COLING 82, J. Horecký (ed.)
North-Holland Publishing Company

A FORMALISM FOR THE STRUCTURAL ANALYSIS OF DIALOGUES

Helder Coelho
Centro de Informática
Laboratório Nacional de Engenharia Civil
101, Av. do Brasil, 1799 Lisboa Codex, Portugal

This paper outlines a formalism for conversational analysis that captures multiple interactional patterns of mixed-initiative and fix the historical skeleton of a dialogue. The formalism, composed by a grammar of dialogues (syntax and semantic component) and a collection of attached scenarios (pragmatic component), is written in Prolog and implemented in a program which converses in Portuguese to provide a library service.

INTRODUCTION

In trying to solve the problem of building programs that interact in natural language no great attention has been paid to formalisms for the structural analysis of dialogues. However, such formalisms may support a clear description of the history of the whole conversation, and convey reasoning power when associated to other tools such as definite clause grammars DCG's [2] , an extension of context-free grammars. And, on the other hand, they promote the elaboration of theories of systematic comprehension of conversations and, in fact, a better understanding of conversations between individuals, because they allow their simulation.

Dialogue taking place, for example, in a library world has structural properties, and rules are derived to form a grammar of dialogues [4] . This grammar, expressed by a DCG, generates all possible exchange forms and is responsible for the organization of the possible interactions in such context, because it contains a description of the various ways in which the dialogue units may be strung together to form dialogues. In fact, the majority of those rules are general and may apply to other task domains. Rules are defined in terms of semantic concepts, like requests or answers, which are supported by sentences of natural language. And, these sentences are analysed, ie, translated into logical structures, by a DCG for Portuguese [1] . Both DCG's are expressed by Prolog's grammar rules [3]

The research behind this paper is guided by the objective of design and implementation of computer programs capable to display intelligent behaviour according to the standards of human perfomance. An associated objective, the comprenhension of human perfomance with the help of computer models leading to theories about intelligent human behaviour, is kept in mind but not discussed along the paper.

A FORMALISM FOR THE ANALYSIS OF DIALOGUE

The organization of taking turns to talk is fundamental to dialogue, as well as to a program able to converse. A formalism for the analysis of dialogue is proposed, and we examine its compatibility, with the representation of the history of dialogues between a program and its users.

Let P be a set of participants and C a set of contributions. By a contribution act we mean a member of the set P×C of participant-contribution pairs. For exemple, <p1,c11> is a contribution act, where p1 and c11 are the first members of P and C, respectively.

Let S be a set of conversational states or configurations. A conversational state s is a sequence of at least two related contribution acts. For exemple, c11 stands for the first contribution regarding the first conversational state.

By a dialogue of length n we mean a member of the set $(P \times C)^n$ of sequences of n contribution acts; and by a dialogue we mean a member of the set

$$T = \bigcup_n (PxC)^n \qquad (n \in N)$$

of dialogues of any length. Each member of a dialogue is of the form

$$\langle s, \langle p,c \rangle \rangle \qquad (s \in S, p \in P, c \in C)$$

wich we identify with the triple $\langle p,s,c \rangle$.
For example, $T = \{\langle p1,1,c11 \rangle , \langle p2,1,c21 \rangle , \langle p1,2,c32 \rangle , \langle p2,2,c42 \rangle \}$
is a dialogue of length 4, with 2 participants, p1 and p2, 2 conversational states and 4 contributions.

We call E = PxSxC the set of events, and any triple $\langle p,s,c \rangle$ an event. A dialogue is a sequence of events, grouped into units, and governed by rules. The conversational units are the invariant structures of dialogue: sub-dialogues, exchanges, monologues and contributions. A sub-dialogue (dialogue course or segment) is any sequential subset of a dialogue,

$$\text{Course} = \{ x:(\exists\, t \in T)\ (x \cap t=0 \text{ and } x \cup t \in T)\}$$

An exchange is a set of two consecutive events, concerning the same conversational state and two different participants. A pre-defined exchange, conducted by the program, is called an exchange pattern. A monologue is a sequence of at least two consecutive events, concerning the same conversational state and the same participant.

$$\text{Mon} = \bigcup_p \bigcup_n (\{p\}xc)^n$$

A contribution of a participant to a dialogue is a sequence of his contributions. The semantics of contributions covers the following types: requests (statements, questions, and commands), answers and remarks (eg. agreement).

The underlying structures of the situations occurring in a certain problem world determine the organization of dialogue and its systems. A grammar of dialogue is a set of rules of dialogue. Rules of dialogue state how participants understand coherent dialogues, and specify the membership of the set of legal dialogues K, such that $K \subset T$, where T is the set of dialogues of any length. Rules of dialogue define the class of coherent dialogue and their attached models. They contain the way contributions are put together. A model is a system of dialogue defined as the triple $\langle P,C,K \rangle$. The core of any model is the set R of rules defining K.

ANALYSIS OF DIALOGUE

We consider the dialogue occuring between the program TUGA [1] and its users in the library world. Such dialogues are evolving dual processes, goal-and-rule-oriented for sharing information between the participants. They are dual because there are only two participantes at a given time. They are goal oriented because they are carried on to satisfy, for example, the following objectives:

i) to satisfy users straightforward request concerning the document collection and the classification system,
ii) to ask users about the library world (eg. the author of a paper), for conversational guidance purpose, and
iii) to present the user with proposed data (eg, the document classification), enabling him to choose from or modify it.

They are rule-oriented because the conversational units are governed by a grammar of dialogueswhich determines the roles played by the program--the librarian and the library's secretary, and by the user. The grammar also allows the inclusion of several courses inside a dialogue, such as topic shifts for simple question-

-answering purposes.

Consider a dialogue between the program and one of its users, defined as a sequential organization of single exchanges. The conversational description is represented in the diagram of the following figure.

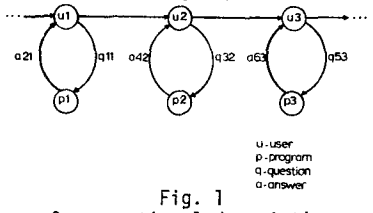

Fig. 1
Conversational description

The diagram shows the history of that dialogue. The interaction proceeds in separate discrete steps from one to another of a finite number of configurations. There are three sucessive conversational states through time, each one corresponding to a single exchange, question (q) - answer (a) pair, conducted under user initiative. This history is described as follows.

$$T = \{ \langle u,1,q11\rangle , \langle p,1,a21\rangle , \langle u,2,q32\rangle , \langle p,2,a42\rangle , \langle u,3q53\rangle , \langle p,3,a63\rangle \}$$

is a dialogue of length 6, with 2 participants, the user (u) and the program (p), with 3 conversational states and 6 contributions (questions and answers). Note that 'q32', for example, means that the third contribution is a question, referring to the second conversational state.

Consider now the following segment of a nested dialogue, concerning the classification of a document:

(1)	u-Classifique "Explorations in Cognition"!	(Classify "Explorations in Cognition"!)
(2)	p-"explorations in cognition" é desconhecido!	("explorations in cognition is unknown!
(3)	Quer ainda classificá-lo?	Do you still want to classify it?)
(4)	u-Está bem.	(All right.)
(5)	p-Vamos classificá-lo!	(Let us classify it!)
(6)	Por favor, de-me o titulo de uma referência do documento em questão.	Please give me the title of a reference from the document in question.)
(7)	u-Qual é o título do Nilsson?	(What is the title of Nilsson?)
(8)	p-"artifical intelligence"	("artificial intelligence".
(9)	Vamos classificá-lo!	Let us classify it!
(10)	Por favor, de-me o titulo de outra referência do documento em questão.	Please, give me the title of another reference from the document in question.)
(11)	u-Não tenho nenhuma.	(I have none.)
(12)	p-Está bem!	(All right!
(13)	Já que não consegui obter nenhuma referência, não sei dar-lhe qualquer sugestão! Quer ainda classificá-lo?	As I could not get any reference, I don't known to give you any suggestion! Do you still want to classify it?)
(14)	u-Não quero.	(No, I don't.)
(15)	p-Está bem! E que mais?	(All right! And what else?)

The conversational description, represented in the figure 2, shows the organization and levels of the dialogue. Numbers, placed on the left of contributions, are used for referencing them to the segment considered.

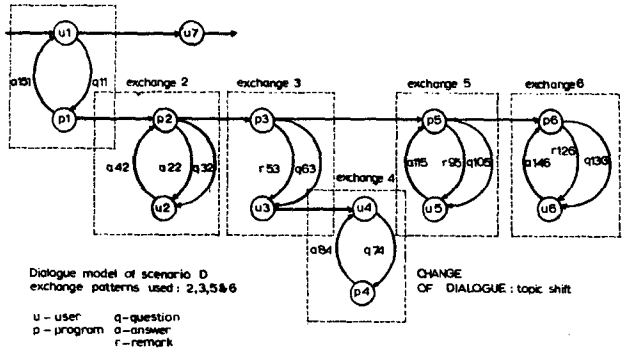

Fig. 2
Conversational description

The history of this dialogue is described as follows,

T={⟨u,1,q11⟩,⟨p,2,a22⟩,⟨p,2,q32⟩,⟨u,2,a42⟩,⟨p,3,r53⟩,⟨p,3,q63⟩,⟨u,4,q74⟩,
⟨p,4,q84⟩,⟨p,5,r95⟩,⟨p,5,q105⟩,⟨u,5,q115⟩,⟨p,6,r126⟩,⟨p,6,a136⟩,⟨u,6,a146⟩,
⟨p,1,a151⟩}

is, a dialogue of length 15, with 2 participants, the user (u) and the program (p), 6 conversational states and 15 contributions (questions (q), answers (a) and remarks (r).

GRAMMAR OF DIALOGUES

The grammar of dialogues of program TUGA is a complete and precise description of the properties of a certain class of dialogues. The properties concern the structures of the dialogues, occurring in a library world, and organized as models. This grammar machinery is able to parse situations, and it is very much like the one able to parse a natural language sentence: the objects recognized, dialogue units, are characterized as structured objects assembled out of recognizable parts according to know rules of assembly.

A dialogue carried out by TUGA has two participants, the program (p) and its users (u), and therefore two mutually exclusive states, the "agent" and the "passive participante". Both participants may take the initiative during the encounter, ie. the program may be an "agent" or a "passive participant". The "agent" claims the turn to speak at any given moment, and plays an active role. The "passive participant" does not claim the turn to speak at any given moment.

Considering two states for each participant, there are four possible conversational states. However, we only consider two states: "agent"— "passive participant" and "passive participant" — "agent". (The other two states represent in some sense a failure of dialogue.)

The BNF specification of the grammar syntax above the discourse level, presented below, characterizes only the class of dialogues considered.

⟨converse⟩ → ⟨opening1⟩ , ⟨converse⟩
⟨converse⟩ → ⟨opening2⟩ , ⟨converse1⟩
⟨converse1⟩ → ⟨converse2⟩, ⟨continue⟩
⟨converse2⟩ → ⟨dialogue⟩
⟨converse2⟩ → ⟨monologue⟩

‹converse3›	→	‹dialogue1›
‹converse4›	→	‹decide›, ‹dialogue›
‹converse5›	→	‹dialogue2› , ‹course›
‹converse6›	→	‹ask-author›, ‹ask-publisher›, ‹ask-date-of-publication› , ‹ask-document-type›
‹converse7›	→	‹converse2› , ‹dialogue›, ‹converse2›
‹continue›	→	‹conversel›
‹continue›	→	‹suspend› , ‹converse›
‹continue›	→	‹close›
‹dialogue›	→	‹user› , ‹program›
‹dialogue1›	→	‹p-question1› , ‹dialogue›
‹dialogue2›	→	‹p-question2› , ‹dialogue›
‹dialogue3›	→	‹p-question3› , ‹dialogue›
‹course›	→	‹converse4›
‹course›	→	‹refusal› , ‹converse4›
‹course›	→	‹change›
‹course›	→	‹dialogue3› , ‹refusal›
‹course›	→	‹return› , ‹converse3›
‹user›	→	‹question›
‹user›	→	‹fact›
‹user›	→	‹command
‹user›	→	‹answer›
‹program›	→	‹response›
‹program›	→	‹response› , ‹converse3›
‹program›	→	‹response› , ‹converse5›
‹program›	→	‹response› , ‹converse6›
‹program›	→	‹response› , ‹converse7›

Let us consider only the first few rules in order to make explicit their meaning. A general dialogue, 'converse', is defined as an opening following by a sub-dialogue which may be followed by a sub-dialogue or closed by user initiative. The user may also suspend temporarily the dialogue without affecting it. This feature justifies the existence of two kinds of opening: one for the dialogue start and the other for the re-start. A dialogue is simply a sequence of exchanges or monologue, or is followed by several models of dialogue. For example, dialogue on the classification of a document is handled by dialogue model 'converse5', which is defined by rules 'course'. These rules define several kinds of possible courses during the interaction. Dialogue on adding new documents is handled by dialogue model 'converse6', which is served by a set of exchange patterns -- a sequence of pre-defined program questions whose order may be altered by user. Some feature of a sentence, together with the current context, can trigger a hypothesis that an instance of some particular model or pattern is being conveyed. All this means that the program can cope with user changes of mind, single or multiple data, provided in any order, and can avoid asking questions whose answers were provided either implicitly or explicitly at some earlier time.

We use rules of interpretation, below the discourse level. The rules of interpretation deal with what the user does, eg, requests (statements, questions and commands) and answers. Other rules deal with what the program does, eg, answers, questions, and remarks (comments and agreements). Here are, for example, three of these rules:

Rule -- If the user makes a statement, the program interprets it as a request for confirmation.
Rule -- If the user asks a closed question (form Q-S, where S means the statement corresponding to the question) and the program responds with an existential E (yes/no), then the program is understood as answering the user with the tratement E-S.
Rule -- If the user issues a command, then the program interprets it as a valid request for an action A only if the following conditions hold:

the request is ended with an exclamation mark, and action A is classifying a document, generating a category, adding data items and, deleting data items.

The first two actions also cover the general purpose of gathering information through a referent: the referenced document or the classification.

DIALOGUE ORGANIZATION

STRUCTURES AND LEVELS

Any original natural language sentence is parsed by a cascade of two DCG's. The first DCG, above the discourse level, represents the syntactic information about dialogue form (exchange patterns and dialogue models) and is closely coupled to a set of scenarios which represent the pragmatic information about the task domain (a collection of situation descriptions). The second DCG, on the sentence level, represents the syntatic information about sentence form and the general semantic information.

Let us observe in more detail the main dialogue forms, engineered and manipulated by the first DCG in order to build up the overall skeleton of the dialogue history. An exchange pattern (eg.'dialogue') is a pre-defined exchange between the program and the user. It is defined by a name and a number, and provided by a message and the number of the following contribution. It consists of a question of an expected form, followed by a simple dialogue. The question is constructed with the value of the message (eg. a proper noun). The simple dialogue is the standard way to interpret user contributions: the question-answer pair. As regards exchange patterns, the user contribution expected is not always an answer: the user also question the program, and by doing so a new dialogue is nested in the previous one. Program questions are motivated by the content of user request. For exemple, interrogating 'the name of a new category' and 'under what categories may it be placed' are generated when the user wants to create a new category in the classification system.

The exchange pattern is called by the grammar of dialogues through its name and number. In case of non acceptance of the program question by the user, the initiative for restarting the dialogue belongs to the user. But the new dialogue may be nested in the previous dialogue, as often occurs in the process of classifying new documents.

A dialogue model (eg. 'converse') is a suite of unconstrained exchanges between the program and the user. It generates detailed expectations about the next contribution, by having an ordering for calling exchange patterns which may be altered by user. The user may give several answers which need not be ordered. Also, he may modify his previous answers. The program uses the success or failure of is predictions to determine what role the user contribution plays in the dialogue. Whenever a dialogue model is activated, an appropriate exchange pattern is invoked,and the program poses a question to the user and interprets the user response. If a failure occurs, the program is able to come back to the same topic. For example, during the classification of a document, the user may oppose the program and request information about the location of a category in the classification system. And, the program only accepts three titles of documents, known to its data base, and according to its classification method. Therefore, it goes on asking the user till it attains that limit, and skips any unknown title. But if the user gives up, the program restarts a new dialogue. These features are implemented either by using recursion or backtracking. Recursion is the ability of a procedure to contain a procedure call to another copy of itself. The declaration of the procedure, for the dialogue model in charge of the classification process, contains a procedure call which matches the name of another copy of the same procedure declaration. Counters control the process of recursion during the program asking for references in the classification process. And, an handling device deals with contradictions arising when a new document is archived. Numbers (1 and 0) are assigned according to existing contradiction or otherwise, and summed over the facts. If the result is

non-zero, a failure aries forcing backtracking and the restarting of the process.

In the course of conversing, the program remembers the events by storing conversational states containing the name of the participant, the number of the contribution, and the active information, such as questions or answers. This knowledge about the current dialogue history and skeleton is maintained in order to be served in any specific situation or instance to which it applies. In fact, it is the support for the organizational ability of the grammar of dialogues .The example below illustrate this ability. Consider the segment of a nested dialogue, presented in figure 2, concerning the classification of a document.This dialogue presents a sub-dialogue and an interruption due to the user. The sub-dialogue corresponds to a dialogue model, which organizes interactions concerning the implementation of the document classification method. The interruption is presented as an exchange,nested in the sub-dialogue. This conversational description shows the organization, the depth of nesting, the change of initiative, and the levels of the dialogue (one for the ordinary tdalogue; the other for the particular dialogue model; and, the last one for the interruption).

Consider another example of a dialogue (numbers appended to the natural language contributions correspond to the conversational states).

(1)	u-Crie Prolog! p-Prolog é conhecido!	(Create Prolog!)(Prolog is known!)
(2)	u-Quero criar uma categoria!	(I want to generate a category!)
(3)	p-Por favor, qual o nome da nova categoria?	(Please, what is the name of the new category?)
	u-"Pattern recognition".	
	p-"Pattern recognition" é desonhecida!	("Pattern recognition" is unknown!
(4)	Por favor, debaixo de que categoria a pretende inserir?	Please, under what category do you want to insert it?)
	u-"Machine vision".	
	p-"Machine vision" é conhecida!	("Machine vision" is known!
(5)	A nova categoria ficou inserida no sistema de classificação e recebeu o número 2141. E que mais?	The new category has been inserted in the classification system and received 2141 as a number. And what else?)

The history of this dialogue is:

$$T=\{\langle u,1,c11\rangle,\langle p,2,a21\rangle,\langle u,2,c32\rangle,\langle p,3,q43\rangle,\langle u,3,a53\rangle,\langle p,4,r64\rangle,\langle p,4,q74\rangle,\langle u,4,a84\rangle,\langle p,2,a92\rangle\}$$

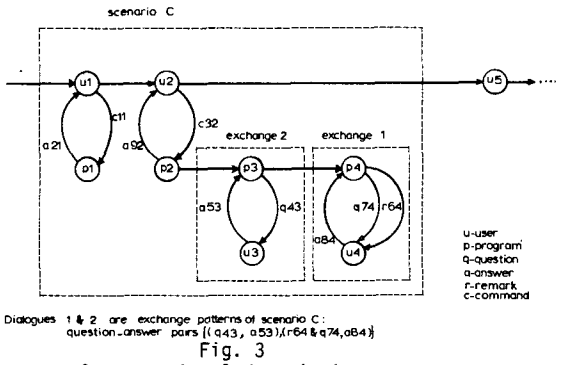

Fig. 3
Conversational description

The program knows that the first user contribution, a command,$\langle u,1,c11\rangle$, opens a dialogue composed of a simple question-answer pair (conversational state 1). The dialogue goes on with another user command, $\langle u,2,c32\rangle$, which invokes a dialogue

model for creating new categories. This dialogue model calls two exchange patterns, and the dialogue is closed with the program answer <p,2,a92> . Note that this last event has the same conversational state number (2) as the event invoking the dialogue model. This information, shared by the program and one of its users, helps the program to decide on what to do and how to proceed. It chooses its course of action by inspecting previous user decisions, through the remembering mechanism.

The dialogue model, responsible for gathering information about new documents, exemplifies the use of backtracking. If the user changes his mind at any stage of the dialogue, the program backtracks to follow up the consequences of the new information. Supplied facts contradicting those already known are detected in the immediate interpretation of the user's input, and when this interpretation is complete, a failure leading to the restart of interpretation at a previous level occurs if a contradiction has been found. This mechanism overrides the repetition of unanswered questions, and skips questions by recognizing the content and form of the user's answers.

SCENARIOS

Scenarios are sets of expectations and presumptions regarding a certain type of situation, They are of the form "if <situation description> then expect <situation description> ", or "if <situation description> is a satisfied then do <action description> ". Situation and action descriptions are triggered by verbs. Scenarios are used in TUGA [1] as recognition devices for classifying and identifying situations in a dialogue, and they call the exchange patterns and organize their invocation. All the embedded knowledge embedded in the scenarios covers the ability.

- to derive questions from relevant information or from the logical consequences of the information that is known about the questioned topic combined with general knowledge of the library world, and

- to handle the user's answers

TUGA is a program able to play two roles in the library world.
It acts as a librarian and as a library's secretary.

Possible events in the library world are grouped into the following scenarios:

Scenario A - information transaction
  Subscenario A1 - data output control
  Subscenario A2 - dictionary extensions

Scenario B - addition and/or deletion of data items
  Subscenario B1 - addition of new documents
  Subscenario B2 - addition of news categories
  Subscenario B3 - deletion of existing documents
  Subscenario B4 - deletion of classification categories

Scenario C - classification category generation
Scenario D - document classification

Scenarios A, B and C may occur inside scenario D. Scenarios B2 and B4 may occur inside scenario C.

Exchange patterns are classified according to their use in these scenarios, as shown in the following figure. The classification is made possible thrpugh two of their arguments: name and number.

This taxonomy for situation recognition is made available during a dialogue. It contains the pragmatic knowledge of the task domain, and supports the program ability to converse with users in a more clever way, when put aside the grammar of dialogues.

Scenario	Sub-scenario	Dialogue model	Exchange patterns
A	A1		asking whether the user wants more data
	A2	handling unaccepted sentences	asking the user's agreement for dictionary enlargement
			asking about a syntatic error in the sentence
			asking whether the unknown word is a proper noun
			asking which are the proper nouns
			asking about the unknown word gender
			asking about the unknown word domain
B	B1	addition of new documents	asking document author and publisher
			asking document date of publication and sort
			asking document categories
			asking the user to confirm his desire for document storage
	B2	classification system alteration	asking the category name under which the new one will be inserted
			asking the new category name
C			
D		classification method	asking the reference title
			asking the maximum of three categories
			asking the user to confirm his desire
			asking the user's choice for document categories
			asking the document classification

Fig. 4

Classification of exchange patterns

CONCLUSIONS

An intelligent automated conversationalist may attempt to construct a process of participation by analysing the other's process, ie. use conversational procedures, develop multiple conversations, fix the skeleton of the dialogue history, build up a conversational context space and appeal to it as a guider for the calculation of responses and disambiguation tactics. We are approaching the complexity of these phenomena by developing an evolving framework for conversational sequencing structure analysis, allowing users to insert sequences and abruptly shift to other topic, and the program to recognize user turns at conversing. However, our framework is not completed and some more work is needed to cover, for example, the implicit conversational mode or internal dialogue. This mode constitutes the background of the explicit flow of what is declared, tackled by the present work. Each participant, before uttering a sentence, discusses with himself to find the best appropriate utterance to match the opponent one. By doing this, the participant executes several mental constructions, like deductions, presupositions, analogies or associations, and compares the result to his beliefs in order to discover the intention of the other. A part of this extension, it would also be required to augment the complexity of the conversational context space by structuring it, because context plays the role of a model directing the reasoning of participants in what concerns the understanding of a sentence. Such improvement of embedding the function and the use of context into a program would also increase its interactional power. Finally, the pragmatic component needs further improvement in order to allow communication (passing messages) among scenarios.

References

[1] Coelho, H. "A program conversing in Portuguese providing a library service", PhD Thesis, Univ. of Edinburgh, 1979

[2] Pereira, F.C.N.;Warren, D.H.D. "Definite clause grammars with augmented transition networks", Dept. of AI Research report nº58, Univ. of Edinburgh, 1978.

[3] Pereira, L.M.; Pereira, F.C.N.; Warren, D.H.D. "User's guide to DECsystem-10 Prolog", LNEC, 1978.

[4] Sacks,H.; Schegloff, E.; Jefferson, G. "A simple systematics for the organization of turn-taking for conversation", Language Vol. 50 nº 4, 1974.