DIALOGUE WITH AGENTS IN A DYNAMIC WORLD

David Duff
MITRE Corporation

Susann Luperfoy
MITRE Corporation
Georgetown University

Introduction: Discourse Semantics for Multimodal Dialogue

We have constructed the discourse component for a multimodal user-interface dialogue system. The larger system acts as a conversational agent engaged in dialogue with the human user. Output originating with the backend application system can take the eventual form of spoken English utterances, printed English text, or changes in the graphical display screen, e.g., appearance, disappearance, movement, or color change of a screen icon, and left or right panning or zooming of the overall display. User input utterances can be typed or spoken English queries and commands, mouse gestures (simple clicks), and typed English or command-language utterances.

The underlying semantics of the system are provided by a knowledge base that contains entries for the objects and actions that reflect the capabilities of the backend application. The knowledge base serves as the belief system defining semantics for the automated dialogue agent. Therefore, all input and output references must be converted into legal forms in the language of the knowledge base. So a mouse click on the icon for a vehicle, the user's spoken or keyboard utterance, "The red bus," or the mixed modality input "This <mouse-click> bus" need to map to the same form in the command/query language of the backend database.

Difficulties in Using a Dynamic World Model

The discourse framework we propose, addresses a family of problems that arise in systems for which there can be no assumption of a close world of reference or a static belief system over the course of the dialogue. In one application of our system the user's task was to understand and modify the contents of a knowledge base. That knowledge base being edited also served as the semantic basis for the Natural Language Understanding (NLU) system that performed interpretation of the user's input. In such a system the user will necessarily refer to concepts that are not yet in the knowledge base and therefore not yet understood by the NLU system. Similarly, reference can be made to objects that have been removed from the knowledge base. In this case, the lexical items are familiar to the NLU system but the corresponding knowledge base entry is absent. This prevents the reliance on world knowledge to help interpret the user's referring expression.

Our discourse module assumes the interpretation components to be limited in coverage to some subset of English and falible even within their scope. From the speech recognizer and lexicon to the discourse module itself, any module can experience incompleteness or uncertainty. This means that at a given moment, the current discourse interpretation may be incomplete or wrong. The interpretation process may fail partially in several ways. For example, given a correctly analyzed "that bus" from the speech, syntax, and semantics analyzers, the discourse system may fail to locate "bus" as a subclass of "vehicle", it may identify the class "bus" and its location in the ontology but not know which instance of that class is being denoted, or it may find the bus in the knowledge base without being able to relate it to anything in focus in the ongoing discourse. Thus, discourse component methods must allow for augmentation of incomplete interpretations and nonmonotic correction of an incorrect discourse model during subsequent dialogue.

A Proposed Discourse Framework

We use an adaptation of the discourse representation documented in (LuperFoy 1991; 1992). It includes three tiers of information available to the discourse component at run time:

1. Knowledge base information

The objects at this level are entries in a knowledge base. The discourse processor exploits world knowledge during interpretation by querying the knowledge base at run time to determine the presence and arrangment of objects. It also invokes the reasoning engine to test the likelihood of hypothesized relationships between denoted objects.

2. Discourse information

The objects at this level are called Discourse Pegs and serve as loci for collections of attributes. There is one peg for each construct under discussion. Pegs, their attributes, and their links to each other and to objects on the other tiers, are all nonmonotonically updatable.

3. Surface form information

The objects at this level are called Mentions as there is one for each occasion in which a Discourse Peg is mentioned by either dialogue participant (user or backend system). Each channel of communication has an independent 'utterance' processor that captures input and converts it into a simple first-order predicate logic expression. For the typed English input channel the NLU modules include morphology, syntactic parsing, and sentential semantics. A mention is created in the discourse representation for each referring expression in the resulting logical form.

Output forms are also converted by the discourse processor into predicate-argument forms based only on the objects or constructs that are "mentioned" by the output utterance. There is no true NL generation module, so output English utterances are simple string templates, the slots of which get filled by discourse pegs.

Mention information on the surface form tier decays rapidly as a function of time. Mouse clicks, screen output events, and English referring expressions vanish soon after being introduced. Discourse tier information decays only as a function of attentional neglect; as long as a discourse peg continues to be mentioned in the dialogue it remains in the attentional focus structure in the discourse model. Knowledge base information does not decay and is removed only in response to deliberate action.

Embodied Rules for Dialogue Interaction

Our representational framework was developed as part of a mixed-modality dialogue interface to a knowledge base editor for Cyc. It has since been adapted to computer-mediation of bilingual spoken dialogues and to the discourse component of a GIS (geographical information system) application supporting dialogue segments such as the following:
user: "bring up the map of Bosnia"
system: <displays a map>
user: "pan left"
system: <updates the display>
user: "zoom in on that lighthouse"
system: <zooms in on a screen icon that appeared as a result of the previous panning operation>

We are currently building the dialogue manager for a spoken dialogue interface to a distributed battlefield simulation, where each user issues spoken commands to control semi-automated opposition forces in the simulation environment.

In the latter application (as in the simulated dialogue application being developed by Webber et al.) there is the potential for two types of dialogue: the modelled dialogue between two simulated agents, and the dialogue between a synthetic agent and the human user of the system. The synthetic agent may be (A) one of the synthetic characters that resides in the simulated world, or (B) the disembodied user-interface agent that controls the simulation. For Case B our system must support dialogues between the user and the controller agent, about the camera position--panning, zooming, jumping, calling up alternative displays, etc. like the example in the preceding paragraph.

When the user chooses to address one of the characters in the simulated world (Case A) the dialogue utterances concern that character's perspective on the world. For example, when the user says "go to the nearest closed door and open it" and the system agent goes to nearest door in the room. If a problem arises in carrying out the action the system must engage the user in a repair dialogue:

EXAMPLE PROBLEM	EXAMPLE RESPONSE
can't interpret utterance	"I don't understand the word open"
can't make sense of utterance	"There is no closed door in this room"
can't execute required action	"I can't open the door because it's locked"

Our dialogue agents of either sort must embody conventions for cooperative dialogue interaction. We offer as one example our own Gricean-styled maxim for multimodal dialogue interaction

Each user input must receive a response

where a Response is defined as an Action, a Reply, or both. The following table shows the options available to the responding agent. (Here "user" can be replaced with "requesting agent.")

RESPONSE := ACTION + REPORT

	ACTION	REPORT
error	nil	error message
execute command	transparent action	nil
		self narration
	inscrutible action	self narration
answer user query	verbal reply	nil

An error condition that prevents the execution of the user's request requires an error message that can potentially expand to a repair dialogue. If the action is carried out and it is transparent, then the agent has the option of generating no report but it can also provide a self-narrative utterance, e.g., "I have now closed the east door." If the action is carried out but it is not observable by the user then a report is required, "All of your files have been successfully deleted." or "ok, I'm thinking of a number between one and ten." Finally, if the request was a query for information and if that information is conveyed in a verbal reply, there is no need to narrate that action.

Foundational Literature

The three-tiered discourse system was developed as an implementation of two equivalent theoretical semantic frameworks, File Change Semantics (Heim, 1982) and Discourse Representation Theory (Kamp, 1981). In the third tier, the discourse model tier, the discourse processor maintains a representation of immediate attentional focus information structured according to Grosz and Sidner (1985). The attentional focus updating process of our discourse component consults a user model designed by Goodman (personal communication). The user model records a classification of geographical information (often realized as display icons) in which users of different types are likely to take interest. Based on this record, the discourse component can determine whether a new icon for, say, a lighthouse which appears as the result of panning the map display, ought to be placed in focus for the current user. Its presence at the top of the focus stack makes it available to the discourse component for interpretation of subsequent context-dependent user input such as "zoom in on that lighthouse" or an otherwise ambiguous mouse click (see Wahlster, 1987) in that region of the map. The introduction of mentions into the surface form tier follows the heuristics defined by Webber (1978).

Bibliography

[1] B Grosz, C Sidner (1985), 'The Structures of Discourse Structure', Technical Report, CSLI-85-39 Center for the Study of Language and Information.

[2] I Heim (1982), 'The Semantics of Definite and Indefinite Noun Phrases', Doctoral thesis, University of Massachusetts, USA.

[3] H Kamp (1981), 'A Theory of Truth and Semantic Representation,' in J Groenendijk et al Eds., Formal Methods in the Study of Language.

[4] S LuperFoy (1991), 'Discourse Pegs: An Analysis of Context-Dependent Referring Expressions and its Application to Human-Machine Dialogue, Doctoral thesis, University of Texas at Austin, USA.

[5] S LuperFoy (1992), 'A Three-tiered Model of User Interface Dialogue', Proceedings of annual meeting, Association for Computational Linguistics.

[6] C Sidner (1979), 'Towards a Computational Theory of Definite Anaphora Comprehension in Discourse', Doctoral thesis, Massachusetts Institute of Technology, USA.

[7] B Webber (1978), 'A Formal Approach to Discourse Anaphora', Doctoral thesis, Harvard University, USA.