Turn-taking in Discourse and its Application to the Design of Intelligent Agents

Toby Donaldson and Robin Cohen
Department of Computer Science
University of Waterloo
tjdonald@neumann.uwaterloo.ca, rcohen@watdragon.uwaterloo.ca

Introduction

We are interested in studying turn-taking in dialogue, in particular human-computer discourse. Humans and computers can communicate in any number of ways, such as through an operating system command line, a direct manipulation interface, or even natural language. *Turn-taking* is a basic fact of life that interacting agents must deal with: who should say what, and when? In some situations, turn-taking is a simple matter of following a predefined script, such as reciting wedding vows, or following the World Wide Web HTTP protocol. However, in much human-human dialogue, a more flexible and dynamic form of turn-taking is needed since it is impossible to always predict how a conversation will progress.

In natural language dialogue systems, it is often assumed that explicit turn-ending signals will be given by the conversants. For example, such systems typically require that you type a sentence and then press <return>, which is a signal to the computer that it is its turn to talk. However, explicit turn-ending signals are not practical in other domains. For example, in a graphical user interface, the user points and clicks with a mouse, while the computer responds by opening windows, moving objects, etc. In this case, the user does not give explicit signals as to when she has finished "speaking". As in human-human dialogue, it is necessary to carefully observe the other conversant and watch for relevant points to take a turn as they arise. To do this effectively, a general theory of turn-taking is needed to determine *why* a turn should be taken, *when* it should be taken, and how turn-taking goals interact with an agent's other goals.

With respect to agents in general, we believe that since agents may engage in interaction that can be treated as a kind of conversation, knowing how and why to communicate with other agents is a necessary ingredient for success. In particular, agents meant to interact with a human must be able to deal with people in a natural and flexible manner. Below we outline a general framework for turn-taking that could be applied to any dialogue situation involving intelligent interacting agents.

Three Steps to Good Turn-Taking

In a conversation, a turn-taking agent must decide such issues as *why* to take a turn, *when* to take a turn, and *how* to take a turn. We use a basic three-part model for agent turn-taking. First, an agent needs a reason or motivation for taking a turn. This reason then triggers the adoption of a turn-taking goal, and, lastly, the turn is executed at an appropriate point in the conversation. Each of these steps is elaborated upon below.

Motivation

Ultimately, an agent can take a turn for any reason, but a number of interesting and important cases stand out. The most concrete kinds of motivations are ones related to anomalous or unexpected situations that might require a turn to be taken immediately (i.e. an interruption). Such motivations include recognizing an inconsistency in a conversant's knowledge base, dealing with a failed stereotyped assumption, or resolving plan-related ambiguities.

In this abstract we will focus mainly on the goal adoption phase of the turn-taking process, but to illustrate the over-all model we introduce a hypothetical course advising system called ARNIE that helps students choose courses. Suppose the student asks the system the following question:

(1) Student: When does CS492 meet?

ARNIE recognizes this as a valid question, and so this causes in ARNIE a motivation to provide an answer to the question, which has propositional content p; effectively, this amounts to ARNIE adopting *simple-answering-goal*(p) (discussed in the next section). If the question is not valid, then a different motivation will be triggered:

(2) Student: When does CS592 meet?

In this case, CS592 does not exist, so ARNIE realizes the student likely has some misconception and a motivation to correct this error is triggered. Thus, a precondition to answering the question is to first correct this misconception. In this particular case, *two* goals are adopted and one is a subgoal of the other, and thus ARNIE has a preference for the order in which satisfaction of the goals should be attempted. In other cases, it might not be so clear in which order goals should be dealt with; handling multiple goals is an important issue, and while we do not deal with it directly in this paper, it will arise again in Section 4.

Goal Adoption

An important aspect of rational behaviour is not giving up one's goals too soon. The idea of being *committed* to one's goals has received attention in both AI and philosophy (e.g. Cohen & Levesque (1990); Georgeff & Rao (1995); Bratman (1987)). Cohen & Levesque (1990) define a *persistent goal* to be a goal that an agent keeps until either the goal is achieved, or the agent comes to believe the goal can never be achieved. We have defined a variation of persistent goals, called *time-bounded persistent goals* (Donaldson & Cohen 1996), which are more sensitive to time than Cohen and Levesque's persistent goals, and thus, we believe, more appropriate for modelling the kinds of turn-taking goals that can arise in discourse. A time-bounded persistent goal is kept until either the agent achieves the goal, or believes the goal cannot be satisfied in some time-interval of relevance. More specifically, a time-bounded persistent goal to achieve ϕ in time T is defined as follows:

Bounded-persistent-goal(ϕ,T)
While: $\textit{simple-goal}(\phi)$
Adopt-when: $B(\textit{holds}(B\neg\phi,\textit{some-head-of}(T)))$
Drop-when: $B(\textit{holds}(B\phi,\textit{some-tail-of}(T)))$
$B(\textit{holds}(B\neg\phi,\textit{some-tail-of}(T)))$
$B(\textit{after}(T,\textit{now}))$

Bp means that the agent believes proposition p to be true. The notation *some-head-of*(T) and *some-tail-of*(T) is shorthand for unspecified intervals of T that touch one end of T. For example, $B(\textit{holds}(B\neg\phi,\textit{some-head-of}(T)))$ means the same as the expression $\exists T_H.\textit{starts}(T_H,T)\wedge B(\textit{holds}(B\neg\phi,T_H))$. A goal is *adopted* when all of the conditions on the While list and Adopt-when list hold, and none of the conditions on the Drop-when list hold. This means that the agent will adopt the bounded persistent goal to achieve[1] ϕ in time T while the agent has a simple (i.e. non-persistent) goal to achieve ϕ, and the agent believes $\neg\phi$ holds in some initial sub-interval of T. An adopted goal is dropped when at least one of the While conditions or at least one of the Drop-when conditions hold. Thus, an agent will drop a bounded persistent goal if the agent believes ϕ holds in some sub-interval that ends T, or if the agent believes $\neg\phi$ holds in some sub-interval that ends T, or if the agent believes T is in the past, or if the agent believes the simple goal to achieve ϕ no longer holds.

It should be pointed out that Cohen and Levesque's intent is to provide a *specification* for how a rational agent ought to behave. They do not claim that their formalism is actually what should be used in an implementation. Our goal is not to provide a logical specification of an agent, but instead to develop a computationally feasible framework for implementing an intelligent turn-taking agent.

To give a flavour of how time-bounded persistent goals work, consider ARNIE and the student. The student first asks:

Student: When does CS492 meet?

A question can be treated as a motivation for adopting a time-bounded persistent goal to answer the question:

Persistent-answering-goal(α, p, T)
While: $\textit{simple-answering-goal}(p)$
Adopt-when: $B(\textit{holds}(B\neg\alpha,\ \textit{some-head-of}(T)))$
Drop-when: $B(\textit{holds}(B\alpha,\ \textit{some-tail-of}(T)))$
$B(\textit{holds}(B\neg\alpha,\ \textit{some-tail-of}(T)))$
$B(\textit{after}(T,\ \textit{now}))$

This means ARNIE has a time-bounded persistent goal to achieve α within time interval T; more specifically, α is the state where the question, which has propositional content p, has been answered. In this case, it is reasonable to expect α to be a simple response, since the student's question is asking for a basic unit of information about a particular course; more complicated questions might require more complex clarification or elaboration, and so in such cases it would be reasonable to expect α to refer to an entire sub-dialogue. Now, we are working under the common assumption that dialogue is an essentially *collaborative* activity

[1]Cohen and Levesque distinguish between *achievement* goals, where the agent tries to make a false proposition true, and *maintenance* goals, where an agent tries to keep an already true proposition true. We will only consider achievement goals in this paper; however, maintenance goals are ultimately important to a rational agent, and the interaction between achievement goals and maintenance goals is an interesting topic for future research.

(Clarke & Wilkes-Gibbs 1990; Grosz & Sidner 1990; Chu-Carroll & Carberry 1994), and so T cannot yet be completely instantiated until there is some indication from the other conversant whether the response was understood. Since, in this case, ARNIE has no other goals that should take precedence, ARNIE acts to try to achieve α by generating a cooperative response:[2]

(3) ARNIE: Tuesdays, 1pm-3pm. Note that you must be pursuing a math degree to take CS492 for credit.

If the student responds with "Okay, thanks", then ARNIE can drop its persistent goal to answer the question. If instead the student responds with something like "What?", then ARNIE should persistent with its goal of trying to answer the student's question by clarifying its response. However, ARNIE should not necessarily try to answer the student's question forever. If the student's misunderstanding persists too long, then ARNIE may well run out of strategies for dealing with the problem, and so instead of repeating methods that have already failed once, ARNIE ought to tell the student he can't do anything more to help him. This amounts to ARNIE choosing the endpoint of T.

This example shows two ways a bounded persistent goal can be dropped: first, by achieving the goal, and, second, by coming to believe the goal cannot be achieved within T. The third condition, believing that T is in the past, arises in situations where the speaker changes the topic in mid utterance. For example, suppose the dialogue continues like this:

(4) Student: Are there any pre-requisites? Oh, wait a minute, I need the credit. What computer courses are open to science students?

Just after the first sentence, ARNIE adopts a goal to answer the student's question about pre-requisites. However, the student then changes his mind, retracting his original question and asking another. Given that CS492 is no longer the topic of conversation, it would be inappropriate for ARNIE to now answer the student's first question. Instead, ARNIE realizes that the time of relevance for this goal has past, and so drops it. Of course, there may be cases where a speaker changes the topic in mid-turn, and the listener decides it is important enough either to interrupt the speaker in mid-turn, or shift the focus back to the relevant topic of conversation.

Ambiguous Plans As another example, we consider how an actual advice-giving system could benefit from time-bounded persistent goals. van Beek et al. (1993; 1994)'s advice-giving system initiates clarification subdialogues to resolve relevant ambiguities in plan recognition. The overall goal of the system is to inform the user about information in a specific advice-giving domain (examples are drawn from both the course-advising and the cooking domain). The system works by performing plan recognition on a user's question; Kautz's plan recognition system is used (Kautz 1990), so the output is a set S of all possible plans that the user could be following. If S is ambiguous, and the ambiguity matters to the formulation of an advice-giving response[3] (S is *relevantly ambiguous*), then a clarification dialogue is entered into with the user. As it stands, the system keeps the goal of clarification until S is no longer relevantly ambiguous. The system's behaviour can be modelled like this:

Clarification-goal′(S)
While: *inform-goal*(u)
Adopt-when: S is ambiguous
Drop-when: S is no longer relevantly ambiguous

A clarification subdialogue consists of a series of yes/no questions asked by the system. Each individual question could be modelled with a goal:

Question-goal(S)
While: *Clarification-goal′*(S)
Adopt-when: there is an individual question to ask about S
Drop-when: the question has been asked

As is clear from these goals, van Beek et al.'s system will only stop trying to clarify S if either *inform-goal*(u) is not the case, or the ambiguity in S has been resolved or no longer matters. Here is a typical dialogue that this system currently handles:

U: Is it ok to make marinara sauce?
S: Are you planning to make a meat dish?
U: Yes.
S: Yes, you can make marinara sauce, but your guest is a vegetarian, so you should avoid meat.

When the dialogue begins, the system adopts *inform-goal*(u). Then after the user's initial question, the system, unclear about the user's possible plans (S is relevantly ambiguous), asks a clarifying question about the main course. Once the question has been asked, the question goal is dropped. After the user's response, the system can now give its final advice and drop its

[2]See Joshi et al. (1984) for details on cooperative response generation.

[3]This is determined by considering the possible faults of plans, towards the production of a cooperative response.

clarification goal, since S is no longer relevantly ambiguous.

Using time-bounded persistent goals and the notation introduced in the previous example, van Beek et al.'s system can be generalized to allow more flexible control of the dialogue:

Clarification-goal(S)
While: *inform-goal*(u)
Adopt-when: $B(\textit{needs-clarification}(S))$
Drop-when: $B(\textit{no-longer-needs-clarification}(S))$

Persistent-interruption-goal(α, S, T)
While: *Clarification-goal*(S)
Adopt-when: $B(\textit{holds}(B\neg\alpha, \textit{some-head-of}(T)))$
Drop-when: $B(\textit{holds}(B\alpha, \textit{some-tail-of}(T)))$
$B(\textit{holds}(B\neg\alpha, \textit{some-tail-of}(T)))$
$B(\textit{after}(T, \textit{now}))$

This definition shows that an advice-giving system may explicitly abandon its clarification goal not only in the case where the clarification has been achieved (as handled by van Beek et al.), but also when the system believes it will be unable to finish the clarification dialogue or when it believes the time of relevance has passed.

We now give examples to show how the various drop conditions can lead to more complex kinds of dialogues. First, consider the following variation of the cooking dialogue:

U: Is it ok to make marinara sauce?
S: Yes.
U: Now, I know I can't make meat because of the vegetarian guest, but is whole wheat spaghetti ok?
S: Yes.

Here, we assume that the system decides not to immediately enter into a clarification dialogue, perhaps because it expects a clarification to be forthcoming, or that now is not the best time to interrupt.[4] On the user's next turn, the ambiguity happens to be explicitly resolved. This causes the system's persistent-interruption goal to be dropped, since the reason for the interruption has been resolved.

Next, consider the following dialogue between a student and a course advisor:

U: Can I take cs150?
S: Have you already taken cs170?
U: What is cs170?
S: It is a basic computer literacy course.
U: That sounds good. Can I take cs170?
S: Yes.
U: Now, about math188 ...

Here, the system never resolves the ambiguity for which it initiated the clarification dialogue, and the user changes the topic. On a subsequent turn, the system could return to address cs150, but if it is not really necessary then such a turn will likely be just an annoyance to the user. Instead, the system drops its persistent goal to interrupt the user because it now believes it cannot perform the clarification subdialogue in the time interval where it would be relevant. This does not mean that the system believes it can never clarify cs150 for the user, but that it cannot finish the clarification dialogue within time T, the time of relevance for cs150.

Finally, consider the case where the system drops a time-bounded interruption goal because it believes the conversation has ended — perhaps the user leaves the room to go do something else. Assuming that the system has other things to do, it would drop its goal to interrupt the user, even though it might believe that the user will at some time return, and it could then interrupt the user. This corresponds to the system believing the conversation has ended.

Turn Execution

When taking a turn, an agent must consider two basic sources of information: its own beliefs, desires, and intentions (BDIs), and the BDIs of the other conversant. Since agents only have direct access to their own BDIs, the BDIs of the conversant must be inferred through their actions. An agent should strive to take a turn at a *transitionally relevant point* (TRP), which is essentially a time in the conversation where it is mutually agreeable to both conversants for a turn-shift to occur (Oreström 1983). In human-human dialogue, potential TRPs are typically implicitly signaled by a number of actions: change in the volume of the speaker's voice, change in intonation, pausing, uttering of a grammatically complete phrase, focus shifts, gestures etc. All of these indicate a potential TRP, and the more of these features that occur at the same time, the more likely that that time is a TRP. In Section 4, we suggest a uniform way for dealing with these TRP signals and an agent's turn-taking goals.

It is our intention to develop a specification for turn execution which incorporates a variety of these features simultaneously. As an example, consider pauses. Pauses are computationally attractive for two main reasons. First, in the context of spontaneous speech, a number of psycholinguistic experiments have been

[4] van Beek et al.'s system begins a clarification dialogue as soon as it believes clarification is necessary, which may not always be the best strategy.

done showing correlations between pauses and mental activity (Goldman-Eisler 1968). Second, pauses naturally occur in domains besides natural language speech; for example, in a graphical user interface, an absence of input from the user can be treated as a pause, which suggests that it might be possible to apply knowledge about pauses in human-human speech to human-computer interface interaction.

Deciding the length of a pause that may signal the end of a turn depends in part upon the particular speaker, and this presents an interesting problem in dynamic user modelling. Goldman-Eisler (1968) reports a number of experiments on pauses in spontaneous speech, showing correlations between hesitations, speech planning, and grammatical boundaries. Using these results, we can get a reasonable default set of values (e.g. a pause of 2 seconds or more indicates a potential turn-ending signal) which can then be adjusted as the conversation continues.

Related Work

We briefly discuss some of the work on turn-taking from computational linguistics which is relevant to the design of communicating agents.

Sidner (1994) defines a descriptive language that allows conversants to negotiate over the meaning of their utterances. Sidner's language treats conversant's utterances as proposals, counter-proposal, acceptances, etc. This characterization of discourse might provide some insight into how to manage a set of time-bounded persistent goals. Moreover, time-bounded persistent goals may help to elucidate the conditions under which various discourse moves may be taken, therefore clarifying how Sidner's language can be effectively introduced into processing algorithms.

Chu-Carroll and Carberry (1995) give a computational strategy for initiating information-sharing subdialogues. When their system is uncertain about whether or not a proposal from the user should be accepted or rejected, an information-gathering subdialogue is entered into. Their system also allows subdialogues within subdialogues. This work therefore also provides us with cases for our study of turn taking: it is worth studying how information-gathering subdialogues can be treated as interruptions and how the use of time-bounded persistent goals can provide for a richer set of possible dialogue structures.

Relevant computational work on resource-bounded language generation has been done by Carletta et al. (1993) and Walker (1993). Carletta et al. are partly interested in the cognitive plausibility of language production, and treat it as a "high risk" activity, where it is acceptable for a speaker to make a mistake because it is often possible to quickly repair this mistake. Their notion of anytime language production is very similar in spirit to the local repair CSP approach we suggest in the next section.

Discussion: A Constraint Satisfaction Approach to Agents

In this section, we discuss some preliminary ideas about a constraint-based approach to managing multiple goals and multiple influences for turn execution.

Turn-taking raises a number of general questions about agents. Abstractly, the turn-taking problem is about when to take a turn in an interaction with another intelligent agent; a turn-taker must weigh two sets of knowledge: 1) its own BDI's, and 2) the actions that indicate the relevant BDI's of the other conversant. A uniform way to handle these two sources of knowledge is to treat taking a turn as solving a single constraint satisfaction problem (CSP). The variables could be divided into two sets: the *goal* variables $V_1 \ldots V_n$, representing the n turn-taking goals the agent can consider, and the *pragmatic* variables such as V_{volume}, $V_{intonation}$, V_{pause}, V_{focus}, etc. For $V_1 \ldots V_n$, constraints can be based on the relative importance of the propositional content of the goal (e.g. how bad would it be if the turn was never taken?), logical consistency, linguistically motivated coherence constraints, such as following standard focus/topic pattern progressions. For example, in utterance (2), two turn-taking goals are triggered, with the constraint that one is a subgoal of the other, and so one must be satisfied before the other. The pragmatic variables can be related by constraints based on linguistic data; for example, in spoken communication if the volume of the speakers voice is going from high to low, and they are nearing a grammatical boundary, then a constraint between the volume variable and the grammar variable could be said to be satisfied.

Along with determining what the relevant constraints are, part of the cost of this uniformity of representation is that the goals that can be represented are now finite in number, and reasoning with these goals is not the same as with a logical representation simply because CSP variables are not logical propositions. However, we believe the CSP representation offers a number of practical advantages that cannot be (straightforwardly) achieved with logical representations. Since conversation is a dynamic activity where new information is presented in an incremental fashion, *local repair* techniques are an appropriate method for solving such CSPs (Minton *et al.* 1992). In general, such methods solve a CSP by starting with some possibly inconsistent instantiation of *all* variables, and proceed to find a consistent solution by *repairing* the value of single variables according to some heuristic (e.g. the min-conflicts heuristic in Minton et al.). Local repair methods appear to be ideally suited to dialogue situations, since they work by modifying fully instantiated sets of variables, and thus can handle changes in constraints, additions of new variables, and changes in domains more easily than traditional constructive CSP solution methods. Furthermore, since all variables are instantiated at all times, local repair methods are appropriate for *anytime* computation, i.e. the algorithm can be stopped at any time, and it will have some reasonable solution to offer. In dialogue, unexpected interruptions might require a conversant to speak at a time when they are not completely ready, and so an anytime algorithm would seem to be needed.

A key issue we are interested in studying is how a turn-taking agent should manage multiple turn-taking goals. Typically, goal management is treated as a kind of planning problem; however, deciding in what order to take turns is more like a *scheduling* problem. Roughly, planning problems require finding some number of actions that transform an initial state to a goal state; the number of actions required is not known in advance. Scheduling, on the other hand, it is known in advance how many actions need to be scheduled. While in general both planning and scheduling problems can be very difficult, Minton et al. (1992) have shown local repair techniques can be extremely successful in scheduling problems.

Conclusion

We a believe that a theory of turn-taking would be useful not only for natural language dialogue systems, but for any intelligent agent that may interact with a person, such as interface agent or a "personal assistant" agent. Our general framework for modelling turn-taking goals is based on a three-step model that goes from motivation, to goal-adoption, to turn execution. Our development of time-bounded persistent goals is a valuable representation which allows for dialogues with more complex clarification and interruption structures than have previously been addressed. We suggest that CSPs, based on local repair solution methods, provide a uniform and computationally attractive model for developing turn-taking algorithms.

References

Bratman, M. 1987. *Intentions, Plans, and Practical Reason.* Harvard University Press.

Carletta, J.; Caley, R.; and Isard, S. 1993. A system architecture for simulating time-constrained language production. Technical report, Human Computer Research Centre, University of Edinburgh. Technical report rp-43.

Chu-Carroll, J., and Carberry, S. 1994. A plan-based model for response generation in collaborative task-oriented dialogues. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, 799–805. AAAI.

Chu-Carroll, J., and Carberry, S. 1995. Generating information-sharing subdialogues in expert-user consultation. In *Proceedings of IJCAI '95*, 1243–1250.

Clarke, H., and Wilkes-Gibbs, D. 1990. Referring as a collaborative process. In Cohen, P. R.; Morgan, J.; and Pollack, M. E., eds., *Intentions in Communication.* The MIT Press, Cambridge: System Development Foundation Series. chapter 23.

Cohen, P., and Levesque, H. 1990. Intention is choice with commitment. *Artificial Intelligence* 42:213–261.

Cohen, R.; Schmidt, K.; and van Beek, P. 1994. A framework for soliciting clarification from users during plan recognition. In *Proceedings of the 4th International Conference on User Modeling.*

Donaldson, T., and Cohen, R. 1996. Time-bounded persistent goals and their role in discourse. Unpublished.

Georgeff, M., and Rao, A. 1995. The semantics of intention maintenance for rational agents. In *Proceedings of IJCAI-95*, 704–710.

Goldman-Eisler, F. 1968. *Psycholinguistics, Experiments in Spontaneous Speech.* Academic Press.

Grosz, B., and Sidner, C. 1990. Plans for discourse. In Cohen, P. R.; Morgan, J.; and Pollack, M. E., eds., *Intentions in Communications.* MIT Press. chapter 20, 416–444.

Joshi, A.; Webber, B.; and Weischedel, R. 1984. Living up to expectations: Computing expert responses. In *Proceedings of AAAI-84*, 169–175.

Kautz, H. 1990. A circumscriptive theory of plan recognition. In Cohen, P. R.; Morgan, J.; and Pollack, M. E., eds., *Intentions in Communication.* MIT Press.

Minton, S.; Johnston, M.; Philips, A.; and Laird, P. 1992. Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence* 58:161–205.

Oreström, B. 1983. *Turn-taking In English Conversation.* CWK Gleerup.

Sidner, C. 1994. An artificial discourse language for negotiation. In *Proceedings of AAAI 94*, 814–819.

van Beek, P.; Cohen, R.; and Schmidt, K. 1993. From plan critiquing to clarification dialogue for cooperative response generation. *Computational Intelligence* 9(2):132–154.

Walker, M. 1993. *Informational Redundancy and Resource Bounds in Dialogue.* Ph.D. Dissertation, University of Pennsylvania.