COST278 and ISCA Tutorial and Research Workshop
on Robustness Issues in Conversational Interaction
University of East Anglia, Norwich, UK
August 30-31, 2004

# Multi-session dialogue modeling and management in spoken dialogue system

*Hoá NGUYEN, Jean CAELEN*

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

## Abstract

We present in this paper a new problem incorporating the multi-session dialogue of a spoken dialogue system. Instead of engaging a single user in a dialogue, we aim at building a dialogue system that allows engage in a dialogue with multiple successive users. The system can then take an intermediary role to communicate with many users in several discontinuous sessions for reaching a good compromise between them. We describe here a new approach for modeling the multi-session dialogue and then we concentrate on the multi-session dialogue management of such a system dedicated to a complete service having several tasks.

## 1. Introduction

The spoken dialogue in most current dialogue systems is normally short and engages a single user to interact with system to perform a certain task. For example, CMU Communicator aims at helping a user to create a travel itinerary consisting of air transportation, hotel reservations and car rentals [10], ARISE allows the users to consult the train timetable [2], etc.

In the context of PVE project [11], our analyses bring an important remark to us: users always require a complete service that is aimed to resolve a whole problem in a face-to-face situation. Several tasks could be invoked during a complex dialogue that concerns possibly multiple users.

Let see the following example: one person D would like to absolutely book the Lafayette room and he calls the system S. Unfortunately, this room is already taken by two persons, P1 and P2. However, D has a greater priority than them (may be due to hierarchical position), so he asks the system to contact and tell them to leave this room for D. S then contacts successively P1 and then P2, and obtains a good agreement with both of them: P1 accepts to put back his meeting to the next day and P2 decides to change his room. After that, the system will recall D to inform him about the result.

**S1: <u>Person D + System S</u>** ***15h30, 17/05/2004***

D: Hello, I am D. Could you book me the Lafayette room for tomorrow from 10h to midday, please?
S: I'm sorry Mr. Director, but this room was already reversed by Mr. P1 from 9h to 11h and by Mr. P2 from 11h to 13h. However, the Aquarium room is available for the whole tomorrow…
D: No, it's too small. Tell them I need the Lafayette and could they leave room for me.
S: OK, I'll contact them and I'll keep you up to date.

**S2: <u>System S + Person P1</u>** ***15h35, 17/05/2004***

S: hello, are you Mr. P1?
P1: yes,
S: I'm contacting you about the Lafayette reservation tomorrow morning. Mr. D would absolutely have it, so could you leave him this room, please?
P1: OK, but put back my reservation to the next day.
S: That's great. You'll have Lafayette at the same time the day after tomorrow. Thank you very much.

**S3: <u>System S + Person P2</u>** ***15h40, 17/05/2004***

S: hello, are you Mr. P2?
P2: yes, P2 is speaking…
S: Mr. D has an important meeting tomorrow morning. So could you leave the Lafayette for him, please?
P2: um… OK, please book the Aquarium room at the same time for me and I don't want any more change...
S: Thank you for your generosity. You'll have the Aquarium tomorrow from 11h to 13h.

**S4: <u>System S + Person D</u>** ***16h00, 17/05/2004***

S: Hello, Mr. D?
D: Yes, it's me
S: Mr. P1 and P2 have already agreed to leave the Lafayette for you at 9h tomorrow.
D: That's very nice, thank you. And inform also all member of PVE project for me please.
S: OK, I'll call them immediately. Good-bye Mr. D.

Thus we can see that the users expect more than more of their requirement towards a dialogue system: the spoken dialogue system should now take the additional role of a mediator to negotiate with several users, in case of necessary, in order to resolve the conflict between them. There are possibly multiple users engaged in a dialogue. Therefore, we consider now the dialogue is expressed by multi-session with multiple users; each session is a dialogue between a single user and the system. In this paper, we introduce an approach for modeling the multi-session dialogue, and then the mechanisms to manage them in a spoken dialogue system.

## 2. Basic principles

This section describes some important elements, which relate to our multi-session dialogue modeling. In relation to the architecture for a spoken dialogue system, we used the modular/multi-agent architecture described in [6] with the two important modules "dialogue manager" and "task manager" related to this paper.

### 2.1. Dialogue act

Austin [1] and Searle [8] 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 [9] 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.

### 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.). A dialogue goal is the goal that is sustained during an exchange between system and user. The beginning of a dialogue makes emerge a new goal. Then this goal is transformed during the exchange and becomes a final goal, at which point the exchange ends by a success or by a failure. A dialogue goal is represented as a logical predicat **b** and its possible states are shown in the following table:

*Table 2*: Evolution of the dialogue goal state

| | **Status** | **Description** |
|---|---|---|
| ?b | new | This goal has just been expressed by user. |
| †b | reached | The predicate b becomes true. |
| ‡b | satisfied | User manifests their agreement on †b, this agreement can be explicit or implicit. |
| -b | pending | System solves temporarily another problem. |
| b' | repaired | Due to a lack of understanding, the goal is modified; user does not go back on his previous goal. |
| sb | sub-goal | The problem is decomposed into sub-problems. |
| @b | abandoned | This state is result of a failure or a voluntary abort. |

A dialogue goal is formed by the abstraction of dialogue act helped by the dialogue agenda (which is specified in the task model). 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 whether this dialogue goal is satisfied, pending or abandonned[5].

### 2.3. Dialogue strategy

The dialogue strategy $\delta$ is the way to handle the talking turns between a user and the system to lead the dialogue goal of user. The strategy aims at choosing the best adjustment direction of the goal at a given moment. We distinguish the types of dialogue strategy by two different categories as following [4]:

*Non-inferential strategies*: the strategies that the system does not need to know initially the user's goal

- Directive strategy: consists in keeping the initiative with the system to drive the dialogue, maintaining the exchange goal and, imposing a new goal.
- Reactive strategy: is used to delegate the initiative to the user either by making him endorse the dialogue goal, or by adopting this goal.

*Inferential strategies*: These strategies are said to be inferential when both user and system 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 user by proposing one or many solutions which directly bring the best way to reach his goal.
- Negotiated strategy: can be involved in a situation where the goals are incompatible and the both user and system want to minimize the concessions.

## 3. Multi-session dialogue modeling

Suppose now that a spoken dialogue system S must perform a whole service having several tasks. In normal dialogue, typically user is either performer of a task or that who desire the task to be done. However, in some special cases, the goal that user wishes to achieve cannot directly be reached and there are hence multiple other users successively engaged in his dialogue in order to obtain his goal.

### 3.1. Approaches

We now consider that a user D interacts with S to reach the goal $?b_D$. There are the following possible cases:

- $b_D$ could be satisfied and becomes $‡b_D$ without conflict with the others,
- D chooses to abort this goal $@b_D$,
- $b_D$ cannot be directly reached because it is in conflict with others goals previously satisfied by others users.

We define:

- *Patient:* a user who owns an already satisfied goal but this goal is in conflict with the goal $b_D$ concerned by a new dialogue between D and S.
- *Dialogue goal in conflict $b^F$*: the dialogue goal animated by the requester D is in conflict with the one already satisfied by the patient P: $\mathbf{b^F} = (b_D, ‡b_P)$
- *Tree of dialogue goals in conflict $T^F$*: more generally the goal $b_D$ could be in conflict with the n satisfied goals of n patients $(P_1,\dots P_n)$ related by AND/OR condition and produces a set of goals in conflict $(b^{F1}, b^{F2}, \dots, b^{Fn})$. This set makes a AND-OR binary tree $\mathbf{T^F}$ of dialogue goals in conflict with $b_D$. Each leaf of this tree represents a goal in conflict from the patient $P_k$, and each node is an AND/OR Boolean operator. For simplicity, we call $T^F$ as the tree of goals in conflict.

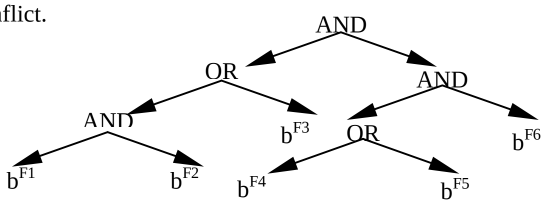


*Figure 1:* A sample tree $T^F$ of goals in conflict

- *Path for traveling $P^F$*: is a path leading from leafs to the root in $T^F$ in respect to all AND/OR conditions along this tree. The set of their leafs makes a list of goals in conflict $P^F = \{b^{Fi} \wedge b^{Fj} \wedge \dots\}$. For example, in tree $T^F$ of figure 1, there are paths like $\{b^{F1} \wedge b^{F2} \wedge b^{F4} \wedge b^{F6}\}$, $\{b^{F3} \wedge b^{F5} \wedge b^{F6}\}$, …
- *Dialogue session*: a set of interactions between a single user and the system, counted from the first utterance until the final.
- *Multi-session dialogue:* In case of $b_D$ in conflict with patients, a set of discontinuous, discrete dialogue sessions must be happened to resolve the conflicts. This set makes a multi-session dialogue. Multiple users should be obviously engaged in a multi-session dialogue.

### 3.2. Sessions coordination

In a multi-session dialogue, the sessions coordination is an important task. It has to ensure the coherence and the progress of such a dialogue.

The resolution of each goal in conflict $b^{Fi}$ should naturally be done by making a new session. And then the resolution of the conflicts for $b_D$ is to find a no-traveled path for traveling by respecting the AND/OR conditions along the exploration of the tree. The goal $b_D$ will be reached if and only if there is at least such a path in $T^F$ having all satisfied elements.

By the evolution of $b_D$, we divide a multi-session dialogue by three following phases:

- ***Emergent*** phase: This phase has only one session by the emergence of new goal $?b_D$ expressed by D. However, this goal is in conflicts with patients' other ones. S has to identify and build the tree $T^F$ in this session. And by requesting of D, S puts $b_D$ in the pending queue and plans the process of conflicts resolution.
- ***Negotiation*** phase: contains possibly several sessions. As long as $b_D$ has not been reached yet and there is still a no-traveled path for traveling, S has to contact the patients to negotiate with them for a good compromise towards $b_D$. This process will be stopped when either $b_D$ is reached or S has scanned all possible paths in $T^F$ and there are no more paths for traveling, and S has to abandon $b_D$.
- ***Notification*** phase: concerns only one session: S re-contacts D and informs about the negotiation result ($b_D$ is either reached $\dagger b_D$ or abandoned $@b_D$). D could also agree or refuse this result but the dialogue will be ended in this session.

In the negotiation process, the system performs each session separately, but the order of handling these sessions depends on the concrete service and we will detail it in the next section.

## 4. Multi-session dialogue management

The multi-session dialogue management has to be done through both the dialogue manager and the task manager in the architecture for a spoken dialogue system described in[6].

When a user D requests a new goal $?b_D$, the task manager has to identify all the goals in conflict and then build the tree $T^F$. Once the dialogue manager receives $T^F$ from the task manager, it has to control the session coordination and manage $T^F$. In the dialogue manager, we have developed a module called "***Expert***" that ensures this task. The multi-session handling in the dialogue and task manager will be detailed in the next sub-sections.

### 4.1. At the dialogue manager level

For a multi-session dialogue, beyond the roles of computing the adequate strategy, determining the dialogue goal and generating the dialogue act of the system for a normal dialogue[7], the dialogue manager has to assume the following complement responsibilities related the resolution of conflicts:

- Find the best strategy for exploring efficiently the tree of goals in conflict $T^F$ to obtain the best negotiation process.
- Coordinate all the sessions in a multi-session dialogue for making a coherent and natural dialogue.

During the emergent session, once the task manager determines that there are conflicts, S notifies the user of these conflicts. We note his attitudes towards $T^F$ as following:

$\mathbf{F^A_D(b_D)}$ => $\mathbf{F^S_S(T^F)}$ : S informs about the problem to D,

$\mathbf{F^{FS}_D(T^F)}$ => d=**directive** Ù $\mathbf{F^S_S}$**(explications** $\mathbf{T^F}$**)** : S explains D the phenomenon of $T^F$,

$\mathbf{F^F_D(T^F)}$ => d=**negotiated** Ù **-**$\mathbf{b_D}$ Ù$\mathbf{F^A_S(T^F)}$ : S changes the strategy to negotiation, and perform $\mathbf{T^F}$ if D requests to do it.

$\mathbf{F^F_D(\neg T^F)}$ => d=**cooperative** Ù $\mathbf{F^{FS}(b'_D)}$ : If D do not want to resolve the conflicts, then S should offer her another solution representing by the repaired goal $b'_D$ and apply the cooperative strategy.

Obviously, we care only the case that D asks S to resolve the conflicts because of multi-session requirement. In this case, the dialogue manager has to hang up the current session with D and start the negotiation process controlled by the **Expert** module. The goals of this module are to:

- Find the best no-traveled path $P^F$ in the tree of goals in conflict $T^F$ by respect all AND/OR conditions.
- Organize the time to contact each patient in $P^F$. The contact order should be considered and planed in this module.

The algorithm allows Expert to ensure his roles was implemented as following:

```
While (bD != †bD){
    PF = GetBestPath(TF);//find the best no-traveled path PF in TF.
    If (PF == noPath){
        Notify(D,@bD);//notify D of the no-solution of bD.
        Exit;   }
    negotiationResult = true;
    For each  bF in PF{
        Negoticate(bF);//open a negotiation session with the related
patient P in order to resolve the local conflict with him.
        UpdateTraveledStatus(TF);//update the traveled status
for all nodes and leafs in TF.
        If (bF != ‡bF) {
            negotiationResult = false;
            break;//terminate negotiate of the current path.
    }   }
    If (negotiationResult) {
        Notify(D, †bD); //Stop the negotiation process and notify D
of the negotiation result †bD.
        Exit;
}   }
```

The strategy to find the best no-traveled $P^F$ naturally depends on the application contexts. However, we can use some criteria as following:

- The reply time limit of a patient that specified the deadline S could have the negotiation result with him (determined based on his agenda). The reply time limit total of a no-traveled path $P^F$ so is the maximum limit of all patients in $P^F$.
- The negotiation flexibility of a patient P that estimates the successful negotiation degree between S and P. This parameter varies from 1 (successful) to 0 (failed) and is defined while building the user model. However, it could be changed after each multi-session dialogue. The negotiation flexibility total of a no-traveled path $P^F$ is so calculated by multiplying all the one of each patient in $P^F$.

Thus depending on each concrete case, Expert can choose the adequate strategy to find the best no-traveled path. For example, in case of urgent request of D, Expert will apply the strategy of finding the no-traveled path having the minimum reply time limit total. However, with an allowing deadline, Expert will choose the best no-traveled path by finding the path having the maximum negotiation flexibility total.

The dialogue strategy applied to start a negotiation session with a patient P should be used depending on the relation degree between roles of D and P in S, which is described in the user model. The following rules should be used to initiate each negotiation session:

- If role(P) > role(D), then the dialogue manager will apply the cooperative strategy to initiate the negotiation session with P: $\delta$=cooperative,

-If role(P) <= role(D), then the directive strategy will be used to start the negotiation session with P, δ=directive.

Naturally, during a negotiation session, the dialogue strategy δ should be recalculated and can be changed as an ordinary dialogue.

The negotiation process for $b_D$ finishes when it has been reached †$b_D$, or all of possible negotiations have been failed and S should abandon this goal @$b_D$.

After that, the dialogue manager launches the notification session as a normal dialogue to inform D about the negotiation result ($b_D$ is either reached or abandoned).

### 4.2. At the task manager level

Beyond the data/object management as usual, the task manager clearly takes an important role in relation to the multi-session dialogue management. The task manager should identify all possible conflicts and then build the tree of goals in conflict $T^F$. It controls also the session trigger for each session in the negotiation process and notification session.

The identification of conflicts is based on the dialogue goal structure, which is represented by a set of indivisible elementary goals, and the previous dialogue goals kept in the agenda, which is managed by the task manager. By comparing the new goal asked by a user with all the goals that have already satisfied but have not occurred yet, the task manager will find all possible conflicts.

The identification phase of goals in conflict should be done in parallel with the determination of logical relations between them. Based on these logical relations, the task manager will build the tree of goals in conflicts $T^F$. Once the task manger formed the tree $T^F$, it sends it directly to the dialogue manager. At here, the dialogue will inform user about the $T^F$ phenomenon and the Expert will take his role to lead the negotiation process.

Another mission of the task manager is to trigger a session. It has to manage a session queue that has been already calculated when it could launch such session.

## 5. Example

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

In the session S1, the requester D manifests directly a new goal *?$b_D$ = person(D)Ùroom(Lafayette)Ùdate(18/05/2004)Ù hour(10h)Ùlength(2h)*. However, the Lafayette was already reserved by two patients P1 and P2 with their satisfied dialogue goals: ‡$b_{P1}$ = person(P1)∧room(Lafayette)∧ date(18/05/2004)∧hour(9h)∧length(2h);and ‡$b_{P2}$= person(P2) ∧room(Lafayette)∧date(18/05/2004)∧hour(11h) ∧ length(2h).

So the task manager identifies these two conflicts $b^{F1}$ = ($b_D$, ‡$b_{P1}$) and $b^{F2}$ = ($b_D$, ‡$b_{P2}$) which are related by AND operator, and builds the tree of goals in conflict $T^F = \{ b^{F1} \wedge b^{F2}\}$. Once the dialogue manger receives the $T^F$, it generates the dialogue act to inform D about these conflicts. Moreover, S proposes also an available room for D.

Although the proposition of S, D does not agree in order of the available room, Aquarium and he forces S to resolve the conflicts because of his role (director) greater than the two patients (researchers). So S registers the tree $T^F$ and plans creating a new session to negotiate with P1 and P2.

After that, the Expert finds a path for traveling in $T^F$: $P^F = \{b^{F1} \wedge b^{F2}\}$ and plans the negotiation phase with P1 and P2. Naturally, in this case of $T^F$, Expert reaches directly $P^F$ even applied all of two strategies mentioned above to find the best no-traveled path in $T^F$.

Firstly, to resolve $b^{F1}$, S calls P1 using the directive strategy at initiation (because of D has the role greater than P1) and the negotiation happens successful with a good solution: P1 accepts for moving his meeting to the next day (and fortunately, the Lafayette room is available at that time) so $b^{F1}$ has been resolved †$b^{F1}$. That leads the system continues calling the P2 (initiated by the directive strategy, with the same reason) and obtains also a good compromise: P2 accepts to leave the Lafayette to D with condition to change the room but keep the date/time.

Finally, S has reached the best result in negotiation process and $b_D$ has become the reached goal †$b_D$. So S launches a new session to notify D of this result.

## 6. Results and conclusion

Multi-session dialogue handling in a spoken dialogue system becomes now a necessary require in increasing the capability of the 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 to the meeting organization service aimed in the PVE project (by French language). Our prototype could currently handle the tasks like the room reservation, meeting convocation, and the reservation modification... By applying our approach of multi-session dialogue modeling and management, it can now act as a real mediator: user could ask the system to negotiate with another in case having room conflicts.

The experimentations, which have also carried out with our prototype, by ignoring speech recognizer module, prove the validity of our theory for the multi-session dialogue management. The first results we have obtained and are obtaining show the importance of multi-session dialogue management in a spoken dialogue system. In the near future, with the speech-recognizer accuracy, our system will be totally completed with the best negotiation capability for many conflicts.

## 7. 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", *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., Stratégies de dialogue, Actes de conférence *MFI'03 (Modèles Formels de l'Interaction)*, Lille, CEPADUES ed, 2003

[5] Caelen J., Nguyen H., "Gestion de buts de dialogue", *Natural Language Processing – TALN conference*, April 2004.

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

[7] 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.

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

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

[10] Xu, W. and Rudnicky, A. "Task-based dialog management using an agenda". *Workshop on Conversational Systems*, 2000.

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