Transition Networks for Discourse Management [1]

David D. McDonald
Jo Ann Brooks†
Beverly Woolf
Philip Werner

Department of Computer and Information Science
University of Massachusetts
Amherst, Massachusetts 01003

†Lisp Machine Inc.
1000 Massachusetts Avenue
Cambridge, Massachusetts 02138

March 31 1986

COINS Technical Report 86-34

Abstract

Transition networks are a natural notation for formalizing the space of actions and interpretations through which a program must navigate when holding a conversation with a human user [7,19]. However, dissatisfaction with the augmented transition network formalism that we used in earlier work on tutoring dialogs [28] has led us to develop the new formalism that we present in this paper. We have implemented a program framework—a virtual machine—that defines a node and arc notation interpreted semantically in terms of situations (arcs) that dictate the actions (nodes) that the system is to take. Our work is based on analyses of recorded conversations and grounded in an experimental man-machine interface by which a human lawyer would get advice from a legal reasoning program. We present the motivations behind a discourse manager and its place within a man-machine interface, a definition of our formalism, and illustrations of its use in two different domains.

[1]This work was supported by the Air Force Systems Command, Rome Air Develop- ment Center, Griffiss Air Force Base, New York 13441-5700, and the Air Force Office of Scientific Research, Bolling AFB, DC 20332 under Contract No. F30602-85-C-0008. This contract supports the Northeast Artificial Intelli- gence Consortium (NAIC). Additional support was provided by contract N00014-85-K-0017 from the Defense Advanced Research Projects Agency.

1. Introduction

To date, most natural language interfaces restrict their conversational interactions to sequences of question–answer pairs. Discourse level control decisions in such systems are predetermined by a set of narrowly specified interactions derived from the subject matter of a specified problem domain. If however, we envisage natural language interfaces which are capable of managing the dynamic conversational exchanges required for advice giving systems in law or medicine, such simple discourse control schemes are soon overwhelmed.

A discourse controller, or "manager", for dynamic conversation must have the ability to handle *contingent* discourse events, rather than those predefined in script-like sequences or discourse grammars. For instance, a discourse manager must be able to handle the problem of indexicality which current interfaces either deny or engineer out. Problems of indexicality in dynamic conversation occur because speakers may shift context several times, or create long distance dependencies between a current topic and a previous topic. In the research reported below, we describe a machine with the capacity to make the control decisions necessary for dynamic discourse, and that facilitates the context dependent interpretations required to understand situated conversation and generate pertinent responses.

This work contributes to two long range goals: giving programs the ability to hold sophisticated, dynamic conversations with human users, and furthering our scientific understanding of the nature of conversation between people. Toward this end we collect and analyze tape recorded conversations between people in task-specific settings, and construct computer programs that can actually participate in domain sensitive conversations. The topics and tasks of the conversations we record are restricted to subject matter and intentions that our programs can realistically be expected to handle in the future. Analysis of the recorded conversations provides the source of conversational principles; their implementation in programs provides us with a way of testing and above all, experimenting with, their adequacy and extensibility.

2. Discourse Management

A *discourse manager* is responsible for the control decisions in discourse construction and inference. A discourse in our framework is defined by reference to a space of possible discourse–level situations, where a situation is a set of predicates tracking the state of the discourse from the perspective of the program engaging in the conversation. Conversational actions selected by the manager and responses from a user change the discourse situation. Given a new situation, the manager must decide what the system should say next and how it should interpret and act on subsequent responses from the user. In the work described below, we do not constrain the set of possible user responses to be only rigid interactional sequences. Hence a desiderata on the design of the discourse manager is that it respond fluidly, coordinating its utterances in a more flexible manner than has been required for question/answer or summarization systems.

Attorney:	I represent a client named HACKINC who wants to sue SWIPEINC and Leroy Soleil for misappropriating trade secrets in connection with software developed by my client. HACKINC markets the software, known as AUTOTELL, a program to automate some of a bank teller's functions, to the banking industry. In 1982, Leroy Soleil, one of HACKINC's personnel, left HACKINC and began working for SWIPEINC on a competing product, TELLERMATIC, also an automated teller program. SWIPEINC has begun marketing TELLERMATIC in competition to AUTOTELL.
Counselor:	Did Soleil enter into a noncompetition agreement or a nondisclosure agreement with HACKINC?
Attorney:	Yes, a noncompetition agreement.
Counselor:	Was it an express written agreement not to work for competitors of HACKINC?
Attorney:	Yes, he signed it in June, 1980.
Counselor:	Was Soleil the only source through which SWIPEINC could get HACKINC's confidential information about AUTOTELL?
Attorney:	Assume that SWIPEINC's sole source was Soleil.
Counselor:	Here is a summary of some arguments and responses you might want to examine in greater detail: You might argue that Soleil breached the noncompetition agreement by working for SWIPEINC, a competior of HACKINC, and by disclosing to SWIPEINC confidential information concerning AUTOTELL. You might argue that there is an implied agreement arising out of Soleil's employment with HACKINC that Soleil would not disclose any trade secret information to which he gained access by virtue of his employment.

Figure 1: **Dialog between an Attorney and Counselor**

3. The Counselor Project

Any AI research project on discourse must be grounded in a subject matter and a task, and as a program, must be working within an interface to some domain-specific system that a person would be interested in conversing with. At UMass we are building a natural language interface to a legal reasoning system named Hypo, developed by Kevin Ashley and Edwina Rissland [1]. Hypo draws on a reference library of litigated cases in trade secret law to give advice to other lawyers about how they could argue a new case. The program reasons analogically along the qualitative dimensions of its reference cases to determine the strengths and weaknesses of a new case in a given fact situation. Hypo incorporates a model of legal argument that enables it to give the lawyer advice on how to best argue her case and counter the arguments of her opponent.

During the spring and summer of 1985, the UMass Counselor Project [17] developed an initial natural language interface to Hypo, with Mumble as the generator [16], Plum as the parser [14], and a new inference and discourse structure component named Cicero as interface to the two linguistic components [18]. The dialog in Figure 1 is an example of the Counselor

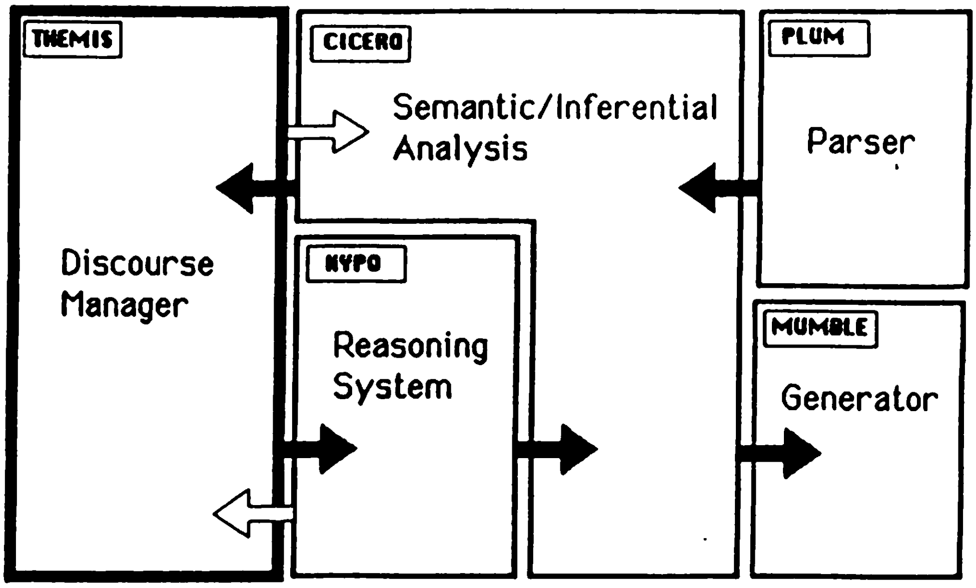

Figure 2: **Components of the Counselor Natural Language System**

interface's ability. [2] In this setting, our discourse manager lies to the left of Cicero and Hypo as shown in Figure 2. The channel that is part of Cicero and surrounds Hypo reflects the fact that inferences and knowledge governed strictly by the semantics of natural language and discourse conventions cannot be totally separated from those governed by the particulars of the underlying conceptual domain.

4. Discourse Action Transition Networks

Figure 3 shows the Discourse ACtion Transition Network (DACTN) that would be used to manage the dialog in Figure ??. [3] The network consists of four abstract actions, shown as nodes, that are connected by directed arcs which define the possible transitions between them. The arcs correspond to situations that the system can be in, as indicated by the labels on them. Only two situations need to be distinguished in order to capture the control criteria that are germane to the management of the dialog portion we have shown.

To the eye, our DACTN has the look of a conventional transition graph of "states" and "arcs" as one would find in an ATN, yet we are talking about it in terms of "actions" and

[2] Except for the final summary of arguments, this dialog ran last August, 1985 with the program installed on a Symbolics 3670.

[3] The demonstration system of August 1985 did not use a discourse manager. The simplicity of the conversation did not require it. New work just getting underway in Counselor will explore the possible arguments and counter-arguments and will be driven by a DACTN.

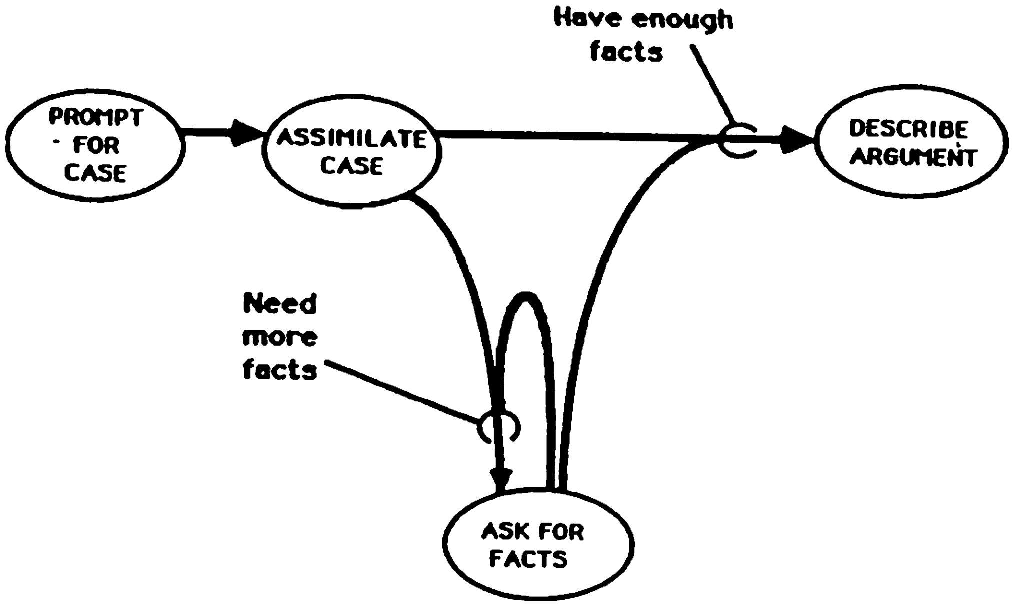

Figure 3: **Partial Discourse Network for the Discourse in Figure 1**

"situations" as one would do in production systems. This mixing of control techniques is deliberate, and reflects the evolution that our ideas about discourse management have gone through. Before discussing our notation and its interpretation, a review of that evolution is in order.

5. History of the Project

In the beginning of our work with the Counselor Project [6], we expected to simply carry over the ATN framework that had been the basis of the discourse manager in Woolf's earlier research on tutorial dialogs [28] We had adopted an ATN originally because our analysis of tutoring dialogs [29] viewed the dialogs as repeating cycles of parameterized discourse moves (including the interpretation of the response of the student). As such, these dialogs were conveniently represented by a network with the individual move schemas organized into a three level refinement hierarchy. The Discourse Manager viewed the state of the system (particularly the student model and topic space) in terms of ATN registers. Movement between local cycles involved non-local arcs called "meta-rules" after Clancey [8]

That work was based on a shell program that drove a simple, template-based generator and simulated the reasoning that a real domain reasoning system would have provided (in the domain of tutorials about weather as analyzed by Stevens et al. [23]). Once we began adapting that component to Counselor, and now had a proper underlying expert program and linguistic interface, we found that we kept stumbling over minor flaws in the ATN, such as having to rewrite earlier rules to accomodate later ones. This suggested to us that something fundamental was wrong about our entire design, and motivated the thinking that lead to the current formalism.

The motivation to completely drop the ATN design stems from the observation that if one is not using all of the capacity that an architecture provides then it is probably the wrong one. In the case of our ATN, the prime un-used facility was non-determinism. In the original ATN design [27], non-determinism is motivated by uncertainty, the need to wait for an accumulated global interpretation before being confident about the local interpretation of each token being scanned. Such uncertainty has has no counterpart in our system. All of its transitions can be taken deterministically.

Our next observation was that discourse management chooses between actions, not interpretations, putting it in realm of planning formalisms rather than parsers. We judged that several changes to the semantics of the ATN notation were then needed. Shifting actions from the arcs to the "states" facilitates notations for expanding abstract actions into substeps. Deterministic execution permits grouping the arcs coming out of a state into a single discrimination net. From here it is a short step to seeing these groupings as the "states", better thought of as situations, and taking the execution cycle to be the association of situations with actions, in the manner of a production system.

6. Definitions

Discourse action transition networks, or "DACTNs" are directed graphs of labeled nodes and arcs. Every node has at least one arc coming into it. Every arc terminates in one and only one node. Typically, but not necessarily, an arc originates at one node; however arcs are allowed to be free floating (with no originating node), or may originate at more than one node (in effect giving the arc a split tail). The control knowledge of a discourse manager will be encoded in a set of these graphs, viewed by the designer as involving a number of different organizational spaces; this distinction plays no role in the computational interpretation of the graphs however.

Nodes in the network correspond to actions that the system can take. The actions are typically abstract, needing to be expanded and refined one or more times before they take on a form that can actually be executed. Action expansion is an activity of the discourse manager; execution of concrete actions is carried out by other parts of the system, typically the underlying domain expert component (e.g. Hypo) or the text planning and inferencing component (e.g. Cicero). Our notion of actions borrows principally from Sacerdoti [20] and Stefik [22].

Every arc corresponds to one and only one "situation" that the system can be in. These situations are abstractions over the system's state and are defined in terms of a predefined set of predicates. At any given moment each predicate will be either true or false. Predicate values change as a result of the actions the system takes. Changes to the predicates are implemented by incorporating calls to set/reset functions directly into the routines of the system's other components. Predicates are indexed to the situations that mention them, allowing the record of the situation(s) the system is in to be continuously updated. (especially including reasoning it may do) and also also as a result of the interpretations the system makes of the actions of the other conversant. The outer loop of a discourse manager is: (1) access what situation the system is in (resolve conflicts if necessary); (2) direct the system's other components to carry out the corresponding action; (3) repeat.

The aggregation of predicates into situations is the chief notational device at the service of the designers of a discourse manager. It is a way of declaring what combination of finer-grained changes (e.g. changes to the values of individual predicates) are consequential in changing the actions of the system as a whole.

Situations will typically be "partial", not specifying what values certain predicates should have. [4] They may also overlap, sharing one or more predicates; if one situation includes all of the predicates in a second we say that it "subsumes" it and corresponds to a more specific situation. At any moment a discourse manager may find itself simultaneously in a number of different partial situations, each leading to a different action. In this case a "conflict resolution" phase is required, analogous to what happens in a production system when the left-hand sides of more than one rule are satisfied. The most specific subsuming situations will be preferred over other situations with which it shares predicates. In the case where the situations are incomparable (i.e. their sets of predicates are disjoint) a conventional priority must be defined, for example, that answering a question from the user takes priority.

When an arc originates in a particular action, there is a tacit predicate included in the arc's situation to the effect that that action must have just been taken. This aspect of the formalism is the prime manifestation of the fact that we are using a network formalism and not simply a set of situation action rules operating over a short-term memory.

7. Multiple networks working together

As an example of a more elaborate DACTN, consider the dialog in Figure 4 and the networks in Figures 5, 6, and 7. These are a re-analysis of the actually occuring dialogs collected by the WHY project [23] that we modeled in our earlier work on tutorial dialogs [28].

Tutor:	Do you know what the climate is like in Washington and Oregon?
Student:	Is it cold?
Tutor:	No, it is rather mild. Can you guess about the rainfall there?
Student:	Normal, I guess.
Tutor:	Well, the Japan current, which starts in the South eastern Pacific, goes along the coast of Japan and across the North Pacific ending up off the coast of Washington and Oregon. How do you think that current affects the climate there?
Student:	It's probably rainy.
Tutor:	It is indeed very wet, there are rain forests in both Washington and Oregon. What does that tell you about the temperature of the Japan current?
Student:	It's warm.
Tutor:	Right.

Figure 4: **Tutoring Dialogue about Rainfall**

[4] It is likely that our notion of "situation" is a close parallel to that defined by Barwise and Perry [2].

Figure 5 shows two conversational move schemas which constitute alternative tutoring strategies. The Explore-Competency Schema consists of three steps: probe topic, ask question, and correct student's response if wrong. The steps of the Teach-Topic schema are: teach information by supplying a fact, ask a leading question, and finally, acknowledge student's answer if correct. The two schemas differ in the manner they employ the question they each ask.

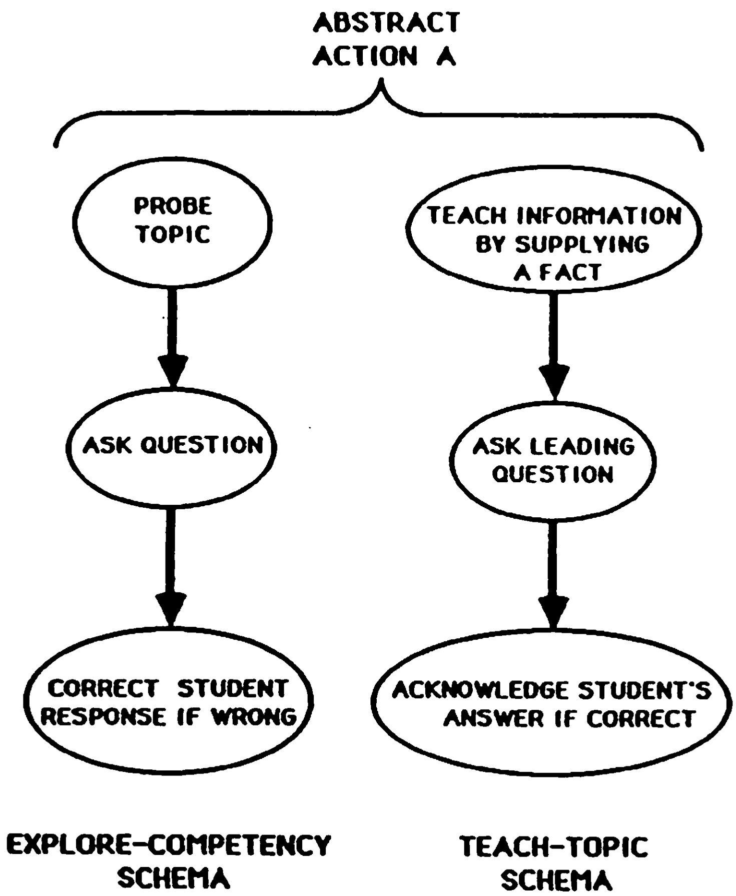

Figure 5: **Alternative Tutoring Strategies**

Both schemas are specializations of a more abstract action named "A". The function of the class of actions denoted by "A" is to question and inform. However, while the action networks corresponding to Explore-Competency and Teach-Topic are empirical (i.e. they are apparent in the recorded conversations), the abstract action is not. It is an emergent generalization, and illustrates the power of our notation as an explanatory framework through the construction of functional abstractions across multiple networks.

Both tutoring strategies take the topic and locale networks shown in Figure 6 as parameters. The subject of the topic network is climate; the actions of the network correspond to the sequential organization of a syllabus, in the spirit of the genetic graph proposed by Goldstein [10]. The locale network corresponds to the geographic locations by which tutoring questions are contextualized.

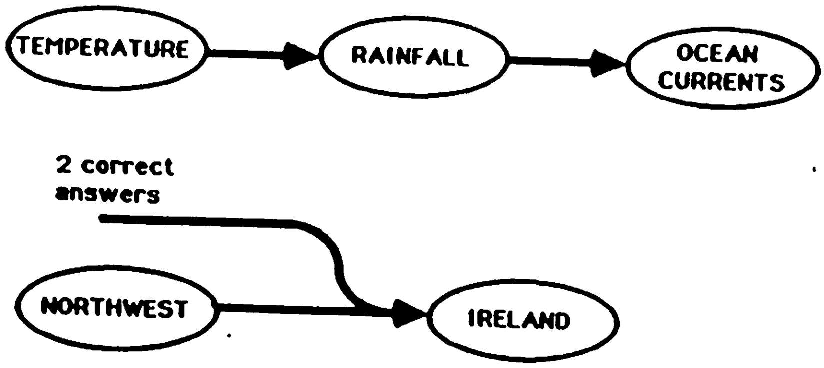

Figure 6: **Topic and Locale Networks**

Figure 7 displays the network controlling shifts between tutoring strategies and topics in the syllabus. By default, a tutorial simultaneously starts in the the probe topic action of the Explore-Competency schema, the temperature action of the climate network, and the Northwest action of the locale network. If a student responds with two correct answers, that situation leads to an action which is a composite of the following actions: the current tutoring strategy, the current topic in the climate network, and the next action in the locale network, namely Ireland. If on the other hand, the student responds with two incorrect responses, the discourse manager's next action is a composite of a new tutoring strategy, Teach-topic, the current action from the climate network, and the current action in the locale network.

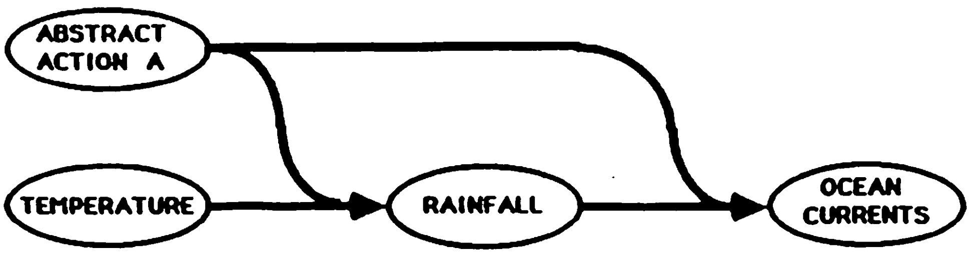

Figure 7: **Shifts Between Topics**

Every situation/arc includes as one of its consitutent predicates the action(s) from which it came, as for example in Figure 6, which also illustrates a situation that does not originate in an action. The predicate "2 correct answers" is established independently by record-keeping in Cicero. This flexibilty, where situations are not constrained to follow a network path but may arise asyncronously, is necessary to cooperatively engage in conversation, when the system must continually adapt to real-time changes in both the knowledge base and the user.

8. Related Work

While there has been a large amount of AI research on discourse, especially in recent years, relatively few performing programs have been demonstrated, and of these still fewer engage in the kind of dialogs that we are studying. Question answering systems [e.g. [26,25] and summary description and explanation systems [e.g. [15,24] finesse the control problems we are concerned with by limiting the pattern of the man–machine interactions a priori. We will limit this review to systems like our own that have engaged in mixed-initiative dialogs where system and user cooperatively navigate through a space of possible discourse situations. Our chief concern will be the techniques such systems used to control their dialogs.

The earliest program with a discourse controller was Gus, an airline reservation system [4]. Gus was built around an agenda, which selected tasks whose execution caused entries for new tasks to be created and placed on the agenda in turn. A supervisory program could reorder the agenda, for example to take advantage of information volunteered by the person before the system would normally have asked for it. Gus can be seen as the first instance of a class of systems that are driven by filling slots in frames. Filling or referencing a frame slot activates simple situation-driven schemas that pose questions or announce results. Other systems in this class include the Wumpus Advisor [21] and the Personal Assistant [5].

The chief weakness of a frame-driven design is that its model of the space of conversational moves available to it is ad hoc and unstructured. This need not be a problem in a simple task like reservation scheduling, but introduces unnecessary complexity as the dialog itself becomes more diverse and dynamic as, for instance, in an argument or a mixed initiative exploration of a topic such as the Counselor Project is developing now.

A second organizing theme in dialog control is the augmented transition network. The ATN represents the dialog space explicitly as alternative sequences of conversational or inferential moves. The SpeechLis system [7], a trip planning system contemporary to Gus, used a single ATN for both sequencing through the steps of the task and managing the system's utterances to the human user.

More recently Reichman [19] uses an ATN to organize her system's choice of conversational move at each sucessive turn in the dialog. Here again there is no explicit representation of the space of conversational moves, just constraints on what is a valid next move. Continuity across turns is represented not by the paths in the ATNs but in registers set during prior turns. In the DACTN framework, turn taking is implicit in the actions that modify the situation.

The system most like our own is Clancey's Guidon [9,8], a tutor built on top of Mycin. Like Mycin, Guidon is also written as a set of situation action rules driven by backwards chaining from goals. This provides the flexibility to accomodate to dynamically changing discourse circumstances. However it also carries with it the liabilities of a pure production system, namely that there are no provisions for sequencing situation action chains except by using ad hoc state variables. In a Discourse ACtion Transition Network, the chaining of arcs from actions is just such a sequencing mechanism.

9. Conclusion

We have proposed a framework that supports the control decisions necessary for dynamic man–machine conversation; the machine we have constructed is the procedural realization of this framework. Fundamental to our perspective is that conversation can be thought of as a navigation through a set of possible discourse situations. Transitions from one situation to another are not structurally predetermined in our formalism, and we are therefore able to track and respond dynamically to contingencies in a conversation.

The segmentation of conversation into a sequence of situations, rather than into a set of turns, topic shifts, or conversational moves, reflects the level of abstraction that we think is operative in discourse management. As defined, situations are collections of predicates in multiple transition networks, where each network contributes a partial description of the resources available for engaging in conversation. When viewed from the perspective of a particular network, a discourse action can locally be said to fulfill the functional role the network specifies. From the global perspective of a particular situation, that discourse action has multiple functional interpretations, each corresponding to the network that the situation's predicates index. This agrees with our intuition that discourse actions normally serve many purposes simultaneously and are situation bound [3] Yet a third perspective for viewing the functionality of a discourse act occurs at the level of sequences of situations; this is the history of situations that have been satisfied during the course of a conversation. We expect that regularities in the history will be found to constitute discourse structures with the theoretical value and stature of the competence-based notions articulated in descriptive accounts [11,19,12]

REFERENCES

[1] K. Ashley and E. Rissland. "Toward Modelling Legal Arguments". *Proceedings of the Second International Congress, Logica, Informatica, Diritto, Automated Analysis of Legal Texts*, Florence, Italy, 1985.

[2] J. Barwise and J. Perry. *Situations and Attitudes*. MIT Press, Cambridge, MA. 1983.

[3] J. Bateman. *Utterances in Context: Towards a Systemic Theory of the Intersubjective Achievement of Discourse*. Ph.D. Dissertation, Department of Artificial Intelligence and the School of Epistemics, University of Edinburgh, 1985.

[4] D. Boborow, R. Kaplan, M. Kay, D. Norman, H. Thompson, and T. Winograd. "GUS, A Frame-Driven Dialogue System". *Artificial Intelligence* 8:2, 1977.

[5] C. Bullwinkle. "Levels of Complexity in Discourse for Anaphora Disambiguation and Speech Act Interpretation". *Proceedings of IJCAI*, MIT, August 1977.

[6] J. Brooks. *Themis: A Discourse Manager*. M.Sc. Thesis, Computer and Information Science, University of Massachusetts, Amherst, 1985.

[7] B. Bruce. "Discourse Models and Language Comprehension". *Proceedings of the ACL*, Boston, 1975.

[8] W. Clancey. "Tutoring Rules for Guiding a Case Method Dialogue". D. Sleeman and J.S. Brown (Eds.) *Intelligent Tutoring Systems*, Academic Press, Cambridge, 1982.

[9] W. Clancey. "Dialogue Management for Rule-based Tutorials". *Proceedings of IJCAI*, Vancouver, 1979.

[10] I. Goldstein. "The Genetic Graph: a Representation for the Evolution of Procedural Knowledge". D. Sleeman and J.S. Brown (Eds.) *Intelligent Tutoring Systems*, Academic Press, Cambridge, 1982.

[11] B. Grosz, and C. Sidner. "The Structure of Discourse Structure". *Computational Linguistics*, to appear.

[12] J. Hobbs. "Towards an Understanding of Coherence in Discourse". W. Lehnert and M. Ringle (Ed.) *Strategies in Natural Language Processing*, Hillsdale, NJ., Lawrence Erlbaum Associates, 1982.

[13] W. Lehnert. *The Process of Question Answering*, Lawrence Erlbaum, New Jersey, 1978.

[14] W. Lehnert and S. Rosenberg. "The PLUM Users Manual". Technical Report No. CPTM 1, The Counselor Project, Department of Computer and Information Science, University of Massachusetts, 1985.

[15] K. McKeown. "Discourse Strategies for Generating Natural Language Text". *Artificial Intelligence*, **27**:1, 1-41, 1985.

[16] D. McDonald and J. Pustejovsky. "Description-Directed Natural Language Generation" *Proceedings of IJCAI*, 1985.

[17] D. McDonald, J. Pustejovsky, W. Lehnert, and E. Rissland. "The Counselor Project", *The Finite String, Journal of Computational Linguistics*, 1986.

[18] J. Pustejovsky. "An Integrated Theory of Discourse Analysis", unpublished ms. Computer and Information Science, University of Massachusetts, 1986.

[19] R. Reichman. *Getting Computers to Talk Like You and Me.* The MIT Press, Cambridge, 1985.

[20] E. Sacerdoti. "Planning in a Hierarchy of Abstraction Spaces". *Artificial Intelligence*, **5**:2, 115–135, 1974.

[21] J. Stansfield, B. Carr, and I. Goldstein. "Wumpus Advisor I: A First Implementation of a Program that Tutors Logical and Probabilistic Reasoning Skills", MIT AI Lab Memo 381, 1976.

[22] M. Stefik. "Planning with Constraints". *Artificial Intelligence*, **16**, 111–140, 1981.

[23] A. Stevens, A. Collins, and S. Goldin. "Diagnosing Student's Misconceptions in Causal Models". D. Sleeman and J.S. Brown (Eds.) *Intelligent Tutoring Systems*, Academic Press, Cambridge, 1982.

[24] W. Swartout. "Producing Explanations and Justifications of Expert Consulting Programs", Tech. Report, MIT/LCS/TR-251, MIT, Cambridge, 1981.

[25] R. Wilensky. "Talking to UNIX in English: An Overview of UC". *Proceedings AAAI-82*, Pittsburgh, August 1982.

[26] T. Winograd. *Understanding Natural Language.* University of Edinburgh Press, 1973.

[27] W. Woods. "Transition Network Grammars for Natural Language Analysis". *Communications of the ACM*, **13**:10, 591–606, 1970.

[28] B. Woolf and D. McDonald. "Design Issues in Building a Computer Tutor". *IEEE Computer: Special Issue on Artificial Intelligence for Human-Machine Interaction*, Sept. 1984.

[29] B. Woolf and D. McDonald. "Human-Computer Discourse in the Design of a PASCAL Tutor". *CHI: Human Factors in Computer Systems*, ACM, New York, 230–234, Dec. 1983.