The WITAS Multi-Modal Dialogue System I

Oliver Lemon, Anne Bracy, Alexander Gruenstein
Stanley Peters

CSLI, Stanford University
California, USA
lemon,bracy,alexgru,peters@csli.stanford.edu

Abstract

We present the first demonstration version of the WITAS dialogue system for multi-modal conversations with autonomous mobile robots, and motivate several innovations currently in development for version II. The human-robot interaction setting is argued to present new challenges for dialogue system engineers, in comparison to previous work in dialogue systems under the travel-planning paradigm, in that dialogues must be asynchronous, mixed-initiative, open-ended, and involve a dynamic environment. We approached these general problems in a dialogue interface to the WITAS robot helicopter, or UAV ('Unmanned Aerial Vehicle').[1] We present this system and the modelling ideas behind it, and then motivate changes being made for version II of the system, involving more richly structured dialogue states and the use of automated reasoning systems over task, ability, and world-state models. We argue that these sorts of enhancement are vital to the future development of conversational systems.

1. Dialogues with mobile robots

Many dialogue systems have been built for use in contexts where conversational interactions are predictable and can be scripted, and where the operating environment is static. For example, a dialogue for buying an airline flight can be specified by way of filling in certain parameters (cost, destination, and so on) and a database query, report, and confirmation cycle. This 'travel-planning' paradigm has been the focus of dialogue system research for several years. In such cases it suffices to develop a transition network for paths through dialogue states to recognisable completion states. Even if the database which the system accesses is dynamic (i.e. the information recorded there changes), the complexity of the interactions needed to support task completion in such an interface is low in comparison to that required to interact with a mobile agent with its own perceptions, in a changing world. Consider an operator's conversation with an autonomous mobile robot with perceptions in a environment which is constantly changing. Dialogues with such a device will be very different (see e.g. arguments in [1]) to those in the travel-planning paradigm. There will be no predictable course of events in the dialogues. The device itself may 'need' to communicate urgently with its operator. There may not be well-defined endpoints to conversations, and relevant objects may appear and disappear from the operating environment. In particular, different 'threads' of a conversation may need to be initiated, set aside, and revisited, and operator and robot will need to negotiate the robot's abilities in any situation. Tasks given to the device will also need to be specified, ordered, and their execution monitored.

The dialogue modelling and management techniques developed under the travel-planning paradigm are not rich enough to support these more complex interaction scenarios, and we have found that different structures and methods need to be adopted. We discuss modelling and implementation of structures supporting these more complex conversational capabilities in in Sections 3 and 4.

1.1. The WITAS UAV

The WITAS UAV is a small autonomous helicopter with onboard planning and deliberative systems, and vision capabilities (for details see e.g. [2]). Mission goals are provided by a human operator, and the planning system then generates a list of suitable waypoints for the UAV to navigate by. An on-board active vision system interprets the scene or focus below to interpret ongoing events, which the dialogue system reports (via the Semantic-Head-Driven NL generation capability of GEMINI) to the operator.

2. Dialogue System Architecture

UAV scenarios clearly present a number of challenges to designers of a dialogue system. In particular the dynamic operating environment and the asynchronous, mixed-initiative nature of the dialogues require a particularly flexible architecture – one which can coordinate multiple asynchronous communicating processes. For these reasons we currently use the Open Agent Architecture (OAA2) [3]. The core of the architecture is OAA's 'facilitator' which manages message passing between a number of software agents which are specialists in certain tasks, for example speech recognition, database queries, or graphical display. In our system there are six main agents[2] each responsible for various subtasks in the dialogue system (see Figure 2):

- Speech Recogniser: a wrapper to a Nuance speech recognition server[3] using a language model compiled directly from our Gemini grammar for UAV operator dialogues.
- NL ("natural language"): a wrapper to the SRI Gemini parser and generator [4] using our UAV grammar (version 1).

[1] Wallenberg laboratory for research on Information Technology and Autonomous Systems (WITAS) Unmanned Aerial Vehicle, under construction at Linköping University, Sweden. This research was funded under the WITAS Project by the Wallenberg Foundation, Sweden.

[2] All are implemented in Java, but for the NL agent (Prolog).
[3] Nuance: *www.nuance.com*

Figure 1: A demonstration of dialogue system I

Multi-modal Utterances	**Dialogue Moves**
Operator (O): *Where are the buildings?*	Raise question
UAV (U): *Here.* [displays buildings on GUI]	Answer question
O: *Go to this one.* [click]	Deictic command
O: *Fly to Main Street.*	Command
O: *No, I meant Circle Road.*	Elliptical revision
U: *Okay.* [Route changed]	Confirmation backchannel
O: *Hide the buildings.*	GUI Command
U: [buildings disappear from GUI].	
O: *Next, continue to the bridge and then here* [click].	Complex command with deictic
O: *Then look at the parking lot, the hospital, and the building.*	Ambiguous command
U: *Where is the parking lot?*	Presupposition check, raise question
O: [click]	Graphical ambiguity resolution
U: *Okay.*	Confirmation, update database
U: *Which building do you mean?* [buildings highlighted on map]	Presupposition check, raise question
O: *The airport.*	Verbal ambiguity resolution
U: *The airport.*	Confirmation backchannel
O: *Go to the ground station, hover here* [click], *and go back to it.*	Anaphoric and deictic command
O: *Then land at the parking lot.*	Command, previously added referent
O: *No, not the parking lot, the waypoint.*	Revision
U: *Which waypoint do you mean?* [waypoints highlighted on GUI]	Presupposition check, raise question
O: *This one* [click].	Clarification
U: *Okay.* [Route changed]	Confirmation backchannel
O: *Plan the route.*	Command
U: *Planning Route. Route planned.* [Route displayed on GUI]	Confirmation backchannel
O: *Roger. Proceed.*	Command
U: *Executing route.*	Confirmation backchannel
U: *Way-point two reached.*	UAV report
U: *Truck 8 is turning left onto Circle Road.*	UAV report generation
U: *The truck is passing the warehouse.*	UAV report generation
O: *Follow it.*	Anaphoric reference to UAV's NP

- TTS ("text-to-speech"): a wrapper to the Festival 1.4.1 speech synthesiser[4]
- GUI: a map display of the current operating environment which displays route plans, waypoints, locations of vehicles including the UAV, and allows gesture input by the operator, see Figure 3.
- Dialogue Manager (version 1): responsible for coordinating multi-modal inputs from the user, interpreting dialogue moves made by the operator and UAV, updating and maintaining the dialogue context, handling UAV reports and questions, and sending speech and graphical outputs to the operator (see Section 3).
- Robot Control and Report: the software responsible for translating commands and queries from the dialogue interface into commands and queries to the UAV, and vice-versa for reports and queries received from the UAV. We currently interface to a simulated version of the UAV, using a real-time CORBA communication layer.

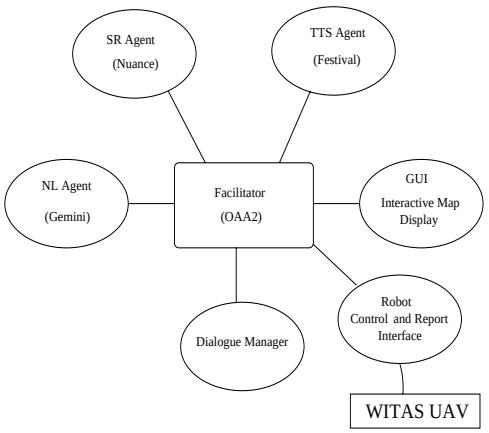

Figure 2: Dialogue system architecture

The operator's speech is recognized by Nuance and parsed into logical forms by Gemini. If these forms do not already indicate the speech act of the user, the dialogue manager inspects the current dialogue Information State (see Section 3) to determine how best to incorporate the utterance into the dialogue. Because Gemini offers a complete mapping from sentences to logical forms, logical forms may also be used in Semantic-Head-Driven Generation of English sentences for the UAV's queries and reports. Reports are passed to Gemini via the Robot Controller and on to Festival, which synthesizes the UAV's speech. The dialogue segments shown in Figure 1 illustrate many of the capabilities of the multi-modal dialogue interface. They can be carried out in continuous sequence using spoken voice input and mouse clicks on a map screen. Videos of such interactions can be found at `www-csli.stanford.`

[4]Edinburgh University, Centre for Speech Technology Research

edu/semlab/witas/demo1/.

Variants of some of these components have been used in other dialogue systems, notably SRI's CommandTalk [5], the NASA Personal Satellite Assistant [6], and SRI's [7]. However, our system stands apart from these in its particular combination of complex dialogue capabilities (including Natural Language generation) with multimodality over a dynamic operating environment. The core of our system, and its most innovative feature, is the dialogue manager, described in more detail below.

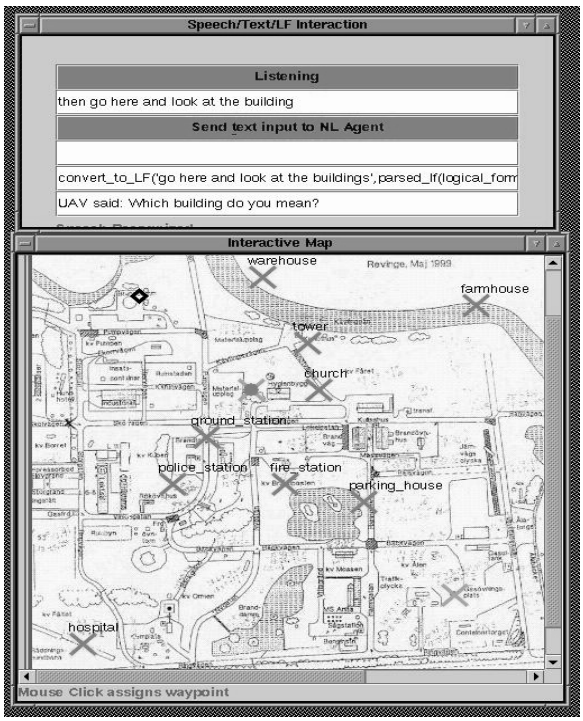

Figure 3: Part of the Graphical User Interface

3. The Dialogue Manager, version I

Our dialogue manager embodies several recent theoretical ideas in dialogue modelling. Essentially it creates and updates an *Information State* (IS) corresponding to a notion of dialogue context. Dialogue moves have the effect of updating information states, and moves can be initiated by both the operator and the robot. A dialogue move might cause an update to the GUI, send an immediate command to the UAV, elicit a spoken report, or prompt a clarifying question from the UAV. Subdialogues can be arbitrarily nested.

Central parts of these information states are an *IR stack* – a stack of public unresolved *issues raised* in the dialogue thus far, and a *UAV Agenda* – a private list of issues which the UAV has yet to raise in the conversation. Under certain conditions, items from the UAV Agenda are made public by an utterance from the UAV (e.g. "Which building do you mean?"), moving the issue onto the IR Stack. Such an operation is a Dialogue Move (in this case by the UAV). The dialogue manager contains a collection of rules which interpret (multi-modal) input from both operator and UAV as dialogue moves with respect to the current information state, and update the state accordingly. Similarly, there are rules which process UAV responses, reports, or questions, again updating the context accordingly.

Logical-form outputs from the parsing process are often already interpreted as speech acts of various kinds (e.g. "Fly to the hospital" is parsed as a COMMAND). Certain utterances do not have a specific illocutionary force, and these are simply specified as DECLARATIONs. The dialogue manager then decides, on the basis of the current IS, what speech act such utterances constitute. This is akin to the robust parsing strategy described in [8].

Another important part of the information state is a *Salience List* consisting of the objects referenced in the dialogue thus far, ordered by recency (see e.g. [9]). This list also keeps track of *how* the reference was made (i.e. by which modality) since this is important for resolving and generating anaphoric and deictic expressions in the dialogues.

A related structure, the *Modality Buffer*, keeps track of mouse gestures until they are either bound to deictic expressions in the spoken input or, if none such exists, are recognised as purely gestural expressions. There are other aspects of updating the dialogue context which are database maintenance tasks.

To recap, in version I of the system our Information States consist of:

- Issues Raised (IR) stack
- UAV agenda
- Salience list
- Modality buffer
- Databases: dynamic objects, planned routes, geographical information, names.

Note that dialogue capabilities can be added in a modular way, due to the structure of dialogue Information States.

4. Version II: Dialogue Move Trees, Task Trees, and Automated Reasoning

Many aspects of human conversational skill are missing from state-of-the-art dialogue systems, even those which do not implement simple state-transition networks (c.f. [1]). While some progress has been made in capturing structural aspects of human conversational skill, much further research and development is needed to understand the structures, knowledge, and algorithms needed to support conversation.

One of our first observations has been that the adoption of *stack* structures to drive dialogue move processing (see e.g. Section 3) is too restrictive in general. In particular it has made navigation back and forth between different sub-dialogues and topics impossible (since some information is lost when issues are popped off the IR stack). Version II of our system uses a tree structure of dialogue states (a *dialogue move tree*) , where edges are dialogue moves, and branches represent conversational threads. This structure allows for more robust dialogue management.

Another development avenue is to provide richer domain knowledge and inference methods for the dialogue manager. For instance, we have implemented a dynamic hierarchical *task-tree* (c.f. [10]), which grows as part of the developing dialogue context, and represents tasks and sub-tasks described by the operator and their temporal ordering. This structure allows reordering and reference to tasks (c.f. the salience list described above). For example "Go to the tower. Show me car 1. Actually, do that first."

We have also implemented an inference-based model of the robot's changing abilities, depending on dynamic information

about the world and the robot's internal state and location. (See [11] for a survey of the uses of automated theorem proving in computational semantics.) This, again, is part of the context which the dialogue manager has access to. The implementation uses the KIF knowledge representation scheme, and inference steps are carrried out using the Java Theorem Prover[5]. This module enables dialogues of the following kind (there is an inference rule stating that the UAV is not allowed to fly to buildings that are on fire):
O: "Where can you fly to?"
U: "I can fly to the tower, the temple, and the hospital."
O: "The tower is on fire. Fly there"
U: "I cannot fly to the tower. I can go to the temple and the hospital"

A similar reasoning and representation module handles information about default operating conditions and operating modes (e.g. "Fly in safe mode at high altitude"). These inference modules are obviously specific to a particular application, but they allow all domain-specific information to be removed from the dialogue manager's knowledge about dialogue moves and structures. Thus, use of these representation and reasoning modules allows the development of a domain-independent dialogue manager (see also [10]).

More such modules are planned for version II, to track and maintain consistency of conversational common-ground (the facts as established in the dialogue) and as user models.

5. Conclusion

We argued that dialogue systems for robots require a qualitative leap in the complexity of dialogue models and managers for dialogue systems. We then described version I of the WITAS dialogue interface, which addresses some of these issues, and described futher innovations in development for version II of the system.

Dialogues supported by the current interface are mixed-initiative, open-ended, and multi-modal, over a dynamic operating environment. A general point of distinction between our system and many others is that it is not restricted to plan-based dialogues. In other words, paths through dialogues need not be specified in advance, as is necessary in many other systems. Our approach based on updates over IS allows us to be much more flexible in the way we process dialogues.

To reiterate, our current system has the following features:

- successfully interfaced to real-time UAV simulator
- commands, questions, revisions, reports, over a dynamic environment
- asynchronous, real-time, multi-modal, mixed-initiative, open-ended dialogues
- Semantic Head-Driven Generation of robot reports
- employs a dynamic information state model of dialogue
- Solaris or Windows NT/2000 implementations available

The WITAS multi-modal dialogue interface interprets spoken language and map-gesture inputs as commands, queries, responses, and declarations to the UAV, and generates synthesized speech and graphical output to express the robot's responses, questions, and reports. Current dialogue capabilities include ambiguity resolution, presupposition checking, processing of anaphoric and deictic expressions, command revision, report generation, and a confirmation backchannel. Our central innovation is a general-purpose dialogue manager which implements a dynamic information state model of dialogue.

Videos of demonstrations of version I of the system (laptop version) are available at `http://www-csli.stanford.edu/semlab/witas/demo1/`

Innovations in version II of the system concern dialogue move trees, task trees, and the use of automated reasoning modules to handle application-specific aspects of negotiation of tasks, resources, and abilities in conversations.

[5] JTP: `http://ksl.stanford.edu/software/jtp/`

6. References

[1] Renee Elio and Afsaneh Haddadi, "On abstract task models and conversation policies," in *Workshop on Specifying and Implementing Conversation Policies, Autonomous Agents'99*, Seattle, 1999.

[2] Patrick Doherty, Gösta Granlund, Krzystof Kuchcinski, Erik Sandewall, Klas Nordberg, Erik Skarman, and Johan Wiklund, "The WITAS unmanned aerial vehicle project," in *European Conference on Artificial Intelligence (ECAI 2000)*, 2000.

[3] David Martin, Adam Cheyer, and Douglas Moran, "The Open Agent Architecture: a framework for building distributed software systems," *Applied Artificial Intelligence: An International Journal*, vol. 13, no. 1-2, 1999.

[4] John Dowding, Jean Mark Gawron, Doug Appelt, John Bear, Lynn Cherny, Robert Moore, and Douglas Moran, "GEMINI: a natural language system for spoken-language understanding," in *Proc. 31st Annual Meeting of the ACL*, 1993.

[5] Amanda Stent, John Dowding, Jean Mark Gawron, Elizabeth Owen Bratt, and Robert Moore, "The CommandTalk spoken dialogue system," in *Proceedings of the Thirty-Seventh Annual Meeting of the ACL*, University of Maryland, College Park, MD, 1999, Association for Computational Linguistics, pp. 183–190.

[6] Manny Rayner, Beth Ann Hockey, and Frankie James, "A compact architecture for dialogue management based on scripts and meta-outputs," in *Proceedings of Applied Natural Language Processing (ANLP)*, 2000.

[7] Didier Guzzoni, Adam Cheyer, Luc Julia, and Kurt Konolige, "Many robots make short work," in *AAAI Robotics Contest*, Menlo Park, CA., 1996, SRI International, AAAI Press.

[8] James F. Allen, Bradford W. Miller, Eric K. Ringger, and Teresa Sikorski, "A robust system for natural spoken dialogue," in *Proceedings of ACL*, 1996.

[9] John Fry, Hideki Asoh, and Toshihiro Matsui, "Natural dialogue with the Jijo-2 office robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS-98*, Victoria, B.C., Canada, 1998, pp. 1278–1283, (See `www-csli.stanford.edu/semlab/juno`).

[10] Jiang Han and Yong Wang, "Dialogue management based on a hierarchical task structure," in *Proceedings of ICSLP 2000*, 2000.

[11] Patrick Blackburn, Johan Bos, Michael Kohlhase, and H. de Neville, "Inference and computational semantics," in *Third International Workshop on Computational Semantics (IWCS–3)*, Harry Bunt and Elias Thijsse, Eds., 1999, pp. 5–21.