided Aspect is ambiguous, e.g. *which 10 car models between 60 000 and 70 000 crowns are most spacious* where information on "space" can be either coupe space or boot space. A clarification subdialogue is not initiated unless the system is able to explicitly provide alternatives to the user.

A special case of clarification request occurs when correct specifications to the parameters Objects and Properties are provided, but the answer is too large to print on the screen. In such cases the system initiates a clarification subdialogue asking the user to restrict the number of items to be printed.

4.3 Responses expressing information about the capabilities of the background system

A_S is used for task-related user initiatives resulting in a system answer which provides information about the database system. Information can be provided on various aspects of what type of information there is in the database and what type of questions can be used to elicit this information. An A_S is utilized for any utterance with erroneous Objects or Aspect. Incompatible Properties and Objects also result in an A_S, which means that although both Properties and Objects are correct, they cannot be used together.

There are also user initiatives which do not depend on the values of Objects and Properties, such as system-related questions, Q_S, i.e. a user's request for information about the system. These are recognized on the grounds of linguistic information provided by the syntactic/semantic analyzer (Ahrenberg 1988).

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

4.4 An example

To illustrate how the action scheme in figure 3 is utilized consider utterance U11 *What is the shape of Ford Fiesta costing 26,800 crowns?* in figure 4. This will be interpreted as a task-related question, a Q_T, with correctly specified Objects parameter. However, the Aspect sub-parameter is erroneous, as there is no information in the database on the concept 'shape'. Furthermore, the system can not provide alternatives to the user, i.e. the system has no information on which properties in the database can denote 'shape'. Thus, the resulting action is an A_S, S12. The next user utterance, U13, does not follow up the 'shape' concept, instead another property, 'rust', is investigated. This utterance is a Q_T with correct Objects, as copied from the previous IR-unit, and a correct Aspect sub-parameter, 'rust'. Thus, the resulting action is a normal A_T, S14, with information taken directly from the database.

5 Applying the dialogue model to spoken interaction

The dialogue model was developed for typed interaction. However, there are reasons to believe that it also can be used in spoken interaction. Theoretical motivations for similarities between spoken and typed interaction are based on the observation that some of the characteristics of a dialogue originate from the fact that it is a dialogue, not if it is spoken or typed (Dahlbäck 1997).

This section presents results from applying the dialogue model to applications utilizing simulated speech interaction (Jönsson 1996). In one application, the SUNDIAL application, users can request flight information via telephone[7]. From this corpus 100 dialogues from ten different subjects have been studied. The other application, the WAXHOLM application[8], provides information on boat traffic in the Stockholm archipelago (Bertenstam *et al.* 1995). In this application, spoken interaction is integrated with text and graphics. From this corpus 10 dialogues were studied.

It is not a straightforward task to compare the results of the analysis of the WAXHOLM and SUNDIAL applications to the CARS and TRAVEL applications. One reason is the design of the system and the properties of the application which influence the analysis and the user's behaviour (Ahrenberg *et al.* 1996). This can be illustrated from how the system responds to utterances which are not specified in enough detail to access the background system. In the CARS and TRAVEL applications the response provides information explaining the system's abilities, without stating any explicit request for new information, e.g.

User: *price*
System: *The price depends on which hotel you choose, when you travel and how long you are away*

[7] These dialogues are from a corpus of Wizard of Oz-dialogues collected and transcribed at the Social and Computer Sciences, University of Surrey, UK as part of the ESPRIT Sundial project (P2218).

[8] These dialogues are from a corpus of Wizard of Oz-dialogues collected and transcribed at the Department of Speech, Music and Hearing, Royal Institute of Technology, Stockholm, Sweden.

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

The segment is closed and a new IR-unit initiated. In WAXHOLM the system instead initiates a clarification request subdialogue asking the user to specify the necessary parameters, (cf. frame dialog Hoeppner *et al.* 1986), e.g.

User: : *I want to go to Vaxholm tonight*
System: *Where do you want to leave from*?
User: *From Stockholm*
System: *What day of the week do you want to go?*
User: *On Thursday*

Frame dialog sequences are used in the TRAVEL2 application where ordering is allowed to collect necessary ordering information, but not otherwise. A difference is that in WAXHOLM the focal parameters are updated whereas in TRAVEL2 an order form is utilized.

Taking into account these differences, which mainly depend on the application and the system design, no important deviations arise from the customization of the written applications, CARS and TRAVEL. The SUNDIAL corpus mainly consists of short fully specified task-related user initiatives followed by an answer from the system. The derived dialogue grammar comprises 13 rules. Only 25% of the user initiatives, require local context, 69% are fully specified and 6% are underspecified, i.e. the system utilizes a frame dialog to acquire more information in order to access the background system. None requires global context.

The WAXHOLM corpus has not been collected solely for the purpose of studying dialogue but also for investigating properties of speech recognition in a noisy environment. Thus, the Wizard often decides not to understand a user's initiative, which results in a clarification request. Such segments do not pose any problem to the Dialogue Manager, but as the proportion of such segments is high, quantitative data from the material will be misleading. However, the copying principle for managing focus holds, only two utterances require global context, and the number of grammar rules is limited.

There is one type of phenomenon which is typical for spoken interaction and which is not found in the written interactions, namely when a user interrupts the system. For instance, the SUNDIAL system always prompts the user with *please wait* to acknowledge the initiative and then the answer follows when the system has accessed the database. In spoken interaction the user can interrupt, for instance uttering *pardon could you repeat that please* or *thank you*. Such types of interruptions do not add new information; however, it can also be the case that the user modifies her initiative, e.g. *pardon flight two two seven*. This is a type of dynamic behaviour which does not violate the model but the Dialogue Manager's dynamic behaviour needs to be modified to allow the focal parameters to be updated and a new access to the background system be carried out with the modified request before an answer is provided.

There are, however, types of dialogue where the dialogue management requires more sophisticated reasoning or where the task gives rise to a more complicated dialogue. For instance, interfaces to task oriented applications, might require more sophisticated reasoning involving the user's task and goals in order to be helpful (cf.

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

Burger and Marshall 1993). Furthermore, combining telephone interactions with a complex assembly task (Oviatt and Cohen 1991) demand a more complicated type of dialogue. In these cases we are faced with a different communicative situation where it is easier to be engaged in a more human-like interaction. For a discussion on the complexity of dialogue models for various tasks see Dahlbäck (1997), Van Loo and Bego (1993) and Ishizaki *et al.* (1996).

6 Summary

This paper has presented a computational approach to the design of dialogue managers for habitable user-friendly natural language interaction. Natural language interfaces for such interaction must take into consideration the characteristics of human computer interaction instead of trying to mimic human interaction. Consequently, the Dialogue Manager has been developed to handle the dialogue for the sublanguage of human computer interaction.

The dialogue management information is modelled in dialogue objects which represent the constituents of the dialogue. Instances of dialogue objects are recorded in a dialogue tree as the interaction advances. Referring expressions are handled by copying information from the previous segment to the current segment which in turn is updated with information from the background system. The dialogue can be controlled from information conveyed in the speech act directly, no reasoning about users' intentions is necessary. Speech act information is assembled into Initiative-Response units which form the basis for interpreting the segment structure. A simple context free grammar models the interaction and the rules are selected on the basis of the specification of properties of objects describing the information provided by the application.

One important aspect of the Dialogue Manager is that it can be adapted to account for the dialogue arising in various applications. The adaptation to different applications is achieved by customizing the dialogue objects, i.e. by specifying which parameters to use and what values they can take. Customization has successfully been carried out for three typed interaction applications. The model has also been applied to applications for spoken interaction.

Acknowledgements

This work results from a project on Dynamic Natural-Language Understanding supported by The Swedish Council of Research in the Humanities and Social Sciences (HSFR) and The Swedish National Board for Industrial and Technical Development (NUTEK) in the joint Research Program for Language Technology and a grant from The Swedish Transport & Communications Research Board (KFB). The work has been carried out at the Natural Language Processing Laboratory at Linköping University, Sweden, and I am especially indebted to Lars Ahrenberg, Nils Dahlbäck and Åke Thurée. Thanks also to Jonas Lundberg and Jonas Kvist for help with the empirical analyses.

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

References

Ahrenberg, L. (1987) Interrogative Structures of Swedish. Aspects of the Relation between Grammar and Speech Acts. *PhD thesis*, Uppsala University.

Ahrenberg, L. (1988) Functional constraints in knowledge-based natural language understanding. *Proceedings of the 12th International Conference on Computational Linguistics, Budapest*, pp. 13–18.

Ahrenberg, L., Dahlbäck, N., Jönsson, A. and Thurée, Å. (1996) Customizing interaction for natural language interfaces. *Linköping Electronic Articles in Computer and Information Science*. Also in *Notes from Workshop on Pragmatics in Dialogue, The XIVth Scandinavian Conference of Linguistics and the VIIIth Conference of Nordic and General Linguistics*, Göteborg, Sweden, 1993, 1(1).) http://www.ep.liu.se/ea/cis/1996/001/.

Ahrenberg, L., Jönsson, A. and Dahlbäck, N. (1990) Discourse representation and discourse management for natural language interfaces. *Proceedings of the Second Nordic Conference on Text Comprehension in Man and Machine*, Täby, Sweden.

Alexandersson, J. and Reithinger, N. (1995) Designing the dialogue component in a speech translation system. *Proceedings of the Ninth Twente Workshop on Language Technology (TWLT-9)*, pp. 35–43.

Allen, J. F. and Perrault, C. R. (1980) Analysing intention in utterances. *Artificial Intelligence* **15**: 143–178.

Anderson, A. H., Bader, M., Bard, E. G., Boyle, E., Doherty, G., Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., Sotillo, C., Thompson, H. and Weinert, R. (1991) The HCRC map task corpus. *Language and Speech* **34**(4): 351–366.

Bertenstam, J., Blomberg, M., Carlson, R., Gustafson, J., Hunnicutt, S., Högberg, J., Lindell, R., Neovius, L., de Serpa-Leitao, A., Nord, L. and Ström, N. (1995) Spoken dialogue data collection in the waxholm project. STL-QPSR 1/1995, pp. 50–73.

Bilange, E. (1991) A task independent oral dialogue model. *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics*, Berlin.

Burger J. D. and Marshall, R. J. (1993) The Application of Natural Language Models to Intelligent Multimedia. In: Maybury M. T., (ed.), *Intelligent Multimedia Interfaces*, pp. 174–196. MIT Press.

Carberry, S. (1990) *Plan Recognition in Natural Language Dialogue*. MIT Press.

Cohen, P. R. and Perrault, C. R. (1979) Elements of a plan-based theory of speech acts. *Cognitive Science* **3**: 177–212.

Dahlbäck, N. and Jönsson, A. (1989) Empirical studies of discourse representations for natural language interfaces. *Proceedings from the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, Manchester, UK.

Dahlbäck, N. and Jönsson, A. (1992) An empirically based computationally tractable dialogue model. *Proceedings of the Fourteenth Annual Meeting of The Cognitive Science Society*, Bloomington, Indiana.

Dahlbäck, N. (1991*a*) Empirical analysis of a discourse model for natural language interfaces. *Proceedings of the Thirteenth Annual Meeting of The Cognitive Science Society*, Chicago, IL, pp. 1–6.

Dahlbäck, N. (1991*b*) Representations of Discourse, Cognitive and Computational Aspects. *PhD thesis*, Linköping University.

Dahlbäck, N. (1997) Towards a dialogue taxonomy. In: Maier, E., Mast, M. and LuperFoy, S. (eds.), *Dialogue Processing in Spoken Language Systems*. Lecture Notes in Artificial Intelligence. Springer-Verlag.

Dahlbäck, N., Jönsson, A. and Ahrenberg, L. (1993) Wizard of Oz studies – why and how. *Knowledge-Based Systems* **6**(4): 258–266.

Grishman, R. and Kittredge, R. I. (1986) *Analysing Language in Restricted Domains*. Lawrence Erlbaum.

Grosz, B. J. and Sidner, C. L. (1986) Attention, intention and the structure of discourse. *Computational Linguistics* **12**(3): 175–204.

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

Guindon, R. (1988) A multidisciplinary perspective on dialogue structure in user-advisory dialogues. In: Guindon, R. (ed.), *Cognitive Science and Its Applications For Human-Computer Interaction*. Lawrence Erlbaum.

Hayes, P. J. and Reddy, D. R. (1983) Steps toward graceful interaction in spoken and written man-machine communication. *International Journal of Man-Machine Studies* **19**: 231–284.

Hoeppner, W., Christaller, T., Marburger, H., Morik, K., Nebel, B., O'Leary, M. and Wahlster, W. (1983) Beyond domain experience: Experience with the development of a German language access system to highly diverse background systems. *Research Report Bericht ANS-16*, University of Hamburg.

Hoeppner, W., Morik, K. and Marburger, H. (1986) Talking it over: The natural language dialog system ham-ans. In: Bolc, L. and Jarke, M. (eds.), *Cooperative Interfaces to Information Systems*. Springer-Verlag.

Ishizaki, M., Den, Y., Tutiya, S., Tamoto, M. and Nakazato, S. (1996) Classifying Dialogue Tasks: Task Oriented Dialogue Reconsidered. *Proceedings of ISSD'96*, Philadelphia, pp. 31–40

Jokinen, K. (1994) Response Planning in Information-Seeking Dialogues. *PhD thesis*, University of Manchester Institute of Science and Technology.

Jönsson, A. (1991) A dialogue manager using initiative-response units and distributed control. *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics*, Berlin.

Jönsson, A. (1993*a*) Dialogue Management for Natural Language Interfaces – An Empirical Approach. *PhD thesis*, Linköping University.

Jönsson, A. (1993*b*) A method for development of dialogue managers for natural language interfaces. *Proceedings of the Eleventh National Conference of Artificial Intelligence*, Washington, DC, pp. 190–195.

Jönsson, A. (1995) A dialogue manager for natural language interfaces. *Proceedings of the Pacific Association for Computational Linguistics, Second conference*, University of Queensland, Brisbane, Australia.

Jönsson, A. (1996) Natural language generation without intentions. *Proceedings of ECAI'96 Workshop Gaps and Bridges: New Directions in Planning and Natural Language Generation*, pp. 102–104.

Jönsson, A. (1996) A Model for Dialogue Management for Human Computer Interaction. *Proceedings of ISSD'96*, Philadelphia, pp. 69–72.

Jönsson, A. and Dahlbäck, N. (1988) Talking to a computer is not like talking to your best friend. *Proceedings of the First Scandinavian Conference on Artificial Intelligence*, Tromsø.

Kaplan, S. J. (1983) Cooperative responses from a portable natural language database query system. In *Computational Models of Discourse*, pp. 167–208. MIT Press.

Kautz, H. A. (1991) A Formal Theory of Plan Recognition and its Implementation. In: Allen, J. A., Kautz, H. A., Pelavin, R. N. and Tenenberg, J. D. (eds.), *Reasoning About Plans*, pp. 69–125. Morgan Kaufmann.

Litman, D. J. (1985) Plan Recognition and Discourse Analysis: An Integrated Approach for Understanding Dialogues. *PhD thesis*, University of Rochester.

Litman, D. J. and Allen, J. F. (1990) Discourse processing and commonsense plans. In: Cohen, P. R., Morgan, J. and Pollack, M. E. (eds.), *Intentions in Communication*, pp. 365–388. MIT Press.

Van Loo, W. and Bego, H. (1993) Agent Tasks and Dialogue Management. *Workshop on Pragmatics in Dialogue, The XIV:th Scandinavian Conference of Linguistics and the VIII:th Conference of Nordic and General Linguistics*, Göteborg, Sweden.

McGlashan, S., Fraser, N., Gillbert, N., Bilange, E., Heisterkamp, P. and Youd, N. (1992) Dialogue management for telephone information systems. *Proceedings of the Third Conference on Applied Natural Language Processing*, Trento, Italy, pp. 245–246.

Meng, H., Busyapongchai, S., Glass, J., Goddeau, D., Hetherington, L., Hurley, E., Pao, C., Seneff, S. and Zue, V. (1996) Wheels: A conversational system in the automobile classifieds domain. *Proceedings of ISSD'96*, Philadelphia, pp. 165–168.

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press

Oviatt S. L. and Cohen P. R. (1991) The Contributing Influence of Speech and Interaction on Human Discourse Patterns. In: Sullivan J. W. and Tyler S. W. (eds.), *Intelligent User Interfaces*. ACM Press/Addison-Wesley.

Polanyi, L. and Scha, R. (1984) A syntactic approach to discourse semantics. *Proceedings of the 10th International Conference on Computational Linguistics*, Stanford, CA.

Pollack, M. E. (1990) Plans as complex mental attitudes. In: Cohen, P. R., Morgan, J. and Pollack, M. E. (eds.), *Intentions in Communication*. MIT Press.

Ramshaw, L. A. (1991) A three-level model for plan exploration. *Proceedings of the 29th Annual Meeting of the ACL*, Berkeley, CA, pp. 39–46.

Reichman, R. (1985) *Getting Computers to Talk Like You and Me*. MIT Press.

Reichman, R. (1989) Convention versus intention. In: Taylor, M. M., Néel, F. and Bouwhuis, D. G. (eds.), *The Structure of Multimodal Dialogue*, pp. 121–134. Elsevier.

Reithinger, N. and Maier, E. (1995) Utilizing statistical dialogue act processing in vermobil. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL-95)*, Cambridge, MA, pp. 116–121.

Seneff, S. (1992) A relaxation method for understanding spontaneous speech utterances. *Fifth DARPA Workshop on Speech and Natural Language*.

Seneff, S., Goddeau, D., Pao, C. and Polifroni, J. (1996) Multimodal discourse modeling in a multi-user multi-domain environment. *Proceedings of ISSD'96*, Philadelphia, pp. 105–108.

Smitz, B. and Quantz, J. J. (1995) Dialogue acts in automatic dialogue interpreting. *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, Leuven, Belgium, pp. 33–48.

Wachtel, T. (1986) Pragmatic sensitivity in nl interfaces and the structure of conversation. *Proceedings of the 11th International Conference of Computational Linguistics*, Bonn, Germany, pp. 35–42.

Zue, V. W. (1994) Toward systems that understand spoken language. *IEEE Expert* **9**: 51–59.

https://doi.org/10.1017/S1351324997001733 Published online by Cambridge University Press