# Design Constraints and Representation for Dialogue Management in the Automatic Telephone Operator Scenario

**José F. Quesada, J. Gabriel Amores, Gabriela Fernández, José A. Bernal, M. Teresa López**

University of Seville (Spain)
jquesada@cica.es

**Abstract**

This paper describes the dialogue layer of a Spoken Dialogue System prototype. We have developed a methodology for the design of dialogue management systems following Knowledge Engineering principles. A dialogue is treated as an activity that requires knowledge, experience, ability and training. We will also propose a specification language for the representation of the linguistic knowledge involved, as well as a task-oriented inference engine for the control and reasoning about the dialogue. This generic model has been implemented in Delfos, an application developed for the automatic telephone task scenario.

## 1. Introduction

Our previous research in collaboration with the Speech Technology Division of Telefónica I+D, has focussed on the study of the automatic telephone operator scenario (section 2) within the ATOS project (Alvarez *et al* 97; Fernández & Quesada 99; López & Quesada 99). As a result of this collaboration, we have elaborated a generic methodology for the design and implementation of dialogue management systems. This paper describes the most relevant characteristics of our methodology, and their implementation in the prototype system Delfos.

The methodology we propose is inspired on Knowledge Engineering principles and distinguishes two major levels in the design of systems: a level for knowledge specification and a level for inference and control (section 3). Besides, our methodology divides the specification level into two further levels: the representation of speech acts (section 4) and the representation of dialogue structures (section 3).

## 2. Design Constraints: A Dialogue System for the Automatic Telephone Task Scenario

This section describes the design constraints of the system. We illustrate a sample conversation (Figure 1) in order to present the functionality we are aiming at (although our corpus is in Spanish, we have translated the conversation into English for expository reasons).

This short conversation will help us to describe the design restrictions we have taken into account:

- **Interaction with the speech recognition system**. Our system is embedded in a Spoken Dialogue System application which takes as input the output of a speech recognition system through the telephone line. Speech recognition errors such as those reported in our sample conversation have been dealt with in previous research work in our group. The natural language processing system Iris (López & Quesada 98) incorporated a number of techniques for the detection and correction of recognition errors during the natural language understanding phase. Nevertheless, the dialogue management system should also be capable of handling recognition errors when configuring the dialogue interaction by using both direct confirmation questions (as in S6) and indirect ones (S7).

- **Task Detection**. Our scenario difers from Task-oriented systems in that the system does not know beforehand the task that the user has in mind. Rather, the user may choose between any of the different functions which have been designed to interact with the PABX. Therefore, the first problem that the system must solve is to figure out which task(s) the user may want to perform.

- **Incomplete functions**. In our scenario, it is common to find situations in which the requested functions are not complete, that is, commands for which the user has not specified all the arguments required to fulfil the task. For instance, in U3 the user is requesting the system to transfer his/her calls, but s/he has not specified the destination number. In this case, the system must be capable of keeping track of the requested information and generating those questions necessary to complete the missing information.

- **Expectations**. The dialogue manager system can benefit from the knowledge of the previous history of the dialogue. In fact, this dialogue history generates answering expectations: in S4 the system is waiting for a confirmation answer either affirmative (“Yes”, “OK”, ...) or negative (“No”, “That’s wrong!”, ...), in S5 the system is expecting a telephone number from the user. This knowledge (expectations generated from the dialogue history) can be further used both by the speech recognition system and the natural language processing module in order to improve the efficiency and restrict the semantic search in the grammar.

- **Multiple paths in a dialogue**. As we can observe in U3, it is common to find cases where one interaction initiates several functions. The system must be capable of handling all of them, while recalling unfulfilled work at every single point. In our example S4 is asking for direct confirmation of one of the functions. Later on, in S5 the system carries on with the fulfilment of

S1: [Welcome message and management instructions]
U1: I would like to know if I have any message in my mailbox.
S2: You have a new message.
U2: Please, I would like to hear it.
S3: The message is: ...
U3: Now I want to delete it and to transfer my calls.
S4: Shall I delete the last saved message?
U4: Yes, please.
S5: The last message has been deleted. To transfer your calls,
I need a telephone number or an extension.
U5: The number is 123456789
S6: So, do you want me to transfer your calls to the number 132456789?
U6: No, to the number 123456789.
S7: OK, your calls have been transferred to the number 123456789.
Would you like to perform any other function?
U7: No
S8: [Goodbye message]

Figure 1: A sample conversation

the function requested in U3 while indirectly confirming the execution of the first function.

- **Dialogue history and anaphoric references**. Representing the previous dialogue history is also useful in order to deal with discourse phenomena such as anaphoric references. In our sample conversation both answers U2 ("I want to hear it") and U3 ("delete it") refer to "the last message".

Taking into account the design restrictions imposed by the application described, we have developed a general methodology which is described in the next section.

## 3. A Knowledge Engineering-based Methodology for the Representation and Reasoning in Spoken Dialogue Systems

The level of representation (Hodgson 91) plays a crucial role in the study of the discourse level of language and in the implementation of dialogue management systems. This is evident, for example, in Discourse Representation Theory (Kamp 81; Kamp & Reyle 93; Eijck & Kamp 97). One of the main areas of research has been the development of dynamic representation models of speech acts which are capable of allowing an incremental interpretation of the utterances within a context (Kamp 81; Barwise & Perry 83; Cooper *et al* 99).

On the other hand, those research works which have studied the implementation of dialogue management systems have made evident the need to specify and represent concrete models or dialogue plans in relation to the application domain tasks. Most implementations in the literature (OVIS (Noord *et al* 98), VERBMOBIL (Alexandersson *et al* 98), Philips (Aust *et al* 95), TRAINS (Trains Web Page), TRINDI (Trindi Web Page)) make use of a frame-based representation model and a dialogue management plan-oriented strategy (Schank & Abelson 77).

Thus, the integration of a dialogue management or planning strategy introduces a higher level of control on top of the speech acts, that gives rise to the idea of a dialogue structure representation model.

Bearing in mind the research work described above, as well as the design constraints explained in section 2, we propose a methodology which is based on the following principles (Figure 2 illustrates the architecture of the system Delfos according to these ideas):

- **Unification-based dialogue management**: basically, every speech act obtained from the module (Iris) is represented as a complex feature structure (Rozenberg & Salomaa 97). This facilitates the integration between the dialogue manager module and unification-based grammars (Shieber 86; Kirchner 90).

- **Communication via the CTAC protocol**: This protocol, formally based on the extended Lexical Object Theory (Quesada 98), guarantees an efficient, bidirectional, flexible and transparent communication between the NLP and dialogue management modules (López & Quesada 98).

- **Declarative Specification of the Dialogue Structures**: Delfos incorporates its own language for the specification of dialogue structures. This language profits from plan-based management techniques, thus avoiding those problems associated with dialogue grammars (Aust & Oerder 95). The specification language of Delfos allows, among other things, for the representation of the history of the dialogue, the control of expectations and the treatment of ambiguity.

- **Dialogue Management as an Inference Engine on a Discourse Declarative Model**: The reasoning level can be regarded of as an inference engine (Hodgson 91) specialised on the treatment of dialogue manager systems.

## 4. Utterance Representation

From a functional perspective, the system receives as input the results obtained from the speech recognition sytem,

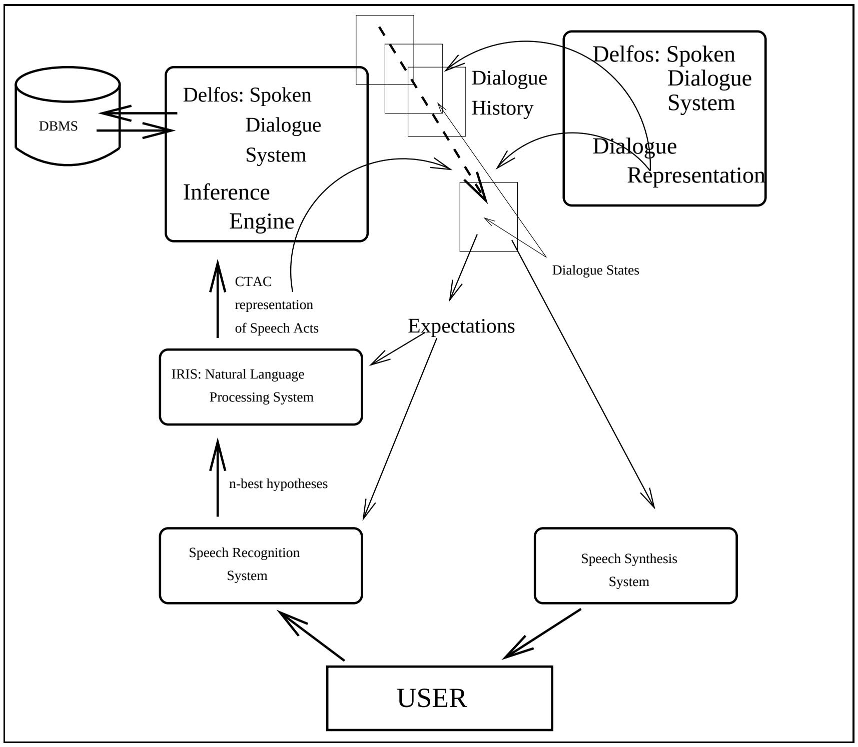


Figure 2: The Delfos Architecture

represented as a set of n-best hypotheses. The NLP system Iris (López & Quesada 98) then carries out the lexical, morphological, grammatical and semantic analysis. It also incorporates a wide range of speech repair strategies (López & Quesada 99).

The NLP system transforms each utterance into a list (one or more) of feature structures according to the CTAC protocol (López & Quesada 98). The CTAC protocol stands for Class, Type, Arg and Contents.

1. **CLASS**: Three classes have been defined for this application: Object, Function and Operation. Object includes any terminal element in the dialogue structure such as Person, Telephone Number, etc. Function describes the set of tasks in the application domain. Finally, Operation will be used for the representation of auxiliary functions such as Confirmation, Cancellation, Help, etc.

2. **TYPE**: This feature specifies the value that each class adopts in a given structure, as described in the **CLASS** section.

3. **ARG**: Some classes may require the presence of one or more arguments. The ARG feature specifies the argument structure of the class. This takes the form of a list in which conjunction, disjunction, and optional operators may appear.

4. **CONTENTS**: This feature represents the particular values associated with each element of the ARG attribute.

As an example, the second part of the interaction U3 above *"transfer my calls"* will be represented as:

$$\begin{bmatrix} CLASS & : & Function \\ TYPE & : & TransferCalls \\ ARG & : & [Name]|[Number] \\ CONTENT & : & \end{bmatrix}$$

For the corpus we are currently using, which contains 60 pieces of dialogues (amounting to more than 1,000 sentences) extracted by a Wizard of Oz, we correctly identified the task and assigned the correct CTAC representation in a 96% of the cases.

## 5. Dialogue Structures Representation

As mentioned above, the module in charge of the dialogue management is called Delfos. This subsystem incorporates its own language for the specification of dialogue states. These states are triggered by the speech acts resulting from CTAC. In turn, dialogue states may trigger expectations, modify the history of the dialogue, and/or execute actions.

Continuing with the example above, the specification of the dialogue state corresponding to the function *Transfer calls* in Delfos is represented in Figure 3.

Though a complete description of this specification language is beyond the scope of this paper, we will briefly describe some of its components. First, the dialogue state is triggered by a process of unification between the CTAC and the `TriggeringConditions` in the state specification. The `DeclareExpectations` field assigns priority values to other states, and specifies how the result of those states will be incorporated into the history of the dialogue. Thus, the CTAC generated for the interaction U5

```
( StateID:        TRANSFERCALL;
  PriorityLevel:  15;
  TriggeringCondition:
          (CLAS:Function,TYPE:TransferCall);
  DeclareExpectations: {
          Name <= NAME;
          Number <= NUMBER;
  }
  SetExpectations: {
          Confirm <= (CLAS:YesNoEnd,TYPE:Confirm,CONT:Yes);
  }
  ExitState:
          (CLAS:YesNoEnd,TYPE:Confirm,CONT:Yes);
  ActionsExpectations: {
          [Name] => {
             UserPrompt("Por favor, indique el destino del desvo."); }
          [Confirm] => {
             @if (@is-TRANSFERCALL.Name) @then {
                     UserPrompt(@concat("Realmente quiere desviar a ",
                          @is-TRANSFERCALL.Name.CONT,
                          " ?")); }
             @else {
                     UserPrompt(\@concat("Realmente quiere desviar al ",
                         @is-TRANSFERCALL.Number.CONT,
                          " ?")); }
          }
  }
  PostActions: {
          UserPrompt("EJECUCION DEL DESVIO");
  }
)
```

Figure 3: The TRANSFERCALL Dialogue State Specification

(*"The number is 123456789"*, which was understood as *"The number is 132456789"*):

$$\begin{bmatrix} CLASS & : & Object \\ TYPE & : & Number \\ ARG & : & [] \\ CONTENT & : & 132456789 \end{bmatrix}$$

will trigger the NUMBER (a king of destination for a telephone call) dialogue state (Figure 4).

The information obtained by this state will be integrated in the previous dialogue state using the expectation specification `Number <= NUMBER`, thus yielding the representation shown in Figure 5.

With this representation, the system can trigger the direct confirmation question S6.

## 6. Conclusion

This paper has described a methodology for the design and implementation of Spoken Dialogue Systems which integrates different techniques from Language and Knowledge Engineering. As a result, we have presented an architecture that divides the Dialogue management level into two main components. Firstly, the representation module which distinguishes the representation of speech acts according to the CTAC protocol, and, second, the representation of dialogue states. The system proposed is capable of dealing with all the design constraints previously specified for the task, such as bi-directional interaction with a speech recognition system, control of incomplete functions, manipulation of expectations and multiple paths in a dialogue, and representation of the dialogue history.

## 7. References


Alexandersson, J., B. Buschbeck-Wolf, T. Fujinami, M. Kipp, S. Koch, E. Maier, N. Reithinger, B. Schmitz & M. Siegel. 1998. *Dialogue Acts in VERBMOBIL-2.* DFKI Saarbrcken and TU Berlin, Report 226, July 1998.

Alvarez, J., Tapias, D., Crespo, C., Cortázar, I. and Martínez, F. 1997. Development and Evaluation of the ATOS Spontaneous Speech Conversational System. *International Conference on Acoustics, Speech and Signal Processing*, 1139-1142

Aust, H. & M. Oerder. 1995. Dialogue Control in Automatic Inquiry Systems. In Andernach, J. A., S. P. van der Burgt & G. F. van der Hoeven. eds. *Proceedings of the 9th Twente Workshop on Language Technology*. University of Twente, Netherlands.

Aust, H., M. Oerder, F. Seide & V. Steinbiss. 1995. The Philips automatic train timetable information system. *Speech Communication*, 17, 249-262.

```
( StateID:        NUMBER;
  PriorityLevel:  20;
  TriggeringCondition:
                  (CLAS:Object,TYPE:Number);
)
```

Figure 4: The NUMBER Dialogue State Specification

$$\begin{bmatrix} CLASS & : & Function \\ TYPE & : & TransferCalls \\ ARG & : & [Number] \\ CONTENT & : & \\ Number & : & \begin{bmatrix} CLASS & : & Object \\ TYPE & : & Number \\ ARG & : & [] \\ CONTENT & : & 132456789 \end{bmatrix} \end{bmatrix}$$

Figure 5: An unification-based approach to the incremental representation of information states


Barwise, J. & Perry, J. 1983. *Situations and attitudes*. Cambridge, Mass: The MIT Press.

Cooper, R., S. Larsson, C. Matheson, M. Poesio & D. Traum. 1999. *Coding Instructional Dialogue for Information States*. TRINDI (LE4-8314) Deliverable D1.1, February 1999.

Eijck, J. van & Kamp, H.. 1997. Representing Discourse in Context. In van Benthem, J. & Meulen, A. ter. eds. *Handbook of Logic and Language*. Elsevier Science. pp. 179-237.

Fernández, G. & Quesada, J. F. 1999. Delfos: Un Modelo Basado en Unificación para la Representación y el Razonamiento en Sistemas de Gestión de Diálogo. *Procesamiento del Lenguaje Natural*, 67–74.

Hodgson, J.P.E. 1991. *Knowledge Representation and Language in AI*. Chichester, England: Ellis Horwood.

Kamp, H. 1981. A theory of truth and discourse representation. In Groenendijk, J., Jansen, T. & Stockhof, M. eds. *Formal methods in the study of language*. Amsterdam: Mathematical Centre tracts 135.

Kamp, H. & Reyle, U. 1993. *From Discourse to Logic*. Dordrecht: Kluwer.

Kirchner, C. ed. 1990. *Unification*. San Diego, California: Academic Press Inc.

López, M. T. & Quesada, J. F.. 1998. Spoken Language Parsing Strategies in a Conversational System. In *Proceedings of ECAI'98: XIII European Conference on Artificial Intelligence*, 203-204.

López, M. T. & Quesada, J. F.. 1999. Error Detection and Error Recovery from Speech Recognition: Language Engineering Strategies. *XIV International Congress of Phonetic Sciences.* San Francisco, CA.

Noord, G. van, G. Bourna, R. Koeling & M. J. Nederhof. 1998. Robust Grammatical Analysis for Spoken Dialogue Systems. *Natural Language Engineering*, 1(1), 1-48.

Quesada, J. F. 1998. The Lexical Object Theory: Specification Level. *Grammars*, 1, 57-84.

Rozenberg, G. & A. Salomaa. eds. 1997. *The Handbook of Formal Languages*. Berlin: Springer Verlag.

Schank, R. and R. Abelson. 1977. *Scripts, Plans, Goals and Understanding*. Hillsdale, New Jersey: Lawrence Erlbaum.

Shieber, S.M. 1986. *An Introduction to Unification-based Approaches to Grammar*. CSLI Lecture Notes 4. Stanford, California: Center for the Study of Language and Information.

(Trains Web Page) University of Rochester, Department of Computing Science. *The TRAINS Project: Natural Spoken Dialogue and Interactive Planning*. http://ftp.cs.rochester.edu/u/trains

(Trindi Web Page) TRINDI: Task Oriented Instructional Dialogue. http://www.ling.gu.se/research/projects/trindi