Title
Making Conversation Flexible

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 11(0)

Authors
Turner, Elise H.
Cullingford, Richard E.

Publication Date
1989

Peer reviewed

Making Conversation Flexible [1]

Elise H. Turner and Richard E. Cullingford
School of ICS
Georgia Institute of Technology

Abstract

The goals of the speakers are the motivating force behind a conversation. The differences in these goals, and their relative priorities, account for many of the differences between conversations. In order to be easily understood, however, the resulting conversation must be constrained by the language conventions shared by speaker and hearer. In this paper we describe how the use of schemas for conversational control can be made flexible by integrating the priorities of a system's goals into the process of selecting the next utterance. Our ideas are implemented in a system called JUDIS (Turner & Cullingford, 1989), a natural language interface for an advice-giving system.

A characteristic feature of conversation is *flexibility*; the "topic" or "point" being pursued shifts unpredictably as the conversation proceeds. Clearly, the conversants adopt a series of communicative goals (Appelt, 1985; Cohen & Perrault, 1979; Grosz, 1977; Hobbs & Evans, 1980) and work cooperatively to achieve these goals. It is clear as well that an important component of conversation is convention (McKeown, 1985; Reichman, 1985). That is, we not only plan our utterances in the presence of perceived intentions, but we also, in a sense, recall how conventional conversations were conducted in the past to achieve similar ends. This paper describes an implemented conversational system, JUDIS, which applies a knowledge structure, the *conversation MOP*, to provide a means of combining intention and convention in discourse. The integration of problem solving goals and linguistic constraints has been shown to improve comprehension (Carberry, 1986; Litman & Allen, 1987) and to be important in determining discourse structure (Grosz & Sidner, 1986). Here, we focus on how these two essential aspects of conversation can be united to provide flexible generation. We present an example of the system processing a typical conversation with the user during a common kind of naturalistic conversational task, commonsense problem solving.

JUDIS

JUDIS (Julia's DIscourse System) provides the natural language interface for Julia (Cullingford & Kolodner, 1986), a highly interactive, advice-giving system being developed at the Georgia Institute of Technology. Julia is designed to perform such commonsense tasks as giving directions, advising consumers, and planning meals, its current task. It is comprised of several subsystems which include: a *from-scratch planner*, JUPITER (Hinrichs, 1988), which plans a meal using traditional methods from artificial intelligence; a *case-based reasoner*, JULIANA (Shinn, 1988), which uses previous experience in meal-planning to help solve the current problem; and the natural language interface, JUDIS (Turner & Cullingford, 1989). JUDIS handles conversational control, focusing on what should be included in the dialogue and when it should appear. This system uses an existing software system, the NLPTool (Cullingford, 1986), to generate English utterances from concepts and to analyze the utterances of the user.

[1] This research has been funded by NSF Grant IST-8608362. Thanks to Richard Billington, Janet Kolodner and Roy Turner for their helpful comments on earlier drafts of this paper.

J1: Would you like to serve guacamole as an appetizer?

U1: Yes.

J2: Would you like chili as a main course?

U2: Would we have to serve it in bowls?

J4: Yes.

J5: Do you have enough bowls?

U3: Yes.

J6: I remembered a meal where one of the guests couldn't eat guacamole because she was allergic to tomatoes.

J7: Do you know if any of your guests are allergic to tomatoes?

U5: No, they aren't.

Figure 1: Dialogue with Julia.

When the problem solving systems of Julia need to communicate with the user, they send a request to JUDIS in the form of a goal. JUDIS must respond to the request in a way that not only is understandable to the user, but also takes into account the urgency of the request, reflected by the goal's priority. Sometimes the request is so unimportant to the problem solver that it can be left out of the dialogue or brought into the conversation at a more natural time. Sometimes, JUDIS must interrupt the conversation to handle a request, such as when a piece of information is needed to allow a problem solver to continue an important task. Other times the request relates to a previous topic of conversation and JUDIS must interrupt the established order of the conversation and refer back to an earlier topic. Figure 1 shows a portion of an actual dialogue with Julia where such a request is handled in J6 – J7. To manage such infrequent requests effectively, JUDIS is guided by both the conventions of conversation and the needs of Julia.

REPRESENTING DISCOURSE STRUCTURE

JUDIS represents its knowledge of discourse structure in *conversation MOPs* (Kellermann et al., 1989; Turner & Cullingford, 1989). MOPs, or *memory organization packets* (Schank, 1982) are schematic structures used to organize long-term, conceptual, episodic memory. Conversation MOPs are much like other rhetorical schema (e.g., McKeown, 1985; Reichman, 1985). Each MOP contains an associated goal or goals which is achieved by the actions, or episode, specified in the MOP. The episode is divided into *scenes* which can be shared with other MOPs. An example of a MOP, one for conducting the problem solving portion of a conversation between a client and a caterer is shown in Figure 2. This MOP is itself a scene in the "complete-conversation-MOP", and the MOP's scenes are also MOPs. They are topic MOPs which in turn include the scenes "topic switch", "discussion", and "close topic".

In JUDIS, scenes can be either *executable acts* or MOPs. The scenes in MOPs can be either *mandatory scenes*, or *optional scenes* which are not necessary for the execution of the MOP but are often associated with it. The scenes may be actual utterances in the dialogue, or they may be other actions associated with those utterances. These other actions describe how an utterance affects the knowledge of the conversants. For example, a parse includes actions to update the knowledge base of the hearer, and a topic switch contains actions to update the focus of the dialogue. Although such actions are not fully implemented in the current version of JUDIS, those that affect conversational control have been included in the appropriate MOPs.

MOPs are formed when episodes are stored in a dynamic memory (Kolodner, 1984; Schank, 1982). When adding an episode, dynamic memory uses the salient features of the episode, called *predictive indices*, to find a place for it in memory. When this happens new MOPs are created which contain *generalized episodes* that reflect the similarities in the episodes.

Caterer's General MOP

goal: plan-meal
characters: caterer, customer
part-of: complete-conversation-MOP
predictive-features: problem-strategy
indices: problem solving strategy/chronological order → chronological-order MOP
problem solving strategy/main courses first → main-first MOP
activity: general info, dessert, appetizer, main-course
mandatory-scenes: general-info, dessert, appetizer, main-course

Chronological Order MOP

sequence of events: ((seq general-info appetizer) (seq appetizer main-course) (seq main-course dessert))

Main First MOP

sequence of events: ((seq general-info main-course)(seq main-course dessert)(seq main-course appetizer))

Figure 2: Caterer's MOP

Specializations of a generalized episode can be found by treating salient features that differ from the generalized episode's as predictive indices. For example, the "caterer's general MOP" has two specializations, each only differing from the more general episode because a scene ordering is specified (the only slot of these MOPs shown in Figure 2). Because the "caterer's general MOP" is associated with problem solving, its specializations are indexed by problem solving strategies such as planning the meal in chronological order or considering the main course first. The specializations for other MOPs may use different indices that are important to their purpose. For example, the topic switch MOP has specializations that relate to whether or not the scene has executed and the kind of change in topics that will take place. It also includes a specialization with no utterance that is indexed by the conditions that allow an explicit topic switch to be omitted from the conversation.

Knowledge of the discourse structure is important to JUDIS and cannot be built during the conversation, as in (Grosz & Sidner, 1986), for two reasons. First of all, like (McKeown, 1985), the schemas can be used as stored plans to make generation easier. Secondly, knowledge of how the discourse is likely to proceed allows JUDIS to decide when it can defer a system goal and wait for its topic to arise naturally in the conversation.

CONTROLLING CONVERSATION IN JUDIS

Conversation MOPs provide JUDIS with knowledge of conventional conversations. However, many conversations do not follow the norm, often due to the needs of the conversants. So, instead of simply following a schema for the current portion of the dialogue, JUDIS combines information about the priorities of its goals and the normal course of conversation to determine which MOP to follow at execution time. We will discuss how a MOP is selected for execution after a brief overview of processing conversation MOPs in JUDIS.

Overview

When a user begins a problem solving session with Julia, JUDIS knows that it is expected to hold a conversation with the user that includes planning a meal. JUDIS starts with the complete conversation MOP and finds the specialization of its scenes that fit the current situation. These

MOPs, called *active MOPs*, form the template for a conventional conversation in which JUDIS maintains the initiative; in other words, they form a high-level plan for the conversation.

JUDIS uses the expectations provided by the active MOPs to help satisfy the requests of the problem solvers within the confines of a coherent conversation. When JUDIS receives a request, it searches memory for a MOP to achieve the goal. It then tries to fit this MOP into the overall conversation by searching up part-of links for an active MOP which may, directly or indirectly, contain the new MOP. Since the MOPs represent conventions in conversation, finding the containing episode allows JUDIS not only to satisfy a goal, but to do it in such a way that is easy for the user to understand. The new MOP and the episodes on the path between it and the active MOP then become active MOPs. This mechanism allows JUDIS to observe conventions in conversation and maintain a coherent dialogue without explicitly planning to achieve these goals, as in (Appelt, 1985; Litman & Allen, 1987).

JUDIS selects a MOP for execution using the algorithm described below. If the selected MOP is an executable act that has been executed by JUDIS or can be inferred to have been executed by the user, it is marked "executed". A MOP is also marked "executed" if it is a MOP that has had all of its mandatory scenes executed. When a scene is executed, the episode which contains that scene is considered an *executing MOP*. The executing MOPs include not only the episode that this scene was chosen to help satisfy, but also any other MOPs which share that scene. These other MOPs are said to be *serendipitously executing*.[1] Serendipitously executing MOPs indicate options that are not part of JUDIS' original plan. For example, JUDIS may know that the problem solvers are following the "chronological order" strategy and select the corresponding "chronological order MOP", shown in Figure 2, as an active MOP. However, if only the "general-info" scene has been executed and one of the problem solvers has a question about the main course, JUDIS can pursue the serendipitously executing "main-first MOP" and ask the question, or it can choose to continue following the active "chronological order MOP".

Selecting a MOP for Execution

To determine which MOP should be executed, JUDIS must consider the conventions for the conversation and the goals of the system. In order to combine the effects of intention and convention, JUDIS relies on an activation metaphor. Activation is divided into two types: *goal-based activation* which reflects the importance of a MOP to the work of the system, and *MOP-based activation* which indicates how well a MOP fits the schema for conventional conversation.

Goal-based Activation

The goal-based activation of a MOP is determined by the priorities of the goals that the MOP can help to achieve. The goals associated with the MOP are considered to be achieved when the MOP is executed, so these goals contribute to the goal-based activation. Also, mandatory scenes inherit goals from their containing episodes because the execution of all mandatory scenes is necessary for an episode to be executed. Finally, any goals of scenes that are contained in the MOP and are not subsumed by the MOP's associated goals contribute to the goal-based activation. For example, if a discussion of a particular topic will include giving the user a useful piece of information and suggesting the dish to be served for a course, the priorities of these goals will be added to the goal-based activation for the topic. By allowing goal-based activation to accrue to the containing episode, JUDIS can move to the portion of the dialogue which satisfies a

[1] A MOP is not considered to be serendipitously executing unless its scenes have been executed in the order specified by the MOP.

large number of goals, even if none of the individual goals have a high priority.

MOP-based Activation

MOP-based activation is afforded to the MOPs which follow some rhetorical convention that is at play in the conversation. There are two types of MOP-based activation, reflecting the two types of MOPs, active and executing, that influence conversation. *Active MOP-based activation* is passed from the highest level active MOPs, usually the MOP for the overall conversation. Since active MOPs indicate the system's plan for the conversation, the value assigned to active MOP-based activation should express the extent to which JUDIS wishes to adhere to its original plan and to remain in control of the conversation. *Executing MOP-based activation* is passed from all executing MOPs. Since all MOPs that are executing represent an accepted schema for conversation, its value represents the system's commitment to following conversational conventions.

MOP-based activation is given to the appropriate MOPs and any of their scenes that are ready to execute. A scene is ready to execute if all of the mandatory scenes which precede it have executed and if no scene which it precedes has been executed. Optional scenes affect the way activation is passed depending on the type of MOP-based activation. For active MOP-based activation, an optional scene with an active associated goal is treated like a mandatory scene. Here, the optional scene can be executed to help JUDIS achieve a goal and so can be seen as part of the system's plan. However, optional scenes do not affect how well a dialogue adheres to convention, so those which are ready to execute and the mandatory scenes which they precede receive executing mop-based activation. When activation is passed in this way, optional scenes require a higher goal priority than mandatory scenes to be included in the dialogue. This is an extension of the work of McKeown (McKeown, 1985), where optional scenes are selected based on the focus of the dialogue and the availability of the knowledge the scene is to contain. In JUDIS, an optional scene that meets these criteria may be omitted from the dialogue if it is not important enough for inclusion. This allows JUDIS to leave out information that is often relevant, but may not be pertinent to the current situation.

Each time JUDIS needs to select a MOP for execution it chooses the MOP with the highest activation. If the selected MOP is not an executable act, JUDIS must choose one of its scenes to execute. JUDIS uses the method described above with the selected MOP serving as the sole active and executing MOP to pick a scene to execute.

INTEGRATING REQUESTS INTO A COHERENT DIALOGUE

The processing for conversational control described above allows JUDIS to produce a variety of conversational meanderings that are motivated by the goals of the system but moderated by the conventional structures of discourse. JUDIS uses the priority of system goals as the primary factor in determining what to say next, instead of relying mainly on pre-described discourse structures or rules of focus (Grosz, 1977; McKeown, 1985; Reichman, 1985; Sidner, 1983; Webber, 1983).[2] However, since JUDIS places requests from the problem solvers in conventional discourse structures, it is able to integrate them into a coherent dialogue that is also responsive to the import and urgency of the system's goals.

An example of a request forcing an interruption in the expected course of the dialogue is shown in Figure 1. In the interest of brevity, we will address only those active MOPs which play an interesting role in JUDIS' decision and ignore serendipitously executing scenes and component

[2] A useful extension to JUDIS would be to include focus to help determine when an opportunity may be lost to handle a request with ease. This would involve explicitly incorporating focus into MOP-based activation so that requests that are at a high level of focus (Reichman, 1985) can receive additional activation.

MOP	*executing act.*	*active act.*	*goal act.*	*total act.*
J5	40	20	10	70
Dessert	0	0	20	20
Appetizer	0	20	32	52
	Figure 3a			
Dessert	40	20	20	80
Appetizer	40	20	32	92
Topic switch with failure	40	20	2	62
Topic switch without failure	40	20	0	60
Discussion	0	0	30	30
	Figure 3b			
Topic switch with failure	40	20	2	62
Topic switch without failure	40	20	0	60
Discussion	0	0	30	30
	Figure 3c			

Figure 3: Activations for Example

scenes which are not important to the example. In this conversation JUDIS has a strong commitment to keeping the dialogue coherent, so the constant for executing MOP-based activation is 40. At the start of this example, the conversation MOP that has the "chronological order MOP" as a scene is a top-level active MOP. JUDIS follows the dialogue laid out by this MOP through J4 which causes the main course topic MOP to be "executing". Here, one of the problems solvers has noticed that people do not often have enough bowls and the associated question has been given a priority of 10.

Meanwhile, the case-based reasoner has requested JUDIS to find out if any guests are allergic to tomatoes. The goal of the request has a priority of 40 since if there are guests that are allergic, a new constraint will be added to the problem which may affect both future problem solving and past decisions. When JUDIS receives the request it inserts it into the template for the dialogue by noticing that a question is part of a discussion of a topic. The subject for this topic MOP will be the appetizer because that is the focus of the case-based reasoner at the time the request is made.[3] The appetizer topic MOP has already been executed, but since the topic is a scene in the caterer's MOP, a new appetizer topic MOP is formed and attached to the caterer's MOP. Because the topic MOP is being added, it has no preceding scenes and is immediately capable of receiving any mop-based activation passed from the caterer's MOP.

JUDIS follows the indices in the topic switching MOP to find the specialization which pertains to previously executed topics. This specialization contains its own specializations including many, such as "Let's go back to ...", which cannot incorporate telling the user about the failure. One specialization, "I remembered..." allows this information to be included.[4]

Now JUDIS must decide when to execute the request. The activations for the MOPs of interest are shown in Figure 3a. The MOP containing J5 has the highest activation. Although there was an active goal with a much higher priority, the fact that the main course scene was currently executing gave J5 enough activation to execute at this time. The main course topic is marked executed after J5 has executed. Now the caterer's MOP becomes the executing MOP and both the dessert and appetizer MOPs and their scenes receive activation from it as shown in Figure 3b. The new appetizer topic MOP is chosen for execution, but it is not an executable act.

[3] We do not address the complexities of finding the topic of a request. Instead, we rely on the focus of the problem solver to correspond to some topic MOP. This is adequate for JUDIS since the conversation MOPs closely follow the problem solving strategies in the system.

[4] For this type of topic switching, we allow only entities explicitly mentioned in the previous discourse to be used. Here, for example, JUDIS can refer to guacamole, but not to the tomatoes which it contains. This is a simplification which, once again, is adequate for our needs.

Using this topic MOP as both the active and the executing MOP, and repeating the selection algorithm, the topic switch scene is selected. The activation levels for this selection are shown in Figure 3c. The topic switch MOP that includes the failure is chosen. The other specializations will never be chosen because the topic switch scene in the appetizer topic MOP is marked executed and these MOPs cease to receive activation. Now the appetizer topic MOP is executing. Next the question MOP, which satisfies the request, is selected and executed in J7.

Other requests may be handled differently, depending on the priorities of their goals. A request with a very high priority, say 100, would cause the current topic to be interrupted immediately, without executing a topic switch. However, a low priority request that is part of a topic MOP which is not yet ready for execution will have to wait until its topic is executed in the expected course of the conversation.

CONCLUSION

This paper describes an approach to the control of conversation, as implemented in a system called JUDIS, and an implemented example of its use of conversation MOPs to flexibly manage an interaction with the user. We believe JUDIS and its MOPs provide a method for combining intention and convention to generate flexible discourse and a means to sequence the conversational activities requested by different problem solvers in a composite system

REFERENCES

Appelt, D. E. (1985). Planning English referring expressions. *Artificial Intelligence*, 26:1–33.

Carberry, S. M. (1986). Pragmatic modeling in information system interfaces. Technical Report 86-07, Department of Computer Science, University of Delaware. Ph.D. thesis.

Cohen, P. R. & Perrault, C. R. (1979). Elements of a plan-based theory of speech acts. *Cognitive Science*, 3:177–212.

Cullingford, R. E. (1986). *Natural language processing: A knowledge engineering approach.* Totowa, New Jersey: Rowan and Littlefield.

Cullingford, R. E. & Kolodner, J. L. (1986). Interactive advice giving. In *Proceedings of the 1986 IEEE International Conference on Systems, Man and Cybernetics*, pages 709–714, Atlanta, Georgia.

Grosz, B. J. (1977). The representation and use of focus in a system for understanding dialogs. In *Proceedings of the Fifth International Conference on Artificial Intelligence*, pages 67–76, Los Altos, California. William Kaufmann, Inc.

Grosz, B. J. & Sidner, C. L. (1986). Attention, intention, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

Hinrichs, T. (1988). Towards an architecture for open world problem solving. In *Proceedings of the DARPA Workshop on Case-Based Reasoning*, pages 182–189, Clearwater Beach, Florida.

Hobbs, J. R. & Evans, D. A. (1980). Conversation as planned behavior. *Cognitive Science*, 4(4):349–377.

Kellermann, K., Broetzmann, S., Lim, T.-S., & Kitao, K. (1989). The conversation mop: Scenes in the stream of discourse. *Discourse Processes*, 12(1):27–61.

Kolodner, J. (1984). *Retrieval and organizational strategies in conceptual memory.* Hillsdale, New Jersey: Lawrence Erlbaum Associates, Publishers.

Litman, D. J. & Allen, J. F. (1987). A plan recognition model for subdialogues in conversation. *Cognitive Science*, 11:163–200.

McKeown, K. R. (1985). *Text generation: Using discourse strategies and focus constraints to generate natural languauge text.* New York: Cambridge University Press.

Reichman, R. (1985). *Getting computers to talk like you and me: Discourse context, focus, and semantics (An ATN Model).* Cambridge, Mass: The MIT Press.

Schank, R. (1982). *Dynamic memory.* New York: Cambridge University Press.

Shinn, H. (1988). The role of mapping in analogical transfer. In *The Tenth Annual Conference of the Cognitive Science Society*, pages 738–744, Montreal, Quebec, Canada.
Sidner, C. L. (1983). Focusing in the comprehension of definite anaphora. In Brady, M. & Berwick, R., (Eds.). *Computational models of discourse.* Cambridge, Mass.: The MIT Press.
Turner, E. H. & Cullingford, R. E. (1989). Using conversation mops in natural language interfaces. *Discourse Processes*, 12(1):63–90.
Webber, B. L. (1983). So what can we talk about now. In Brady, M. & Berwick, R., (Eds.). *Computational models of discourse.* Cambridge, Mass.: The MIT Press.