# Multi-session management in dialogue-spoken system

*Hoá NGUYEN, Jean CAELEN*

University of Joseph Fourier
Laboratory CLIPS, BP 53, 38041 Grenoble Cedex 9, France.
Ngoc-Hoa.Nguyen@imag.fr, Jean.Caelen@imag.fr

## Abstract

In order to increase the role of machines in supporting more capabilities as regards a dialogue-spoken system, we present in this paper a new problem incorporating multi-session in such a system. Instead of only handling single dialogue, such a system can take an intermediary role to communicate with many users in several discontinued sessions for reaching a compromise between them. We describe here a new approach for modeling the multi-session and then we concentrate on the multi-session management of such a system dedicated for a complete service having several tasks.

## 1. Introduction

The dialogue-spoken system has attracted much attention as the way of communicating with machines through speech. These systems normally enable users to interact with them and to perform a certain task: for example the CMU Communicator system is aimed at helping a user to create a travel itinerary consisting of air transportation, hotel reservations and car rentals [13], ARISE allows the users to consult the train's timetable [2], TRINDI enables the users to make choices in the performance of the route planning [14], etc. The dialogue in these systems contains just some exchanges between a user and the system.

In the context of company's voice portal PVE (Portail Vocal d'Entreprise) project [15], our analysis of use, which we carried out in hospitals, judicial and company offices, show that the voice service is very useful for applications such as information requests, confirmation of a request, secretarial work such as transferring calls, scheduling appointments, reserving rooms… Spoken dialogue in these situations is normally short but contains complex utterances. However, users always require a complete service that is defined like a complete resolving problem in a face-to-face situation. For example, in the room reservation service, the dialogue-spoken system (machine) must act as, behave as and take on the role of a virtual secretary. This means that the user is not only able to reserve a room, but also to request the confirmation of all participants and their availability. Moreover, the user should also be able to ask the system to negotiate with others in order to obtain a good compromise between them.

Let see the following example:

One user D would like to book the room Lafayette and he calls the system S. Unfortunately, this room is already taken by the person P. However, D has greater priority than P (may be due to hierarchical position), so he asks the system to contact P to tell him to leave this room for D. The system then contacts P and reaches an agreement with him: he accepts to put back his meeting the next day. Once the system has the response, it will recall D to inform him of the results.

**S1: <u>Person D + System S</u>**

X: hello, I am D, could you book me the room Lafayette for tomorrow at 9 o'clock, please?
S: I'm sorry Mr. D, this room is already taken by Mr. P…
X: Tell him I need it and could he leave this room for me.
S: OK, I'll contact him and I'll keep you up to date.

**S2: <u>System S + Person P</u>**

S: hello, are you Mr. P?
Y: yes,
S: I'm contacting you about the Lafayette reservation. Could you leave this room for Mr. D, please?
Y: Let me see… OK, I'll put back my meeting to tomorrow.
S: That's great, thank you very much.

**S3: <u>System S + Person D</u>**

S: Hello, Mr. D?
X: Yes, it's me
S: Mr. P has already agreed to leave the Lafayette for you at 9 o'clock tomorrow.
X: That's very nice, thank you.

So the users' requirement expecting on a dialogue system increases more and more: the dialogue-spoken system should take the role of a mediator to negotiate with several users in order to resolve the conflict between them. Therefore, we consider the dialogue between users and machine as the multi-session; each session is a dialogue between one user and the system. In this paper, we introduce an approach for modeling the multi-session, and then the mechanisms to manage them in a dialogue-spoken system.

## 2. Basic principles

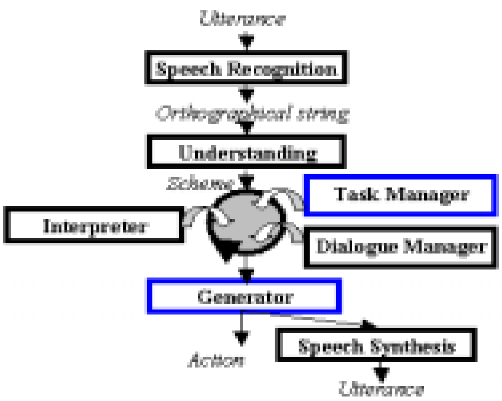


Figure 1: Architecture for a spoken dialogue system

This section describes some important elements, which are used for our multi-session modeling. In relation to the architecture for a dialogue-spoken system, we used the modular/multi-agent architecture described in [6] and as illustrated in figure 1. The multi-session management shown in the session 4 will be implemented in two modules: the dialogue manager and the task manager.

### 2.1. Speech act

Austin [1] and Searle [10] consider all utterance as an act of communication called a speech act. A speech act might contain just one word, several words or a complete sentence. By combining with the notion of illocutionary logic, Vanderveken [12] defined the illocutionary force of a speech act. Then, as Caelen [3], it is useful to retain the following illocutionary forces in the human-machine dialogue domain:

*Table 1 : Illocutionary forces of a dialogue act*

| Act | Signification |
|---|---|
| $F^A$ | Do or execute an action. |
| $F^F$ | Ask the hearer to perform an action. |
| $F^S$ | Communicate information in assertive way. |
| $F^{FS}$ | Ask for information. |
| $F^P$ | Give a choice, make an invite. |
| $F^D$ | Oblige to do without giving an alternative. |

Based on speech act theory and illocutionary logic, we define the notion of a dialogue act. A dialogue act is a speech act that is annotated by the illocutionary force. We represent a dialogue act as an illocutionary force that specifies what the speaker wishes to achieve, and a propositional content representing the semantic schema of statement. Each utterance can contain more than one dialogue act. For example, the utterance "Jean Caelen is calling… I would like to book a conference room" may be interpreted as follows:

$$\mathbf{F}^{\mathbf{S}}[\text{FirstName}(\textit{jean})\&\text{LastName}(\textit{caelen})]\ \&$$
$$\mathbf{F}^{\mathbf{F}}\underbrace{[\text{Action}(\textit{toReserve})\&\text{RoomName}(\textit{x})]}_{\text{Propositional content (p)}}$$

Illocutionary force Concept

The user dialogue act is built by the *Interpreter* module and the dialogue manager has to generate the system dialogue act.

### 2.2. Dialogue goal

A goal is generally a task state or a mental state that one wishes to reach (for example: to obtain information, to resolve a problem, etc.). The start of an exchange (a series of talking turns during which a goal is sustained) is initiated by the emergence of a new goal. Then this goal is transformed during the exchange and becomes a final goal (a task state or of a situation at the end of an exchange) at which point the exchange ends by a success or by a failure. The success obeys the double condition of being a reached goal and a satisfied goal. The final goal is not always predictable at the start.

A dialogue goal is the goal that is sustained during an exchange. In the human-machine dialogue it results from the type of considered task. For example, a room reservation implies a goal (of the task) as a request for a room and a dialogue goal that leads to a communication/negotiation to reach the goal. Thus, the dialogue goal can be satisfied while a general goal may not necessarily be satisfied [4]. The states of a dialogue goal are shown in the following table:

*Table 2: Evolution of the state of dialogue goal*

| Symbol | Status | Description |
|---|---|---|
| ?b | New | This goal has just been expressed. |
| †b | reached | The predicate b becomes true. |
| ‡b | satisfied | The speakers manifest their agreement on †b, this agreement can be explicit or implicit. |
| -b | awaited | One solves temporarily another problem. |
| b’ | repaired | Due to a lack of understanding the goal is modified, one does not go back on the previous goal. |
| sb | sub-goal | The problem is decomposed into sub-problems. |
| @b | abandoned | Following a failure or a voluntary abort. |

A dialogue goal is formed by the abstraction of dialogue act helped by the dialogue plan (which is specified in the task model by elementary goals, called task goals, and managed by the task manager). Once the dialogue manager has formed the dialogue goal, it sends this goal to the task manager to know if this goal is either reached, impossible to reach, or missed information (states concerning tasks). And then, the dialogue manager must decide itself if this dialogue goal is satisfied, awaited or left [5].

### 2.3. Dialogue strategy

The dialogue strategy $\delta$ is the way to handle the talking turns between speaker and machine to lead a dialogue goal. The strategy aims at choosing the best adjustment direction of the goals at a given moment. It is strongly a decisive factor in the dialogue efficiency, which is calculated by the speed of convergence of the dialogue acts towards the final goal. We distinguish the types of dialogue strategy by two different categories as following [5]:

- *Non-inferential strategies*: the strategies that the speaker does not need to know finally the goal of his partner
  - Directive strategy: consists in keeping the initiative to drive the dialogue: maintaining the exchange goal and keeping the initiative, imposing a new goal.
  - Reactive strategy: consists in delegating the initiative to the speaker either by making him endorse his goal, or by adopting his goal.
  - Constructive strategy: consists in moving the current goal in order to invoke a detour, for example to make it notice an error, make a quotation, and undo an old fact...
- *Inferencial strategies*: These strategies are said to be inferential to the extent that the two partners need a perceptive knowledge of their respective goals. In these strategies, the two speakers have a shared initiative.
  - Cooperative strategy: consists in adopting the goal of the hearer by proposing one (or many) solution which brings about the most relevant way to achieve their goal.
  - Negotiated strategy: can be involved in a situation where the goals are incompatible and the speakers want to minimize the concessions. The negotiation is expressed by argumentative sequences (argumentation/refutation) with the proposal for a

sub-optimal solution until convergence or acknowledgement of failure.

# 3. Multi-session modeling

Suppose now that a dialogue-spoken system must perform a complete service having several tasks. We consider that a dialogue initiated by a user D for satisfying a goal related to this service is divided into a set of discontinued sub-dialogues, each sub-dialogue representing a session $S_k$, including an opening phase, the different speech turns between the concerned user $P_k$ and the system S, and a closure phase. So, the framework of dialogue is a sequence of sessions, the first one with the requester D, then the next with the different addressees $P_k$ if necessary and at the end with D for the conclusion.

## 3.1. Definitions

We suppose during the first session $S_1$, the user D interacts with the system S for resolving the goal $b^D$. There are three possible cases at the closure of the session:

1. $b^D$ is satisfied (noted by ‡$b^D$),
2. D chooses to abort his goal (noted by @$b^D$),
3. $b^D$ cannot be reached because it is in conflict with others goals previously satisfied by others users of the service.

The third case leads new sessions to try to resolve $b^D$: in a first step the system S put the goal $b^D$ in the awaited state (noted $-b^D$) and then expands the different solutions to resolve the conflict and initiates a negotiation with the users which goals are in conflict with $b^D$.

We define:

- *Dialogue goal in conflict $b_f$*: the dialogue goal animated by the requester D is in conflict with the one already satisfied by the user P: $b_f = (-b^D, ‡b^P)$

- *Tree of dialogue goals in conflict*: more generally the goal $b^D$ is possibly in conflict with the n satisfied goals of m other users, called for the next the '*patients*', $(P_1,\ldots P_m)$ related by AND/OR operators. This set of conflict goals $T_f=(b_{f1}, b_{f2}, \ldots, b_{fm})$ makes a AND-OR tree of dialogue goals in conflict with $b^D$. Each leaf of this tree represents a goal in conflict from the patient $P_k$.

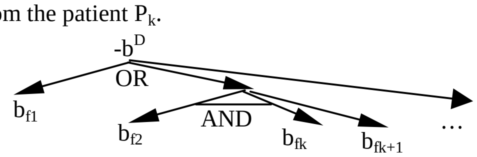


*Figure 2: A tree $T_f$ of dialogue goals in conflict*

The resolution of the conflict for $b^D$ is to find a path from the leafs to the root in respect to the AND/OR conditions along the tree. The resolution of one particular goal in conflict $b_{fk}$ in a leaf should be done by a special session issued of a dialogue $S_k$ with the user $P_k$.

## 3.2. Session coordination

Thus, the resolution of $b^D$ needs to manage dynamically several new sessions. The sessions sequence obeys the exploration of the AND-OR tree $T_f$. The goal $b_D$ will be satisfied if and only if there is at least a path in $T_f$ having all satisfied elements.

The algorithm to reach $b_D$ during sessions is shown as following:

*While* $b^D$ is not reached *Do,*

From the tree $T_f$ *Extract* the best path unmarked to reach $b^D$ and *For all* the leaf along this path, *Open* a negotiation dialogue with the concerned patient in order to solve the local conflict,

*In case of* breakdown *Mark* the path and *Try* again from the previous step

*In case of* success *Stop*

*EndWhile*

*Notify* the result to D: $b^D$ is reached (noted by †$b^D$) or abandoned (@$b^D$). From here, D could of course accept or not this result and then the dialogue could continue

In the negotiation process, the system performs each session separately, but the order of handling these sessions depends on the best path founded at each step.

# 4. Multi-session management

The multi-session management has to be done through both the dialogue manager and the task manager. The main idea here is how to manage the tree of dialogue goal in conflict. At the dialogue manager level, it controls both the dialogue goal in a session and the tree of dialogue goals in conflict $T_f$.. In relation to the task manager, it has to control the triggering, the development/execution of a session, and moreover, the coordination of the sessions sequence.

## 4.1. At the dialogue manager level

In this section, we are only interested in the management of the goal in conflict (the management of a normal dialogue goal as well as the dialogue strategy were described in [9][13], and we do not mention them here). The task manager computes and sends the tree of dialogue goals in conflict $T_F$ to the dialogue manager. Once the dialogue manager receives the tree it manages the sessions and interacts with the task manager which acts as a problem-planner.

During each negotiation the goal $b_{fi}$ goes forward according to the attitudes of $P_i$ towards $b_{fi}$. Its possible attitudes are:

- give up $b_{fi}$ to D without conditions,
- do not abandon $b_{fi}$ in all cases,
- leave out $b_{fi}$ to D within conditions as modifying $b_{fi}$, requesting a new goal $b'_{fi}$.

In the two first cases, it seems that are not complicated as the third, which depends on new conditions of $P_i$, which can be:

- feasible without the influence of others P,
- not feasible,
- feasible but it can lead to a new conflict with another P via a new session.

These attitudes manifest directly to $b_f$ via the dialogue acts of user and are recognized by the dialogue manager. The negotiation process for $b^D$ finishes when it has been reached †$b^D$, or all of possible negotiations have been failed and D has to abandon his goal @$b^D$.

## 4.2. At the task manager level

The task manager clearly takes an important role in relation to the multi-session management in a dialogue-spoken system. For ensuring the coherence of multi-sessions, it should contain the planning of all possible sessions, manage the multi-session sequence, and supervise progress of the goal in conflict.

During a session, the task manager must dynamically build $T_f$ in case of having conflicts. Once user requests to perform $T_f$, the task manager will develop a plan to negotiate with patients. For each patient $P_i$, the task manger will launch a session to resolve $b_{fi}$, and once the dialogue manager has the $b^D$ state that has already reached or abandoned.

### 4.3. Example

For modeling the multi-session in a dialogue-spoken system, we used the room's reservation service via telephone as a case study. Let us use the above example in section 1 to illustrate our approach:

In S1, the requester D manifests directly a new goal $?b^D = person(X) \wedge room(R1) \wedge date(D1) \wedge toReserve(X,R1,D1)$. However, the room requested by D was already reserved by P as $‡b^P = person(P) \wedge room(R1) \wedge date(D1) \wedge toReserve(P,R1,D1)$

By interacting with the task manager, the dialogue manager determines a room and date conflict represented by $b_f = person(x) \wedge room(R1) \wedge date(D1) \wedge toReserve(x,R1,D1)$. And then, the task manager creates $T_f = \{b_f\}$ and the dialogue manager plans a new session to negotiate with P.

Then, the task manager interacts with the dialogue manager to launch a new session $S_2$ for resolving $b_f$. The system S calls P and suppose the negotiation in this case happens successfully: P accepts for moving his meeting to the next day so the goal in conflict has been resolved, because $‡b^P$ becomes $‡b^P = person(P) \wedge room(R1) \wedge date(D2) \wedge toReserve(P,R1,D2)$ with D2=D1+1 and $b_f=b^D$. The task manager should acknowledge these new situations and plans making a new session to inform the results to D.

The third session $S_3$ is just to notify to D the state of $b_f$, a reached goal now. Naturally, D could also deny $b_f$ by such reasons, but fortunately, he recognizes $b_f$ and manifests it to be satisfied. So the dialogue animated by D has been completed.

## 5. Results and conclusion

Multi-session management in a discontinued human-machine dialogue has become necessary in increasing the capability of the dialogue-spoken system. Based on the dialogue management which is reduced as much as possible the dependence on task model, we have built a prototype of such a system dedicated for the reservation service aimed in the PVE project (by French language). Our prototype could currently manipulate the sessions like the room reservation, meeting convocation, and moreover, the cancellation/modification of a reservation. By applying our methodology of multi-session modeling and management, our prototype can now act like a real mediator: users could ask the system to negotiate with another user in case having conflicts of room, date.

The experimentations, which have carried out with our prototype with the corpus collected during the Wizard of Oz step in the PVE project, prove the validity of our theory for the multi-session management. We have also done a lot of tests within multi-session for resolving the room/date conflict, and we will publish the official result evaluation later In the near future, with the speech-recognized improvement, the robust comprehension/interpretation, our system will be totally completed with the best negotiation capability.

The first results we have obtained and are obtaining not only show the importance of multi-session management in a dialogue-spoken system, but also open a new direction in the way of bringing intelligence and speech to machines.

## 6. References


[1] Austin, J. L., *How to Do Things with Words*, Harvard University Press, 1962.

[2] Baggia P., Gauvain J.L., Kellner A., Perennou G., Popovici C., Sturm J., Wessel F., "Language Modelling and Spoken Dialogue Systems - the ARISE experience", in Proc. Sixth European Conference on Speech Communication and Technology, September 1999.

[3] Caelen, J., "Attitudes cognitives et actes de langage". In Du dialogue, Recherches sur la philosophie du langage, n° 14, Vrin éd., Paris, p. 19-48, 1992.

[4] Caelen J., "Modèles formels de dialogue", Actes du GdR I3 - *Information, Interaction Intelligence*, CEPADUES Editions, p. 31-58, 2002.

[5] Caelen J., Stratégies de dialogue, Actes de conférence MFI'03 (Modèles Formels de l'Interaction), Lille, CEPADUES ed, 2003

[6] Caelen J., Nguyen H., "Gestion de buts de dialogue", presented at the 11th conference on Natural Language Processing - TALN, April 2004.

[7] Mittal S., Dyn C.L., Morjaria M.. "PRIDE: An expert system for the design paper handling systems", in Application of knowledge-based systems to engineering analysis and design, C.L. Dyn ed., American Society of Mechanical Engineers, 1985

[8] Nguyen H., "Vers une architecture générique de système de dialogue oral homme-machine". RECITAL-France 6-2003.

[9] Nguyen H., Caelen J., "Generic manager for spoken dialogue systems". DiaBruck: 7th Workshop on the Semantics and Pragmatics of Dialogue, Proceedings pp.201-202, 2003.

[10] Peckham K., *A new generation of spoken dialogue systems: Results and Lessons from the SUNDIAL project*, Proceedings of Eurospeech, p. 33-40, 1993.

[11] Searle, J., *Speech Acts: An Essay in the Philosophy of Language*, Cambridge University Press, 1969.

[12] Vanderveken D., *Meaning and Speech Acts*, Volumes 1 and 2, Cambridge University Press, 1990.

[13] Xu, W. and Rudnicky, A. "Task-based dialog management using an agenda". ANLP/NAACL 2000 Workshop on Conversational Systems, May 2000, pp. 42-47.

[14] http://www.ling.gu.se/projekt/trindi/

[15] www.telecom.gouv.fr/rnrt/projets/res_01_5.htm