# User's Multiple Goals in Spoken Dialogue

*Didier Pernel**

Thomson-CSF
Laboratoire Central de Recherches
91404 Orsay cedex - FRANCE
*email: pernel@thomson-lcr.fr*

## ABSTRACT

This paper deals with a problematic not deeply studied as yet: user's goals interaction. A situation of multiple goals occurs as soon as the user utters a new goal whereas the previous one has not been solved yet. We propose an algorithm to identify the kind of multiple goals according to the task state and to the goals themselves. We define ten strategies to process those situations. Three meta-strategies order the strategies relevant for given situations. The system checks the preconditions of strategies to be sure they can be triggered. When a strategy is applied, the system updates the dialogue history and the task state. Some strategies push a goal in a stack and pop it when the first processed goal is fully reached.

## 1. Introduction

Dialogue management systems are now used in vocal servers [1] [2] [3]. However, none of them, is able to manage more than one goal at a time, whereas several corpora collected in application contexts show that users often make several simultaneous requests [4].

We define a situation of *multiple goals* as occurring when the user of a system utters a new goal before the full resolution of the previous one. The emergence of such a second goal changes the dialogue state, and may result in a different system reaction.

Figure 1 gives the general process to manage user's multiple goals and we propose in this paper several strategies to manage dialogue states involving them. These strategies are based on the types of goals and on their relations. We only studied goals occuring in database querying. The computer is the master which holds knowledge and the user does not really contest the system answers. He can only select items in database or scan them through precision requests.

Up to now, our system is implemented to manage two goals, giving that as more complex cases may come down to this one. Our solving strategies are based on a typology of goals and are triggered through preconditions.

| Identification of situation | → | Inventory of possible strategies | → | Choice of the strategy (metaplaning) |
|---|---|---|---|---|

Figure 1: Processing of multiple goals

After presenting related works, we detail every stage of the general process of management of multiple goals. We give some information about our implementation and supply a dialogue example before concluding.

## 2. Related work

Few works have been interested in multiple goals; moreover, they never have been addressed with this problematic specifically. MOLGEN, Stefic's system [5], can solve several goals by integrating them in predefined plans. Those plans belong to three levels: the domain level, the conception level and the strategic one. The general strategy is to decrease the goals differences. If no plan can be applied, already waiting plans are then checked.

Allen and Litman [6] define meta-plans which combine system plans: change, insertion, juxtaposition, clarification, correction, etc. Van Beek and Cohen [7] show that it is not always useful to choose among several plans when a more general one can factorize them.

[8] is one of the most full research on this topic, in which Wilensky describes several strategies to determine which goals should be solved first. In that work, he directly processes goals, and not plans, which is a more efficient approach: it is actually hard and time-consuming to retrieve the real user's plan. He describes every dialogue state where goals may interact; then he supplies strategies to solve the cases of interaction: preconditions allow the system to determine the best strategy according to the current state.

However, his approach cannot be extended to spoken dialogue systems, because many strategies involve everyday-life actions instead of speech acts only. The natural language output is sequential which prevents

*This work was partly realised at LIMSI/CNRS, France.

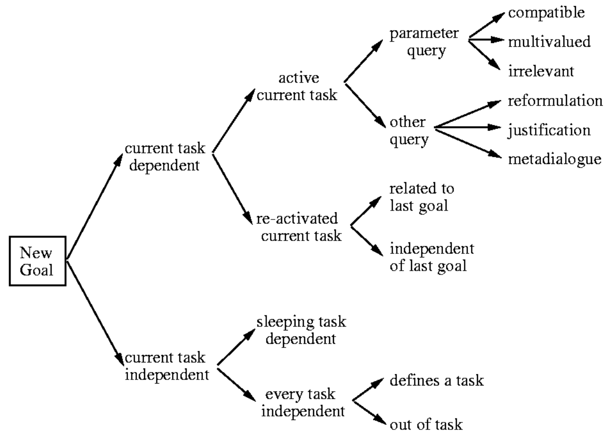


Figure 2: Taxonomy of multiple goals

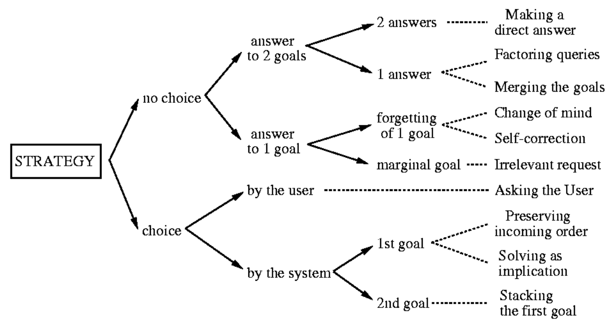


Figure 3: Taxonomy of strategies

an actual parallel goals processing. Another difference is that we assume that partners always have a well cooperative behaviour in order to solve the user's problem.

## 3. Handling a new goal

The system can identify 6 main types of user's goal: *DB-Request*, that requests to the application database, *Precision-Ask* about a previous system answer, *Answer* to a system question, *Give-Information* about a parameter value, justification or explanation[1], *System-Command* to do an action and *Unknown Type* when nothing else is identified.

The taxonomy of multiple goals situations (see figure 2) is based on the state of the current task (i.e. the task related to the first goal), and on the link between the second goal and the current task. Four task states are defined in the task model: (1) the task is *active* when it is currently under process; (2) it is *solved* when the initial goal of the user has received a first response, but the dialogue may continue on the same task (e.g. to be precised); (3) it is *sleeping* when it has not been solved yet, and out of the current focus of the dialogue; (4) a sleeping or a solved task may be *re-activated* when a new goal arises and is related to it.

A goal may be related to a task through the following links: precision, confirmation, prerequisite, reformulation, clarification and subgoal of the main goal.

When a new goal arises while another one is being solved, the system tries to relate it to a previous task. The first attempt is on the current task. If it fails, sleeping tasks are then tried in turn.

## 4. Strategies for solving multiple goals

The system has to show a general behaviour consisting in relevant and cooperative reactions. However, it is not able to process two goals simultaneously. Hence we define strategies aiming at determining a priority among the goals.

The strategies answer to the following questions depending on the dialogue state: (a) is it necessary to make a choice between the two goals? if so, which one shall be processed first? (b) if it is not possible to satisfy both goals, should one of them be forgotten? (c) is it possible to find a unified answer satisfying both goals?

Each strategy is defined though a set of preconditions to be sure that it is relevant in the given situation. When a strategy is applied, it first updates the new dialogue state through the dialogue history and the state of each involved task. Then the process described by the strategy is run.

We define 10 strategies (see figure 3) to be applied in the possible dialogue states found in our applications [9]. They fall into three main groups, corresponding to the questions (a), (b) and (c). Firstly, some strategies attempt to give answers to the two goals by: *making a direct answer* (if one of the two goals may be solved immediately, and the other one processed as usual), *merging the goals* (if one of them is included in the other), or by *factoring queries* if the two goals have common parameters. Secondly, some strategies solve the conflict by deleting one of the goals in case of *change of mind*, *irrelevant request*, or *self-correction* from the user. Thirdly, some strategies choose one of the goals to be frozen, by *preserving incoming order*, *stacking the first goal*, *solving as implication* (when one of the two goals is a precondition to the other), or explicitly *asking the user*. The other goal is then solved when the selected goal resolution is fully finished.

We now detail the *solving as implication* strategy. It is triggered when the first goal is a prerequisite of the second one. The corpus [4] supplies such an example: "Do you know the 'Ring' motorway? If yes, is there a gas station located on this road?" The system solves the first goal; if the answer is positive, it then solves the second one without any feedback to the first goal; if it is negative, it directly answers to the first goal without looking for the resolution of the second one.

[1] The system is aware of some frequent user's plans.

| Strategies \ Situations | Direct Answer | Factoring Queryies | Merging Goals | Change of Mind | Self-Correction | Irrelevant Request | Asking the User | Incoming Order | As Implication | Stacking |
|---|---|---|---|---|---|---|---|---|---|---|
| compatible | | | X | | | | | | | |
| multivalued | X | X | | X | | X | X | X | | |
| irrelevant | X | | | | | | | | | |
| reformulation | | | X | | | | | | | |
| justification | | | | | | X | | | | |
| metadialogue | X | | | | | X | | | | |
| related to last goal | X | | X | | | | | | X | |
| independant of last goal | X | X | | | | | | X | | |
| dependant sleeping task | X | X | | | X | | | X | | X |
| new task | X | X | | | | X | X | X | | |
| out of task | X | | | | | | | | | X |

Table 1: Possible strategies according to situations

## 5. Choice of strategy

A strategy can solve several situations of multiple goals and, *vice-versa*, a situation can be processed by several strategies. Static links supply the set of possible strategies being able to be triggered according to the situation of multiple goals. The table 1 points out those links. The dialogue state, the semantics of goals and their types allow to compute dynamically which strategies can really be used in the context.

If, after checking, several strategies are selected, the system bases its final choice on 3 ordered meta-strategies to select the most relevant one.

- use the strategies which make no choice (in order to be sure not to disappoint the user),
- use the strategies which solve the two goals at the same time,
- use the strategies which give a final answer rather than only begin the solving process.

A triggering strategy implies several modifications on the dialogue context: it determines the goal to process, the freezing of the other goal, the task management and it updates the dialogue state.

The dialogue history contains every relevant information: the choosen strategy, the state of each goal, the updated value of the task state and the new dialogue structure. When the system gives an answer to one goal, it must remind that another goal is waiting. The system can then react to the new active goal.

## 6. Design

The goals interaction module was first developed in Prolog at Limsi/CNRS in an Esprit project: Pragmatics-based Language Understanding System (P5254). It has been re-implemented in Lisp on a SGI Indigo2 station in the COVEN project (ACTS program) [10]. This new application deals with a database querying system devoted to office furniture: a user holds a spoken dialogue with the system in order to arrange a meeting room. The speech recognition system is Nuance Communications's and the text-to-speech system is Lernout & Hauspie's.

The dialogue system consists in classical modules: a parser based on Lexicalized Tree Grammar [11] (derived from Tree Adjoining Grammar) (215 words) and a contextual interpretation which solves references [12] and identifies the speech act [13] based on user's intentions. A speech act (or several in case of multiple goals) is provided to the dialogue module which manages the situation, computes the system reaction and updates the dialogue history, the task state and the dialogue structure. The dialogue module consists in several submodules: the multiple goals manager, the task module which queries the database, and the response planner. Finally the natural language generator, based on canned sentences to be filled in, is triggered.

The whole process from the user's input to the system output takes less than 2 seconds.

## 7. Example

The following dialogue is a sample of the dialogue we are working on. $U_x$ introduces every User utterance and $S_x$ the System reaction.

$U_1$: I want to put a table near the door
$S_2$: Which door are you talking about?
$U_3$: Where are the restrooms?
$S_4$: On the right, at the back of the room.
$S_5$ Do you want to put the table near them.
$U_6$: No, place the table here. *The user points out a place.*
$U_7$: Hang up a white board in front of the table
$S_8$: Before going futher, I need more information about the table. What is its size?
$U_9$: I don't know
$U_{10}$: But we will be 8 at the meeting
$S_{11}$: 4 face to 4?
$U_{12}$: Yes
$S_{13}$: Where do you prefer to put the white board? Here or there?
$U_{14}$: On this side. *The user points out a wall.*
...

$U_3$ introduces a multiple goal since the task defined by $U_1$ has not been solved yet. The system cannot infer any relation between the 'restrooms' and the table. So it creates a ***new task*** situation as multiple goal. The 10 line of the table 1 shows five potential strategies. The meta-strategies are then triggered to select only one of strategies. The first meta-strategy keeps the two first strategies only: *direct-answer* and *factoring queries*. The second one does not provide anything for the selection; but the last one implies to try to trigger the first strategy. This strategy produces the system reaction $S_4$. $S_5$ shows that the system goes back to the processing of the first goal.

$U_7$ defines a new multiple goal: a *new task* situation. The system looks for triggering the strategies in the same order. But the first two ones cannot be triggered because there is neither a direct answer nor a possible factorization. The *irrelevant request* cannot be used because the two goals are legal goals for the system. Finally, the system triggers the *preserving incoming order* strategy because every precondition is true. So, the system freezes the second goal and tries to solve the first one: $S_8$ is produced.

The last situation of multiple goal is defined by $U_{10}$. The recognized situation is *compatible parameter* because to give the number of people is a way of defining the size of the table. There is only one strategy which can be applied: the values *unknow* and *for 8 people* are merged into the more precise one. The system knows that the assistance number is not sufficient to precise the table size; so the system reaction is $S_{11}$.

$U_{12}$ allows to the system to finish the task of the creation of a table. Then it marks the task as a *solved* one and *re-actives* the *waiting* ones. It only remains the task about the blackboard. The system requires more precision to also complete this one.

## 8. Evaluation and conclusion

We use as evaluation criterion the notion of natural and user-friendly interaction: would a human react like the system in the same situation? Presently we have not fully tested the system. First results show that the performance of the system reaches 70 %. Further analysis will aim at classifying the errors to determine which ones are really due to the multiple goals management.

The goals interaction management is a phenomenon poorly taken into account by dialogue systems in spite of its frequency. Our approach is based on task state. The system checks potential strategies and chooses one according to general meta-strategies. Such a module is generic as far as the domain is concerned; strategies and meta-strategies are domain independent. Now there remains a constraint: the module is devoted to a specific dialogue task, which is querying a database.

We intend to look into the assumption of generalisation to three goals and more. We meet from time to time such a situation especially through meta-dialogue and misunderstanding situations. When all of the goals are not produced by the user, those goals can be overlapped. Some new strategies should then be inserted to avoid deadlocks, in case of misunderstanding for example. This other direction should be also investigated.

## REFERENCES


[1] James F. Allen. Discourse structure in the trains project. In *Speech and Natural Language Workshop*, pages 325–330, Pacific Grove, California, February 1991.

[2] Stephanie Seneff, David Goddeau, Christine Pao, and Joe Palifroni. Multimodal discourse modelling in a multi-user multi-domain environment. In *Fourth International Conference on Spoken Language Processing* [14], pages 192–195.

[3] S. Bennacef, L. Devillers, S. Rosset, and L. Lamel. Dialog in the railtel telephine-based system. In *Fourth International Conference on Spoken Language Processing* [14], pages 550–553.

[4] Didier Pernel. Le corpus "plus". Technical Report 91-19, Notes et document LIMSI, Orsay, November 1991.

[5] Mark Stefik. Planning and meta-planning. *Artificial Intelligence*, 16(2):141–169, mai 1981.

[6] James F. Allen and Diane J. Litman. Plans, goals, and language. *Proceedings of the IEEE*, 74(7):939–947, juillet 1986.

[7] Peter van Beek and Robin Cohen. Resolving plan ambiguity for cooperative response generation. In *Proceedings of the twelfth International Joint Conference on Artificial Intelligence*, volume 2, pages 938–944, Sydney, 1991.

[8] Robert Wilensky. *Planning and understanding. A computational approach to humain reasoning.* Addison-Wesley Publishing Company, 1983.

[9] Didier Pernel. *Gestion des buts multiples de l'utilisateur dans un dialogue homme-machine de recherche d'informations.* PhD thesis, Paris XI University, 1994.

[10] Véronique Normand and Jolanda Tromp. Collaborative virtual environments: the coven project. In *2nd Framework for Immersive Virutal Environments International Conference*, pages 76–78, Palazzo dei Congressi, 1996.

[11] Ariane Halber and David Roussel. A robust linguistic analysis for spoken language interface based on a lexicalized tree grammar. In this volume.

[12] Xavier Pouteau. *Dialogue homme-machine multimodal : une communication naturelle pour l'opérateur ?* PhD thesis, Henri Poincaré - Nancy I University, 1995.

[13] John R. Searle. *Expression and meaning.* Cambridge University Press, 1979.

[14] *Fourth International Conference on Spoken Language Processing*, Philadelphia, PA, USA, 1996.