Ambiguous Turn-Taking Games in Conversations

Gemma Bel-Enguix, Maria Dolores Jiménez-López

Rovira i Virgili University
Research Group on Mathematical Linguistics (GRLMC)
{gemma.bel ; mariadolores.jimenez}@urv.net

Résumé

L'interface homme-machine a besoin de modèles de structures de dialogue qui expliquent la variabilité et la spontanéité au dialogue. Le contexte sémantique et pragmatique évolue continuellement pendant le développement de la conversation, surtout par la distribution de *turns* qui ont un effet direct dans les échanges de dialogue. Dans cet article nous utilisons un paradigme de langue formel pour modéliser les conversations de système de multiagent. Notre modèle computationnel combine des unités minimales pragmatiques -les actes de parole- pour construire des dialogues. Dans ce cadre, nous montrons comment la distribution de *turn-taking* peut être ambiguë et proposer un algorithme pour la résoudre, considérant *turn coherence*, *trajectories* et le *turn-pairing*. Finalement, nous suggérons *overlapping* comme un des phénomènes possibles naissants d'un *turn-taking* non résolu.

Mots-clés : dialogue, *dialogue games*, *turn-taking*.

Abstract

Human-computer interfaces require models of dialogue structure that capture the variability and unpredictability within dialogue. Semantic and pragmatic context are continuously evolving during conversation, especially by the distribution of turns that have a direct effect in dialogue exchanges. In this paper we use a formal language paradigm for modelling multi-agent system conversations. Our computational model combines pragmatic minimal units –speech acts– for constructing dialogues. In this framework, we show how turn-taking distribution can be ambiguous and propose an algorithm for solving it, considering *turn coherence*, *trajectories* and *turn pairing*. Finally, we suggest overlapping as one of the possible phenomena emerging from an unresolved turn-taking.

Keywords: dialogue modelling, dialogue games, turn-taking.

1. Introduction

Dialogue acts are a key issue in pragmatics that has been traditionally tackled by artificial intelligence. Nevertheless, computer science has been more interested in the structure and the interaction of agents in conversation (Litman and Allen, 1990) rather than in single utterances. From that perspective, some authors (Sinclair and Coulthard, 1975; Coulthard *et al.*, 1981) have distinguished different levels inside the dialogue : *act*, *move*, *exchange* and *transaction*. Such classification has become classical, especially referring to moves (Kowto *et al.*, 1993; Traum and Larsson, 2003).

Traum and Hinkelman (1992) give a new perspective in the topic gathering different types of acts in four groups: *turn-taking*, *grounding*, *core speech acts* and *argumentation*. One of the novelties of this taxonomy consists in the inclusion of turn-taking into such categories.

A similar idea can be found in Bunt (2005), who defends the existence of *task-oriented acts* and *dialogue control acts*. Both types modify the linguistic and cognitive context. However, task-oriented acts only change the semantic context and dialogue control acts mainly affect the social or physical context. He considers turn-taking to be included into *interaction management functions*, which belong to *dialogue control functions*.

On the other hand, some researchers (Carlson, 1983; Prakken, 2005) have used the term *dialogue games* referring to task-oriented acts, which are usually considered *moves*. The idea is that agents behave like in games, planning and executing moves to achieve their goals. Several papers (Kowto *et al*., 1993; Poesio and Traum, 1997) have given sequences of moves for completing dialogue games, as well as interesting computational implementations (Poesio and Traum, 1997; Hajdinjak and Mihelic, 2004).

Regarding the interaction between the agents, Reed and Long (1997) introduce an interesting distinction between *collaboration* and *cooperation*. While collaboration refers to scenarios where agents work together to reach a desired outcome, cooperation is related to situations where the outcome is achieved through 'selfish' participation constrained by contextual factors.

This paper in focused in the study of *turn-taking* as a *dialogue game*, and the consequences of its *ambiguous assignation* in *exchanges* with *cooperative* (non-collaborative) agents. Perhaps turn-taking acts do not modify the semantic context, but we claim with Bunt (2005) that they are very important in the configuration of the *pragmatic context*. In the paper, we describe a protocol for the turn-taking assignment and explain how its distribution can cause some dysfunctional phenomena in conversations, as the turns overlapping and the breaking up of the dialogue in several sub-dialogues. Our goal is not to give a new taxonomy of sequences of moves inside dialogue games. Nevertheless, we intend to put together different approaches and considerations about turn giving with the purpose of introducing a formal algorithm for dealing with turn management.

The steps performed in this paper in order to describe these dysfunctional pieces of dialogue caused by ambiguity in the turn-taking are the following: a) to formalize a class of dialogue systems to be the theoretical framework of the work (section 2) ; b) to explain the way the turn-taking is distributed in the model, and the trajectories for the turn-taking, which correspond to different configurations of turn-taking depending on the dialogue games (section 3) ; c) to establish the protocol of turn-taking distribution for ambiguous cases (section 3) ; and d) to describe the phenomena of *overlapping*, that is generated in a dialogue with non-collaborative agents by an unsolved turn-taking (section 4).

2. Definition of Dialogue Systems

Since the aim of the paper is not to offer a classification of acts, we have considered here a restricted set of them, based on the one given by Kowto *et al*. (1993), which has been simplified and adapted in order to fit to a simple explanation of turn-taking. In our framework, dialogue systems are conceived as formal generators of connected speech acts.

Definition 1 *Dialogue systems can be defined as a 4-uple* (N, V, Σ, G)*, where:*
- $N = \{N_1, N_2, \dots, N_n\}$ *is the multi-agent structure, where every one of the agents has a configuration* $N_i = (R_i, C_i)$*, being:*
  - R_i *the set of rules. Every one has at the left side turn-taking for the rule to be applied. At the right side: a) the rule of generation, and b) the agent whom the speech act is addressed to, if it exists.*

  – C_i *is the state of the communication channels. Every agent has bidirectional (input/output) channel with the others, that can be set either as open or closed. For the communication to be possible between two given agents* N_n *and* N_i*, the channel has to be open in both directions. For the other combinations (open-closed, closed-open, closed-closed) communication is not possible.*
- $V = \{\kappa, \alpha, \gamma, \varphi, \pi, \varepsilon, \sigma\}$ *is the set of stores of core speech acts, being their configuration as follows:* $\kappa = \{k_1, k_2, \ldots, k_n\}$, query *;* $\alpha = \{a_1, a_2, \ldots, a_n\}$, answer *;* $\gamma = \{g_1, g_2, \ldots, g_n\}$, agree *;* $\varphi = \{f_1, f_2, \ldots, f_n\}$, reject *;* $\pi = \{p_1, p_2, \ldots, p_n\}$, prescription *;* $\varepsilon = \{e_1, e_2, \ldots, e_n\}$, clarify *;* $\sigma = \{s_1, s_2, \ldots, s_n\}$, assert.
- $\Sigma = \{\boldsymbol{K}(Query), \boldsymbol{A}(Answer), \boldsymbol{G}(Agree), \boldsymbol{F}(Reject), \boldsymbol{P}(Prescription), \boldsymbol{E}(Clarify), \boldsymbol{S}(assert)\}$ *is the set of turn-taking dialogue acts.*
- $G = \{G_1, G_2, \ldots, G_n\}$ *is the set of registers of the system, the place where the dialogue games are stored. There is a different generation register for every conversation started by the system.*

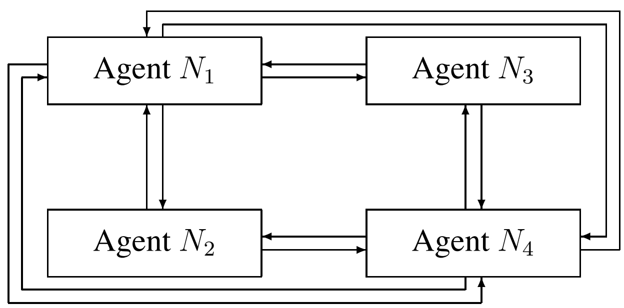

Figure 1. A Connected Dialogue Multi-Agent System

The system is able to generate dialogues of several types starting from a simple formalization, as it can be found in (Bel-Enguix and Jiménez-López, 2005). Moreover, the definition of agents as related contextual generators allows these devices to deal with pragmatic and semantic context, that are continuously evolving during the process of utterance generation.

Considering what has been said above, the purpose of the paper is to tackle turn-taking as one of the most important pragmatic aspects in conversation, so as turn-taking assignation can model the whole development of conversation as well as the final configuration of the context.

3. Turn-Taking Games

As it is shown in the definition, turn-taking is distributed by the rules of generation. For instance, a rule like $K \rightarrow a; A_3$ allows the agent to answer a question. After the generation, the speaker sends the turn A to the agent N_3. When the agent N_3 receives A, it takes the turn if and only if it has a rule with A at the left side. This right-left concatenation between speaker and hearer/speaker is called *turn coherence*.

Turn-taking cannot be considered to be a conversational game in the scenario above presented, since it is always well established. But the general experience in everyday life is quite different in this point and we know that the turn is not always clearly assigned. For the system to model those situations, rules like $E \rightarrow k; K$ are allowed, where the agent utterance does not have a clear addressee. In this situation, ambiguous turn-taking distribution can emerge, as it can be seen in figure 2.

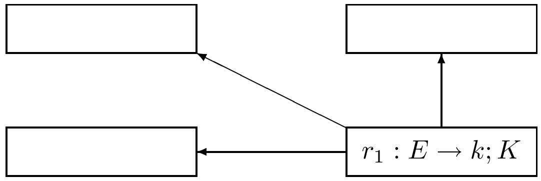

Figure 2. Ambiguous Turn-Taking

Our goal is to establish protocols for disambiguation of turn-taking when this is not clearly assigned by rules. These protocols should help the system to decide what agent has to talk at every step. In such scenario the process for deciding the assignation is seen as a game. For this situation we establish a protocol to help the system to disambiguate the turn assignation.

There are three criteria for the system to decide the correct agent to talk:

1. Turn coherence ;
2. Trajectories ;
3. Agent pairing.

These three aspects are in order of precedence. Therefore, in the situation of ambiguity the system tries to solve the problem applying turn coherence. If it is not sufficient, trajectories are considered. If trajectories are not able to give correct parameters to decide, then agent pairing is introduced. If the three components introduced here do not lead to disambiguate the turn-taking, then, taking into account that agents are not collaborative, at least two of them will talk at the same time.

3.1. Turn Coherence

Turn coherence is a condition that must be fulfilled in every exchange of turns in conversation. The fact of being obligatory makes it a good tool for turn disambiguation. Indeed, in figure 2 a system is shown where the agent with N_4 makes a query, which is an utterance with an ambiguous turn distribution. Therefore, the turn-taking is ambiguous from the point of view of the addresser.

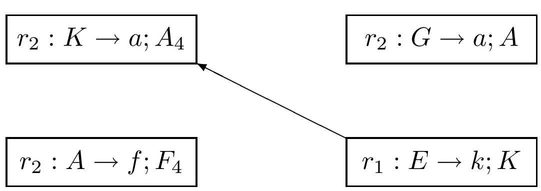

Figure 3. Disambiguation by Turn Coherence

Nevertheless, sometimes this situation can be easily disambiguated just looking at rules stored by every one of the addressees. In the example of figure 3 it is shown how agents N_1, N_2 and N_3 have rules to talk, but only the one in N_1 can be applied, because it is the only one fulfilling the turn coherence with r_1. Turn coherence is a simple method that resolves many of the situations of ambiguity in turn-taking, either in a total or partial way.

3.2. Trajectories

Turn-taking trajectories define some standard configurations of dialogue games. From the set of speech acts defined above, a simple arrangement of these trajectories will be developed. We assume that, from the set of acts selected in the paper, only π (prescription), κ (query) and σ (assert) can be the starting move in an exchange. Therefore, as a methodological option for explaining the concept of trajectory, three main types of exchanges are established: *prescription exchanges*, *query-answer exchanges* and *assert exchanges*.

For *prescription*, the possible responses are *agree* or *reject*. If the reply is *agree*, then the game is closed. If it is *reject*, there are two possibilities: a) to close the exchange, b) to perform a *clarify* movement, which has to be followed either by an *agree/reject* or by a *query* act. In the last case, a path *answer* + *agree/reject* is expected. *Prescription* exchanges must finish with an *agree/reject* movement performed by an agent different from the one that uttered π.

The scheme of prescription trajectories is the following, assuming no iterative paths are performed, and taking into account that only *agree/reject* can be a final move, is the following:

– prescription - agree/reject ;
– prescription - reject - clarify - agree/reject ;
– prescription - reject - clarify - query - answer - agree/reject.

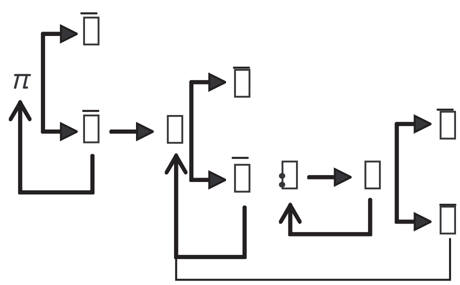

Figure 4. Trajectories for Prescription Exchanges

In that general scheme iterative cycles should be also considered. For instance, the adjacency pair *query* + *answer*, can be repeated many times. In the same way, *prescription* + *rejection* could constitute a buckle. The same can be said about the path *clarify* + *reject*. All this can be seen in figure 4, where closing movements are marked with a bar.

There are also several trajectories for a *query-answer exchange*, involving the possibility of a cycle with the *query-answer* adjacency pair and *clarify* movements if the answer is not satisfactory. The final movement can be either an *answer* or a *agree/reject* act.

In query-answer exchanges, the agent uttering the last act can be either the one that started the trajectory or another one in the system. This is because often an answer does not need an *agree/reject* movement.

Therefore, the scheme of query-answer trajectories is the following:

– query - answer ;
– query - answer - agree/reject ;
– query - answer - reject - clarify - agree/reject.

If iterative paths are considered, the general scheme of query-answer exchanges can be seen in figure 5.

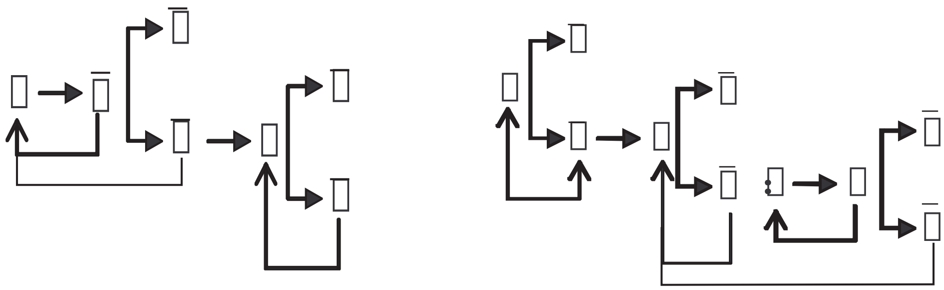

Figure 5. Trajectories for Query-Answer and Assertion Exchanges

The trajectories for an *assert exchanges* have an immediate response which is either *agree* or *reject*. If the answer is *agree*, some illocutionary influence is given. If the movement is *reject*, then a *clarify* movement may be performed. Rejections can introduce also *query-answer exchanges* nested in assertion.

The general structure of trajectories for assert exchanges is the following:

– assert - agree/reject ;
– assert - reject - clarify - agree/reject ;
– assert - reject - clarify - query - answer - agree/reject.

As it can be seen in figure 5, the structure of prescription and assertion is quite similar, whereas query-answer can be easily considered as part of the other exchanges.

Once trajectories have been described, they can be used to validate an exchange performed by the agents and, if necessary, to disambiguate the turn-taking. Figure 6 is an example of turn-taking ambiguity that cannot be solved by turn coherence, because every agent fulfils this condition. Therefore, the adaptation to trajectories is the next criterion to apply.

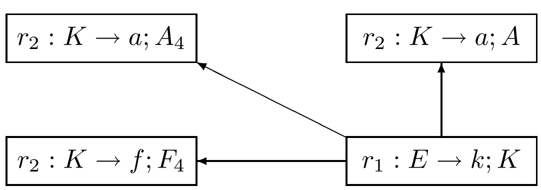

Figure 6. Unsolved Turn-Taking Ambiguity

By means of turn-taking trajectories, one of the moves is dismissed, being the result the one shown in figure 7. Now, the system has to choose between N_1 and N_3, but trajectories are not able to decide. In this moment, another concept has to be introduced: agent pairing.

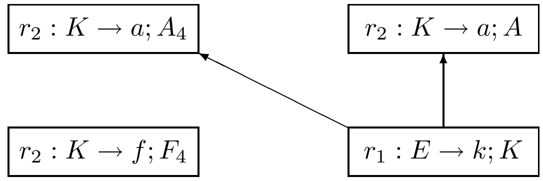

Figure 7. Turn-Taking Resolution by Trajectories

3.3. Agent Pairing

The notion of *agent pairing* can be explained as follows: in a conversation with more than two agents, if an agent N_i sends an ambiguous turn-taking, and two or more of the other agents fulfil turn coherence and the trajectories criterion, then the agent which the turn goes to is the one that can address its utterance to N_i.

Figure 8 shows the application of the method to resolve the ambiguity in figure 7. N_2 has been dismissed by trajectories. Agent pairing helps to choose between N_1 and N_3. By this method it is seen that, whereas the option of N_3 generates another ambiguity, N_1 addresses its utterance to N_4, replying the sequence generated by that agent.

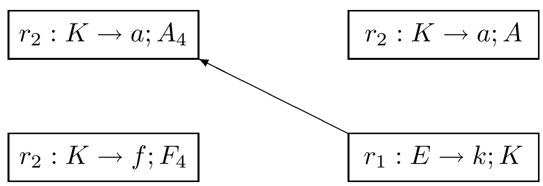

Figure 8. Turn-Taking Resolution by Agent Pairing

If the turn-taking assignation is not decided by these methods, then, for our cooperative (non collaborative) system, agents will start talking at the same time. In this scenario, overlapping arises.

4. Overlapping

Overlapping is a pragmatic phenomenon that arises due to an unsolved turn-taking in a conversation between non-collaborative agents. Overlapping describes the situation where several agents participate simultaneously for several steps in two strands of the same conversation. This phenomenon occurs during the development of the computation, never from the beginning. Nevertheless, all the agents are gathered again in a common conversation before the production finishes. From the moment overlapping starts, turn-taking can be developed in two or multiple lines. Table 1 shows an example on how a dialogue becomes overlapped during two steps to come back later to just one strand.

Overlapping is, thus, like a merging of dialogues, with an additional feature: it is reversible. This can be the basis for the description of split, an irreversible breaking up of a conversation in two or more productions where the communication channels between agents in different groups are closed.

Configuration	1	2	3	4	5	6	7	8
Speaker Agents	1	2	3	2	1 3	3 2	1	3

Table 1. A Derivation Scheme of Overlapping

5. Conclusions

The approach we have presented here offers a consideration of turn-taking as a dialogue game in pragmatics. Pragmatics, in our model, has to do with the structure of agents, their relationship (connection) and their interaction.

In the general framework referred above, this approach gives a simple explanation of turn-taking ambiguity, as well as a simple protocol to solve it. We suggest a list of elements that should be taken into account for turn-giving with an order of precedence, in a way that the algorithm can be completed in a sequential way.

Moreover, the model is also quite realistic, because it describes overlapping, which is the situation that arises when it is not possible to solve the ambiguity. Overlapping is produced only with non cooperative agents and it is not very much approached by computational models. However, it is a quite common phenomenon in every-day conversations.

We highlight that this algorithm is valid for the modelling of an important aspect in the evolution of the pragmatic context of conversations, given by the distribution of turns and the assignation of the agents that have to talk.

For the future, we suggest an improving of the algorithm including more detailed semantic, kinetic and pragmatic acts that can help to turn-management. Moreover, a deeper study of different aspects of bad configurations and failures of the system is needed, that can help to improve a formal theory of human computer interaction.

Finally, it seems this system is quite easy to implement, due to the simplicity of the formalism and the computational background of the agent theory we use. Achieving a valid and simple computational implementation of this disambiguation protocol for turn management is the major research line for the future.

References

BEL-ENGUIX G. and JIMÉNEZ-LÓPEZ M. (2005). ”Linguistic membrane systems and applications”. In G. Ciobanu, G. Păun and M. Pérez-Jiménez(eds.), *Applications of membrane computing*, Natural Computing Series : Springer, Berlin., p. 347-388.

BUNT H. and GIRARD Y. (2005). ”Designing an Open, Multidimensional Dialogue Act Taxonomy”. In C. G. . B. Gaiffe(ed.), *DIALOR'05, Proceedings of the Ninth International Workshop on the Semantics and Pragmatics of Dialogue. Nancy, June 9-11, 2005.*

CARLSON L. (1983). *Dialogue Games. An Approach to Discourse Analysis.* Reidel, Dordretch.

COULTHARD M., MONTGOMERY M. and BRAZIL D. (1981). ”Developing a Description of Spoken Discourse”. In M. Coulthard and M. Montgomery(eds.), *Studies in Discourse Analysis* : Routledge & Kegan Publishers, p. 1-50.

HAJDINJAK M. and MIHELIC F. (2004). ”Information-Providing Dialogue Management”. In *Text, speech and dialogue : 7th International Conference, TSD 2004 Brno, Czech Republic:*

proceedings, LNAI 3206, 595-602.

KOWTO J., ISARD S. and DOHERTY G. (1993). *Conversational Games Within Dialogue*. Rapport interne Research Paper HCRC/RP-31, Human Communication Research Centre, University of Edinburgh.

LITMAN D. and ALLEN J. (1990). "Discourse Processing and Common Sense Plans". In P. Cohen, J. Morgan and P. M.E(eds.), *Intentions in Communication*. Cambridge (Mass) : MIT Press.

POESIO M. and TRAUM D. R. (1997). "Conversational Actions and Discourse Situations.". In *Computational Intelligence*, 13(3), 309-349.

PRAKKEN H. (2005). "Coherence and Flexibility in Dialogue Games for Argumentation". In *Journal of Logic and Computation*, 15, 1009-1040.

REED C. and LONG D. (1997). "Collaboration, Cooperation and Dialogue Classification". In K. Jokinen(ed.), *Working Notes of the IJCAI97 Workshop on Collaboration, Cooperation and Conflict in Dialogue Systems*. Nagoya. p. 73-78.

SINCLAIR J. and COULTHARD R. (1975). *Towards an Analysis of Discourse: The English Used by Teachers and Pupils*. Oxford University Press, Oxford.

TRAUM D. and HINKELMAN E. (1992). "Conversation Acts in Task-Oriented Spoken Dialogue". In *Computational Intelligence*, 8 (3), 575–599.

TRAUM D. and LARSSON S. (2003). "The Information State Approach to Dialogue Management". In J. V. Kuppevelt and R. Smith(eds.), *Current and New Directions in Discourse and Dialogue* : Kluwer.