Title
Using Conversation MOPs to Integrate Intention and Convention in Natural Language Processing

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 10(0)

Authors
Turner, Elise H.
Cullingford, Richard E.

Publication Date
1988

Peer reviewed

Using Conversation MOPs to Integrate Intention and Convention in Natural Language Processing*

Elise H. Turner and Richard E. Cullingford
School of Information and Computer Science
Georgia Institute of Technology

Natural language processing systems must consider both convention and intention since both are a part of language. The conventions allow a society of language users to agree on many different facets of the language, from word meanings to discourse structure. Although it is easy to imagine a situation where convention almost completely governs a conversational exchange, a speaker often manipulates the conventions to help achieve some goal. When this happens the hearer must be able to understand the intention of the speaker to be able to fully understand the utterance.

Most previous work in natural language processing has focused on either intention or convention. Finding conventions and using them in computer system interfaces has been the focus of a great deal of research (for example, Grosz, 1977; McKeown,1985; Reichman, 1985; Sidner, 1983). In this work high level discourse structures are isolated and automated so that conversational flow appears natural.

Other research in natural language processing has focused on the intentional component of language. Philosophers of language recognized that each utterance is an act and, as such, can be subject to the same sorts of successes and failures as any other action (Austin, 1962; Searle, 1969). This insight led computer scientists to study speech acts in terms of planning (Allen & Perrault, 1980; Carberry, 1986; Cohen & Perrault, 1979; Litman, 1986). In their work, speech acts function as operators in plans. Conditions that must hold for the speech act to be used and the effects of the speech acts are associated with each operator. This method has been extended to handle difficult discourse such as indirect speech acts, where the literal meaning of the speech act is not intended by the speaker, and ellipsis, where a part of the utterance must be inferred. Planning methods provide flexibility by allowing the system to reason deeply about the intentions of the speaker and to form new dialogues from actions which can be put together in any way that follows the plans of the speaker and hearer.

Planning methods are not cognitively plausible and may expend computational effort unnecessarily because they do not take advantage of the conventions in language. Carberry and Litman incorporate discourse structures into their systems, but both take a plan-based approach. Our approach follows convention, whenever possible, as a shortcut in planning. However, the conventions are represented in such a way that they are flexible. The conventions can also be used as plans for inferring the user's plan or finding an action for the system if the need for that information warrants the additional computation.

We are implementing our ideas in JUDIS (JUlia's DIscourse System), a system which provides a natural language interface for Julia (Cullingford & Kolodner, 1986). Julia is a highly interactive advice giving system using a variety of problem solvers to provide assistance in common sense domains. The current version of Julia functions as a caterer's assistant helping the user to plan a meal. In the initial implementation of JUDIS, we have focused on the beginning portion of conversation, called the *initation phase* (Douglas, 1984), to narrow considerations to those which are directly affected by a fairly conventional, yet potentially flexible, conversational structure.

MOPS AND CONVERSATION MOPS

In our view, conversational conventions are learned from many past experiences with conver-

*This research is supported by the National Science Foundation, grant number IST-8608632. A longer version of this paper has been submitted to *Discourse Processes*

sation. These conventions can be used to achieve the same goals that have been associated with them in the past. However, since no conversation is expected to exactly repeat the past, the representation for conventions must allow general rules to be instantiated for a specific situation and must be flexible enough to allow the intentions of the conversants to impact them.

To represent these conventions, we use MOPs, or *memory organization packets* (Schank, 1982). A MOP is a schematic structure used to organize long-term, conceptual episodic memory. Each MOP represents a generalized episode which satisfies a goal. The building blocks of these episodes are *scenes*. Scenes represent pieces of the episodes and are associated with instrumental goals (Schank & Abelson, 1977) that are used to achieve the higher level goal of the MOP. For example, a MOP to represent buying theater tickets could include such scenes as: GET-MONEY, GO-TO-THEATER, GO-TO-BOX-OFFICE, ASK-TICKET-AVAILABILITY, SELECT-SEATS, PAY-FOR-TICKETS, and CHECK-TICKETS. The goals associated with the scenes and MOPs are important for capturing the intention in a discourse fragment.

MOPs can be stored in and retrieved from a *dynamic memory* (Kolodner, 1984; Schank, 1982). A dynamic memory stores MOPs using indices based on their important features and creates new MOPs from specific events that share features. In current implementations, the programmer decides which features will be important, although this could theoretically be learned by the memory. A specific value of an important feature is called a *predictive index*. These are used to find information in the memory. When a new episode is to be stored, a place for it is located in memory using the same indices that will be used to find it. If some predetermined number of episodes are found in this location, a generalized episode is created and the specific episodes are indexed under the MOP by their important differences from the MOP. If the location in which to store an episode already contains a MOP, the episode is indexed under this generalized MOP.

There is evidence that *conversation MOPs* represent conversational structures in human beings (Kellermann, Broetzmann, Lim, and Kitao, in press). Kellermann et al. studied the MOPs shared by members of a society, focusing on conversation MOPs that represent informal initial conversations such as those that occur between two people meeting in a grocery store or at a party.

The conversation MOPs that are used in JUDIS reflect these shared MOPs.[1] JUDIS currently has conversation MOPs representing general rules of conversation, such as turn-taking and question answering, as well as rules for the expected structure of a complete conversation. A complete conversational system will have very general MOPs for conversational goals that arise in most conversations, and more specific MOPs to capture conventions from particular contexts. Since many goals may be active in one conversation and additional MOPs will be chosen to guide the conversation in the given context, many MOPs will apply at once. JUDIS does not construct a complete plan from the conversation MOPs, but follows a MOP when an utterance or a goal of the system brings the MOP to the attention of the planner. (Details of this process are discussed below.) This method allows the system to respond to user utterances that do not fit into the MOP that the system is following. The method demands a mechanism for conversational control that can keep track of the many active conversation MOPs.

Since Julia is a caterer, JUDIS has a caterer's conversation MOP which gives the topics expected in a conversation between a client and a caterer. The caterer's MOP has a scene for the initiation phase. We have adapted the initiation phase found by Kellermann et al. (in press) to apply to a computerized caterer's assistant.

A simplified version of a conversation MOP for the initiation phase between a computer caterer and a new client appears in Figure 1. The initiation phase MOP contains the characters that appear in this scene. JUDIS assumes that it is functioning in the role of caterer, but it could use this MOP if it were the customer of a caterer. The goal of the MOP is simply to start a conventional conversation. This goal could be in service of some other goal such as getting to know someone, getting information, or, in the case of Julia, being a good and polite businessperson.

The scenes in the initiation phase give the actions that usually take place in this phase of the conversation. The scenes in the initiation phase are fairly well-defined; in later portions of the dialogue, scenes may be more vague. For specific

[1] JUDIS does not currently learn conversation MOPs, although this is a long-term goal of this research. We believe that the MOPs the system has been given are potentially the products of generalization in a dynamic memory.

characters: caterer, customer
goal: conventional-conversation
scenes:
  greeting
    The user and Julia exchange greetings such as "Hello." or "Hi."
  introduction
    Julia introduces itself; the user may introduce him- or herself if the user id given at login is not the user's name
  positive-evaluation
    exchange "It's nice to meet you." or some other positive evaluation of the introduction.
  ask-health
    both can ask "How are you?"
reason-for-presence
    Julia asks the reason for the person's visit
seq-of-events: greeting, introduction, positive-evaluation, ask-health, reason-for-presence
mandatory-scenes: greetings, introduction, reason-for-presence

Figure 1: The Initiation Phase MOP

rules of conversation, scenes may be very general actions, such as forming a question. Here, information from the current context will be used to fill in specifics of the question at the time the question is asked. The order of the scenes is given by seq-of-events. In this initiation phase MOP the scenes are totally ordered, but our representation also allows partial orderings. The scenes listed as mandatory scenes are expected to occur in this phase simply because of convention, not due directly to the active goals of the system.

Scenes which are not mandatory but which are present in the MOP are considered *optional scenes*. The system initiates an optional scene if it wants to achieve the associated goal. If the user initiates an optional scene, the system can infer that the user had the associated goal. Optional scenes give the conventions some flexibility by giving the system a conventional way to solve a goal, so it does not have to do extensive planning, but not forcing execution if the associated goal does not need to be achieved.

Later portions of the conversations with Julia will be controlled more by the system's and the user's problem solving. The goals associated with conversation MOPs can account for intention in conversation, and the use of MOPs in the problem solving systems of Julia allows the problem solving plans of the user to be available to JUDIS. The caterer's conversation MOP allows JUDIS to have reasonable expectations of topics to be covered during the entire course of conversation, without expending a great deal of computational effort. This overview allows JUDIS to integrate questions and information from Julia's different problem solvers. If an utterance requested by one of Julia's subsystems does not relate to the current topic and the utterance is not too important to the problem solver, the system can check to see when the topic will be discussed and include the utterance at the appropriate time.

AN OVERVIEW OF JUDIS

The current focus of JUDIS is on using conversation MOPs in the initiation phase. This phase is guided almost exclusively by its conversation MOP. The conversants are concerned with beginning the conversation, not with any problem that may be discussed later. This limits the initiation phase to goals that are related to conversation and not problem solving. The goals associated with the optional scenes will be persistent goals, like *polite-conversation*, that will set the stage for the rest of the conversation. To infer the user's plan we do not need to look at possible problem solving strategies. The system can also ignore relating problem solving to its own planning at this stage. Although we do not consider highly irregular dialogues, the initiation phase does contain optional scenes which allow us to study several aspects of the system's potential flexibility. The initiation phase also forces the system to handle several MOPs which account for the same utterance.

Finding the Initial MOPs

Initially, JUDIS must choose which MOPs are pertinent to the conversation. It starts with the goal of being a good caterer which includes the goals for carrying out a coherent conversation. The context of being in the catering environment is used to select the caterer's conversation MOP from memory. The familiarity of the caterer with the client is used as an index into memory to select the MOP for the initiation phase.

Other conversation MOPs are also selected from the memory at this time. The MOP for turn-taking is selected because of the goal to have a conventional conversation. This MOP is used in so many contexts and has become so general that it is selected whenever the system participates in a conversation.

Serendipitous Goal Satisfaction

There are three different levels of activation that can be associated with a MOP. A *dormant* MOP is in memory and is available to be activated by the conversation system, but is not associated with any active goal of the system or does not apply in the current context. In the case of a new user logging into Julia for the first time, the initiation MOP for familiar users would remain dormant throughout the session. The question–answer MOP is dormant at the start of the dialogue because it is not needed to satisfy any goals. An *active* MOP is associated with a goal that is active, but the MOP is not currently executing. If JUDIS has a question that it is waiting to fit into the conversation, the question–answer MOP for that question is active. MOPs are also active if they are the scenes of active MOPs. For example, while the initiation MOP is executing, the later scenes of the caterer's conversation MOP are active by virtue of being part of the active caterer's MOP. The initiation MOP is considered to be an *executing* MOP because it represents a plan that is currently being followed.

When an executable act or episode being performed to satisfy a scene in one MOP also satisfies a scene in another MOP, we refer to this as *serendipitous goal satisfaction*. We do not seek to minimize the number of actions executed by finding serendipitous goal satisfaction because we try to limit the amount of computing, not the number of generations, performed to achieve the conversational goal. We believe this method more closely reflects human processing. Very often serendipitous goal satisfaction occurs as part of executing a MOP and does not need to be coerced. For example, a question–answer sequence always satisfies turn-taking.

It is important to recognize when serendipitous goal satisfaction has occurred because it allows JUDIS to avoid repeating plan steps (usually utterances) unnecessarily. It is also important because serendipitous goal satisfaction can change the flow of conversation. To find serendipitous goal satisfaction, whenever an utterance is made, JUDIS checks the active MOPs to see if the utterance can be used to execute a scene in one of them. If it can be, that MOP is seen as executing serendipitously.

JUDIS currently only considers active MOPs when finding serendipitous goal satisfaction. We adopt this strategy to help the planner to maintain its focus. When serendipitous goal satisfaction occurs as the result of an action by the system, if a MOP were important to the goals of the system it would already be active. If the action forced the conversation into following a dormant MOP that was partially satisfied by the action, the system would find the MOP while trying to understand the user's utterance. If the user's utterance executed a scene that was dormant, JUDIS would execute the currently active MOPs, if it could, to respond to the utterance. However, as we work with more complicated portions of dialogue, we may find that we need to examine dormant MOPs more carefully.

Selecting a MOP for Execution

When serendipitous goal satisfaction occurs, more than one MOP is executing. The system cannot simply follow the MOP that it originally chose to execute, since it is not the case that following the originally chosen MOP will guarantee the continued satisfaction of the MOP that is executing due to serendipitous goal satisfaction. Sometimes the way the original goal is achieved demands a response from outside of the chosen MOP. If one person wanted to get the attention of another, he or she could say "Hello" or

"What's up?". "Hello" leaves room for a wide variety of responses available in the initiation MOP, but "What's up?" throws the dialogue into the question–answer MOP and demands an answer. The plan for conversation can also be disrupted by the user. Consequently, the system must choose which scene to execute next.

The choice of the next scene to execute is affected by three factors. As is usual in planning systems, goals are given different priorities and the higher the priority of the goal, the more likely a plan for the goal will be chosen for execution. This is especially important in later phases of the dialogue when an urgent need of problem solving may require the dialogue to be disrupted. The priority of a goal is usually set by its function in the problem solving task, although goals concerned only with maintaining the conversation have a high enough priority to keep the dialogue coherent. Since many of the most general rules of conversation are satisfied in the course of applying other conversation MOPs, such as satisfying turn–taking while executing the question–answer MOP, conversation goals such as *polite-conversation* are given a low priority so that the more specific MOPs have a chance to be executed.

Conversation MOPs are powerful because they can give more guidance than just associating a plan with a goal. The other two factors affecting MOP selection concern the relationship between scenes. The strength of the sequencing, whether or not the executed scene must be immediately followed by the scene being considered for execution, affects the choice of the next scene. If a question has been asked, it is important to answer it with the next utterance. On the other hand, if the appetizer scene has just been discussed, the main course does not have to follow immediately. Another factor to consider in selecting a scene is whether the scene being considered is mandatory or optional. It is important to execute mandatory scenes before optional ones to decrease the likelihood of the dialogue being terminated while mandatory MOPs are left unexecuted.

We use a simple activation model for scene selection. Each of the three factors above contributes to the activation of a MOP. The activation contributed by the goal remains proportional to the priority of the goal. The priority of the goal may increase or decrease, and when this happens the activation that it contributes to the scene will increase or decrease. As long as a goal's priority stays constant, the amount that it contributes to the scene's activation remains constant. However, the activation contributed by position and whether or not a scene is mandatory in the MOP is allowed to decay. This happens because as the time from one utterance to the next utterance in the MOP increases, executing the next scene in the MOP becomes less important. Sometimes the distance is so great that a topic shift must be executed in order to execute the scene.

For example, suppose one conversant asked two questions in a row without allowing the respondent to answer the first. The other conversant must decide which question to answer. Suppose he or she answered the first question first and this led to a discussion of the answer. The longer the discussion continued, the less likely it would be that the second question would be answered. This loss of interest is reflected by the decay of activation in our system.

The activation model also has implications for processing optional scenes. Since optional scenes need stronger activation from the goal than mandatory scenes do, for the system to execute an optional scene the associated goal for the scene must be active.

Two Examples of Initiation Phases

An example of an initiation dialogue that could occur between a user (U) and Julia (J) appears in Figure 2. From the login, Julia knows that this is a new user. The user says "hello" as part of the exchange greeting scene. This also counts as a turn for the user, so the turn-taking MOP is executing as a result of serendipitous goal satisfaction. The scene of returning the "hello" must follow the user's "hello", so JUDIS responds with "hello". The next scene in the initiation MOP is the introduction, so JUDIS has Julia introduce itself. There are no mandatory scenes in the initiation scene and no scene must follow the introduction, so JUDIS waits for the user to take his turn. Since the user must execute this scene, the system can only wait. The user says "How are you today?" which increases the strength of the conversational goal of *polite-conversation*. The "ask-health" scene contains an answer to "how are you?" built into its MOP. When someone says "how are you?" to a person he or she has never met, the hearer does not actually answer the question, but always says something positive. JUDIS knows how to respond from the initiation MOP

U1: Hello.
J1: Hello.
J2: My name is Julia.
U2: How are you today?
J3: Fine.
J4: How are you?
U3: Fine.
J5: May I help you plan a meal today?

Figure 2: Example Initiation Phase 1

and does not need to find the question–answer MOP. This scene continues with the system asking the user's health. The next scene to be selected for execution is reason-for-presence scene. There are no other scenes in the initiation phase, so this scene is chosen. It matches the caterer's goal of catering and can now be executed since mandatory scenes are not taking precedence.

In the first example, the user had the goal of being polite at a high enough priority that he executed the optional goal. But if the user did not have this goal, utterances U2–U3 would be left out of the dialogue. The new dialogue would be executed in the same way as the previous example for the first two utterances. As before, both the initiation MOP and the turn-taking MOP are executing. The initiation MOP has many optional scenes and no scene which needs to directly follow the introduction. Julia has a goal to cater, and the reason-for-presence scene is mandatory, but the scene does not need to closely follow the introduction so it does not have enough activation to be selected for execution. The user utterance in the turn-taking MOP must immediately follow a system utterance, so that scene is selected for execution. Since the scene must be executed by the user, JUDIS waits. But as the system waits, the contribution made by the strength of the connection between the scenes decays. Eventually, it decays enough that the decision for the next MOP relies mostly on the goals; "May I help you?" is executed since its goal has more strength than the goal of *polite-conversation* that is associated with turn-taking.

CONCLUSION

JUDIS handles a restricted conversational behavior, but we believe it can be extended to handle much more complicated parts of a dialogue. Some of the power of our conversational controller will be derived from its association with a dynamic memory and with problem solvers. The dynamic memory is useful for finding the appropriate MOP for a given context. The problem solvers can be used to infer general information about the world needed to understand a conversation. Their problem solving capabilities can be used to help plan conversation when active MOPs do not specify enough detail. As we continue our research, the conversation MOPs themselves will need to be expanded as the system handles dialogue that is less constrained. The factors contributing to the activation of a MOP will have to be expanded. We will also have to consider more complicated MOPs, such as those for topic switching.

As we move further into the dialogue, we will need to explore how problem solving affects the use of convention. Here, the intentional component of natural language processing will be even more important and a user's plan will more often need to be inferred. The system will also need to make general MOPs more specific by filling in information from the problem solving context. The problem solving systems and the user will place more demands on JUDIS's flexibility and will require many MOPs to be active at one time.

Although we must continue to develop our system in order to handle more interesting conversational phenomena, our current implementation has given us a solid base to work from. We have used the philosophy of minimizing cognitively implausible reasoning effort in a system that can be integrated with more intensive reasoning when necessary. By using conversational MOPs to represent conversational rules, we have laid the groundwork for a system which can be flexible while retaining the efficiency of following conventions.

ACKNOWLEDGMENTS

Many thanks to Roy Turner and Mark Graves for their helpful comments on earlier versions of this paper.

REFERENCES

Allen, J. & Perrault, C. R. (1980). Analyzing intention in utterances. *Artificial Intelligence 15*, 143–178.

Austin, J. L. (1962). *How to do things with words.* (J. O. Urmson Ed). New York: Oxford University Press.

Carberry, M. S. (1986). *Pragmatic modeling in information system interfaces.* (Tech. Rep. No. 86-07 and Ph.D. thesis). Newark: University of Delaware, Department of Computer and Information Sciences.

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

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

Douglas, W. (1984). Initial interaction scripts: When knowing is behaving. Paper presented at the Annual Meeting of the International Communication Association, San Francisco.

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

Kellermann, K. & Broetzmann, S., Lim, T., Kitao, K. (in press). The conversation MOP: Scenes in the stream of discourse. To appear in *Discourse Processes.*

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

Litman, D. (1986), Understanding plan ellipsis. In *Proceedings of the National Conference on Artificial Intelligence,* (pp. 619–624), Los Altos, California: Morgan Kaufmann.

McKeown, K. R. (1985). *Text generation: Using discourse strategies and focus constraints to generate natural language 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. C. & Abelson, R. P. (1977). *Scripts, plans, goals and understanding: An inquiry into human knowledge structures.* Hillsdale, N.J.: Lawrence Erlbaum Associates.

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

Searle, J. (1969). *Speech acts: An essay in the philosophy of language* Cambridge, UK: Cambridge University Press.

Sidner, C. (1983). Focusing in the comprehension of definite anaphora. In *Computational models of discourse,* M. Brady and R. Berwick, eds., Cambridge, Mass.: MIT Press.