Words at the Right Time: Real-Time Dialogues with the WITAS Unmanned Aerial Vehicle
Extended abstract

Erik Sandewall[1], Patrick Doherty[1], Oliver Lemon[2], and Stanley Peters[3]

[1] Department of Computer and Information Science
Linköping University
S-58183 Linköping, Sweden
[2] School of Informatics
University of Edinburgh
Edinburgh EH8 9LW, Scotland
[3] Center for the Study of Language and Information
Stanford University
Stanford, Calif. 94305, USA

Abstract. The WITAS project addresses the design of an intelligent, autonomous UAV (Unmanned Aerial Vehicle), in our case a helicopter. Its dialogue-system subprojects address the design of a deliberative system for natural-language and graphical dialogue with that robotic UAV. This raises new issues both for dialogue and for reasoning in real time. The following topics have been particularly important for us in various stages of the work in these subprojects:

- spatiotemporal reference in the dialogue, including reference to past events and to planned or expected, future events
- mixed initiative in the dialogue architecture of a complex system consisting of both dialogue-related components (speech, grammar, etc) and others (simulation, event recognition, interface to robot)

and more recently as well

- identification of a dialogue manager that is no more complex than what is required by the application
- uniform treatment of different types of events, including the robot's own actions, observed events, communication events, and dialogue-oriented deliberation events
- a logic of time, action, and spatiotemporal phenomena that facilitates the above.

This paper gives a brief overview of the WITAS project as a whole, and then addresses the approaches that have been used and that are presently being considered in the work on two generations of dialogue subsystems.

1 The WITAS Project: Goals, Structure, and the WITAS System

The WITAS Unmanned Aerial Vehicle Project [14, 15] is an ambitious, long-term basic research project whose main objectives are the development of an integrated hardware/software VTOL (Vertical Take-Off and Landing) platform for fully autonomous missions and its future deployment in applications such as traffic monitoring and surveillance, emergency services assistance, photogrammetry and surveying. Basic and applied research in the project covers a wide range of topics that include both the development of traditional AI technologies, core functionalities and their pragmatic integration with other technologies in a prototype experimental UAV system.

The following is a non-exclusive list of some of the activities in the project:

- Development of a generic distributed deliberative/reactive software architecture for (aerial) robotic systems.
- Development of a helicopter control system with flight modes for stable hovering, takeoff and landing, trajectory following, and reactive flight modes for interception and tracking of vehicles.
- Development and integration of numerous AI technologies. These include both path and task-based planning systems, a chronicle recognition system for identifying complex vehicular patterns on the ground, and other high-level services for reasoning about action and change.
- Development and integration of numerous knowledge representation technologies. These include an on-board geographical information system; a dynamic object repository to anchor, manage and reason about dynamic objects such as vehicles discovered during mission execution; a qualitative signal processing framework for dynamic construction of trajectories and histories of world behavior with associated reasoning mechanisms; and development of knowledge structures for signal-to-symbol conversion and approximate reasoning in soft real-time.
- Development of an on-board dynamically programmable image processing system.
- Development of multi-modal interfaces (including dialogue) for ground operator/UAV communication with both speech generation and recognition capability.
- Development of simulation environments with hardware-in-the-loop for both testing and visualization of control, reaction and deliberation functionalities.

Each of these activities is expected to contribute both to the construction of the project's integrated demonstrator, and by development and publication of specialized basic research results in their respective areas. The main demonstrations of the integrated WITAS demonstrator are due before the end of 2003.

The VTOL platform used in the project is a slightly modified Yamaha RMAX helicopter manufactured by Yamaha Motor Company. It is commercially available in Japan as a radio-controlled platform. The RMAX is approximatively 2.7 x 0.7 meters, with a main rotor 3 meters in length.

The *WITAS System* is developed under the direction of Patrick Doherty. It includes on-board facilities for autonomous control, sensor data interpretation and integration, a system for reactive task procedures, trajectory planning, high-level action planning, ground-to-UAV communication, and others more. The *WITAS*

System uses a distributed architecture [15] that is based on real-time CORBA, and which facilitates reconfiguration of the system as well as integration of additional facilities and interface to other and remote services.

Although the development of the *WITAS System* requires the integration of a large number of subsystems and technologies, a few of the facilities for the intended final demonstrator are developed as separate subprojects while planning for eventual integration across well defined interfaces. This applies in particular for the advanced vision facilities that are being developed within the project by Gösta Granlund and his group, and for the natural-language dialogue facilities that are the topic of the present paper. The *WITAS Demonstrator* will consist of the *WITAS System* (its major part), a dialogue system, and optionally an augmented vision system.

The *WITAS System* has been tested in actual flights at a considerable number of occasions, but the integration of the other subsystems to form the *WITAS Demonstrator* is in progress at the time of writing (June, 2003) and has not yet been tested in flights.

Before proceeding to the dialogue system, a brief overview of the *WITAS System* is appropriate since it sets the context within which the dialogue is performed.

The first-generation *WITAS system* was developed during 1997 and 1998, and operated entirely in a simulation environment. After an intermediate period of analysis and redesign, work started in year 2000 on the present generation of WITAS system which is designed to be used in real flights, with the resulting requirements on reliability and on real-time performance.

A great deal of effort has gone into the development of a control system for the WITAS UAV which incorporates a number of different control modes and includes a high-level interface to the control system. This enables other parts of the architecture to call the appropriate control modes dynamically during the execution of the mission. The ability to switch modes contingently is a fundamental functionality in the architecture and can be programmed into the 'task procedures' associated with the reactive component of the architecture. At the time of writing we have developed the following control modes which are used in actual flights on a regular basis:

- hovering (H-mode)
- trajectory following (TF-mode)
- proportional navigation (PN-mode)

PN-mode is a reactive flight mode for interception and tracking. Additional take-off and landing modes are in an early testing stage at present.

A number of fully autonomous flights have been demonstrated successfully in tests during 2002 and 2003. These include stable hovering, predefined 3D trajectory following including 360 degrees banked turns, vehicle interception and road following. Acceleration and braking with no overshoot have been tested at speeds of 55 km/h, and coordinated banked turns have been tested with a turn rate of 20 degrees/second. We have completed autonomous missions of up to a duration of 20 minutes using a combination of control modes in addition to interaction with a ground operator.

One of the more sophisticated missions flown of this character is based on an emergency services scenario where a train carrying a bio-chemical ingredient has collided with a vehicle and contaminated a small region around the train which partially intersects a small town. The goal of the mission is to interactively fly to specific points in the contaminated region and provide initial video sequences of injured inhabitants at these points. Successful completion of the mission involves repeated use of H- and TF- modes, on-line trajectory planning to avoid colliding with building structures in the region, and command-mode communication from the ground to inform the UAV of new coordinates to fly to.

A number of knowledge representation techniques have also been developed which are currently being integrated with the UAV software architecture. These techniques have been developed to meet the specific constraints and needs associated with the type of missions flown by the UAV. These involve such issues as real-time querying of knowledge structures, integration of quantitative and qualitative representations, incremental refinement of existing knowledge structures through machine learning techniques and techniques for defining tolerance and similarity measures on primitive and complex data structures. Details concerning these and other knowledge representation related tools and techniques may be found in the following publications, [12, 11, 10, 4, 8, 9, 13].

2 Real-Time Dialogue in WITAS: First Stage

Although the WITAS project started in 1997, we did not begin to address the dialogue problem until in 2000. During the first three years the project was only focused on the problem of fully autonomous flight for achieving a goal that had been set for the UAV before the start of the flight.

The dialogue subproject was added to WITAS in response to both an obvious practical need in the applications being considered, and for purely scientific reasons. Natural-language and multimedia dialogue with an autonomous robot is a challenging research problem which introduces several important issues that are not present in, for example, dialogue with a database or a service provider such as an automated travel agency. Dialogue with an intelligent robot must be able to refer to phenomena along a time-line, including both what has happened in the robot's own actions and in its environment, what the robot plans to do, and what events can be predicted in the environment. The dialogue software must be able to interleave planning and execution-related dialogue threads, and needs to be able to coordinate several simultaneous tasks of the robot (see [22]).

Furthermore, the dialogue itself is located on the same time-line as the robot's actions, so that speedy and concise expressions, break-ins, and other time-aware aspects of dialogue are not only important for user convenience, they are in fact essential for the success of the dialogue and, eventually, of the robot's mission. Meeting these demands sets strong requirements on both the representational power, the deliberative capability, and the computational performance of the dialogue system.

During three years, 2000 to 2002, the dialogue subproject of WITAS was conducted by Stanley Peters and Oliver Lemon at the Center for the Study of Language and Information (CSLI) at Stanford university. Oliver Lemon was the technical project leader. This group developed a first-generation *WITAS Dialogue System*, a stand-alone system that conducts a spoken dialogue with a simulated UAV performing a flight mission over a small town.

The system supports dialogues about tasks such as locating and following vehicles, delivering objects to locations, and flying to locations. It allows specification of local and global constraints on tasks (e.g. "always fly high" and negotiation subdialogues about constraint violations (e.g. "I am supposed to always fly high. Shall I fly to the lake at low altitude anyway?"). It uses open-microphone speech recognition in an asynchronous architecture and has the ability to resolve multimodal inputs (e.g. "Fly here" [user points to map location]). Speech recognition language models are set dynamically, depending on dialogue state, resulting in significant improvements in recognition and concept error rates.

Dialogue management is based on the "Information State Update" approach (see e.g. Bohlin et al., [2]) with a clean separation of task and dialogue knowledge. The following are the components of the information state of this system:

- A *dialogue move tree* that represents the structure of the dialogue by way of conversational 'threads', composed of the dialogue moves of both participants (e.g. *command*, *wh-question*), and their relations;
- An *activity tree* that represents hierarchically decomposed tasks and plans of the UAV, and their states (e.g. planned, current, failed, suspended, cancelled);
- A *system agenda* that represents the planned dialogue contributions of the system and their priorities (e.g. report task failure, ask clarification question);
- A *pending list* that represents open questions raised in the dialogue;
- A *salience list* that contains a priority-ordered list of objects that have been mentioned in the dialogue, and a record of how they have been referred to. It is used for identifying the referents of anaphoric and deictic expressions and for *generation* of contextually appropriate referring expressions (see [21]);
- A *modality buffer* which records gesture inputs from the user (in the form of mouse clicks) for multimodal fusion with linguistic referring expressions.

This design uses domain-independent dialogue move classes and is intended to be applicable in general for dialogues between humans and autonomous real-world systems. Central components of the system have been reused in other domains and genres, for example tutorial dialogues [3].

The WITAS project uses the Revinge training area as its standard flight-test area for advanced flights; Revinge is small town of about 0.6 x 0.8 km which is operated by the Swedish Civil Rescue Service as a training grounds. The flight simulations in the *WITAS Dialogue System* were conducted in a simulation of Revinge which had been prepared in the main (Linköping) part of the project.

This stage of the project developed a system with the dialogue management architecture that has been briefly outlined here, with facilities for:

- multitasking
- interleaved task planning and execution
- constraint negotiation
- revision and repair utterances
- multimodal interpretation and generation
- dynamic language models
- domain-specific task models, reusability

The dialogue competence of the system includes also a number of situation types that the actual UAV can not perform at present, in particular because of the restricted image interpretation capability.

An evaluation of the system was conducted, measuring task completion rates for novice users with no training (see Hockey et al, [17] for full details). We found that 55% of novice users where able to complete their first task successfully, rising to 80% by the fifth task. (An example task is "There is a red car near the warehouse. Use the helicopter to find it. Then land the helicopter in the parking lot." - note that the user can not simply read these task statements out to the system). Performance was significantly improved via the use of a targeted help system in the dialogue interaction.

Besides the normal international publications of these results, the system has also been demonstrated to a variety of audiences. See Lemon et al in [20, 22, 17, 21] for full details.

These results having been achieved, the next step for the project was to integrate the dialogue system with the *WITAS System* that was described in section 1, in order to realize a demonstrator capable of flight control with high-level autonomy, computer vision, and natural-language dialogue between operator and UAV. The integration is done in the Linköping section of the project. It is first of all a practical software challenge, but actually it has raised some more principled issues as well, having to do with the precise treatment of time in a dialogue system. The rest of the present paper is devoted to a report of our current, on-going work in that respect.

3 Reasoning about Action and Change in WITAS

Two of the present authors (Sandewall and Doherty) have been active in research on reasoning about actions and change since many years. When we started the WITAS project in 1997 we expected that this background would become an important technique on the deliberative level in the eventual WITAS demonstrator. However, many other problems had to be dealt with first, in both the first and the second generation of the WITAS system, and the research on actions and change has been a concurrent activity in the project whose integration is only beginning to take place.

Patrick Doherty and his group have developed TALplanner, a forward-chaining planner that relies on domain knowledge in the form of temporal logic formulas in order to prune irrelevant parts of the search space. Details concerning TALplanner may be found in the following publications: [5, 18, 6, 7]. The core engine used in TALplanner is currently being extended for use as both a predictive component and an on-line execution monitor in the UAV architecture.

Although the dramatic improvement in performance of the planner is the most striking aspect of the TALplanner work, there have also been several other developments that are as well important for the overall project. Jonas Kvarnström has developed systematic domain analysis techniques for domain-dependent control [19]. Joakim Gustafsson and Jonas Kvarnström have extended the basic TAL formalism with constructs for object-orientation and shown how they provide elaboration tolerance in several standard testing domains in the literature [16]. Tests using the research software tool VITAL (which is related to TALplanner) showed performance that was much better, often by one or more orders of magnitude, compared with another published tool, the Causal Calculator of Vladimir Lifschitz and his group [1].

During the same period, Erik Sandewall has worked on semantic issues for RAC and, in particular, on the semantic relationship between the world where actions and change take place on one hand, and the logic formulas the represent that action and change on the other hand. In [26] he described an idealized architecture for a cognitive robot that relates actions and change with the formulas describing them, in two ways: the architecture uses such formulas as its data objects, and the formulas characterize what happens when the architecture executes. (This abstract architecture must not be confounded with the actual architecture in the *WITAS System* as described in [15]; they are different things and serve different purposes).

Continuing the same track, [27] proposed a formal characterization of the act of decision, that is, the transition from a situation where an agent has deliberated over several possible courses of action for the future, to the situation where the agent has chosen one of them and it has become a part of its intentions for the immediate future.

Sandewall's work has been using a logic of actions and change called CRL (for Cognitive Robotics Logic) [25] which is similar to TAL (for Time and Action Logic). They have a common ancestry in the analysis of nonmonotonic entailment methods for actions and change in 'Features and Fluents' [24].

4 Real-Time Dialogue in WITAS: Second Stage

During the year 2003 we are integrating the results from the Stanford subproject so that it can operate in conjunction with the *WITAS System*. At the same time, Erik Sandewall and his group is building a next-generation dialogue system called the *DOSAR system* that builds on the experience and the design from the first generation. However, the new system uses a different approach to representing time and action which is based on a formal logic of time and action, and which is closely related to a high-level simulator of helicopter actions. The group also hopes to further improve the robustness of the dialogue so that misunderstandings occur less frequently and so that they can be resolved efficiently. This work makes use of the background on reasoning about actions and change that was described above.

The acronym 'DOSAR' stood originally for 'Dialogue-Oriented Simulation And Reasoning' since it started as a logic-based simulator. It was first written in order to provide an environment in which to test the WITAS dialogue system in its early stages of development, and as an experimental implementation of the double helix architecture [26]. However, we found after a while that the task structuring and the time management capabilities of this simulator were also useful as a platform for a dialogue manager.

Our approach is to represent several kinds of actions in a uniform manner and using Cognitive Robotics Logic, namely:

- physical actions by the helicopter as a whole (take off, land, fly to X, etc) and by its various components such as its video camera system
- speech acts and other communication acts, as performed by both the operator and the dialogue system
- cognitive acts, such as parsing a sentence, or deciding what to say next.

One reason for representing these types of actions in a uniform way is for conceptual economy, in particular since similar logical constructs arise for all of them. Another reason is to prepare for dialogue where the user makes combined reference to several types of actions, for example "where were you when I told you to fly to the parking garage?"

All these types of actions may fail, and such failure is important for the dialogue management. Actual or potential failure of physical actions is obviously a topic of discussion. Failure of the cognitive acts is equally relevant since it typically leads to misunderstandings that need to be resolved in the further dialogue. In order to analyze and characterize the failure of these cognitive actions, they have to be decomposed into more elementary subactions that represent successive steps in the actual software, for example, a simple sequence of speech understanding, parsing, and so forth. Each of these steps may fail, and each kind of failure will be reflected in the continued dialogue.

The *WITAS Dialogue System* demonstrated the importance of properly managing faults in the cognitive actions that are involved in the dialogue. The *DOSAR System* therefore takes a further step and introduces the use of a logic of actions and change for characterizing these faults. The Cognitive Robotics Logic (CRL) which it uses, provides a concise representation for the succeed/fail distinction. It is used there for a formal characterization of goal-directed behavior in the sense of "if one method or plan for achieving the goal fails, then try another one". These aspects of the logic are part of the reasons why it was chosen as the representation language for the DOSAR dialogue system.

Although it is conceptually attractive to use a uniform representation for different kinds of actions, there are also significant differences between different types of actions, and these differences have to be reflected by the knowledge represenation. Consider, in particular, the following classification of action types:

- the *speech act proper*, that is, the period of time where the phrase is actually uttered
- the *understanding of the phrase*, which may consist of sub-actions such as speech understanding (speech to text), parsing, identification and screening of semantic content, etc.
- the *decision how to react*, where the reaction may be, for example, to answer a question, to send a command to the helicopter, or to verify that a command has been correctly understood before it is sent to the helicopter.

Both our generations of dialogue systems use a classification more or less along these lines. All of these steps are specific to the particular received sentence. In terms of the information-state structure that was described above, the third step also includes attaching the current phrase, or its 'semantic' representation, to the dialogue move tree. In some cases the phrase or its derivatives are attached to other parts of the information state as well.

However, these types of actions are also different in two important aspects. First, the time scale: physical actions in a helicopter take from a few seconds to tens of seconds (rarely more, in our confined testing area). The speech acts proper often take several seconds as well. However, the understanding of the phrase, and even more its distinct subactions is performed more quickly than that, which of course is a necessity since a delay of several seconds between successive dialogue moves would be very annoying. The same applies for the decision how to react, although of course the decision may have been anticipated in deliberations that took place earlier in the course of the dialogue. This requires the representation of actions to work with two different time scales.

The other difference concerns the effects of actions. Physical actions have effects on the state of the world, which is modeled using objects and features (properties) of objects in our system. The cognitive actions, on the other hand, can better be thought of as transformers that convert one symbolic expression (for example, a parse tree) to another symbolic expression, of another kind. Traditional formalisms for actions and change are not well adapted to represent actions of this kind.

We are extending the CRL formalism in order to cope with these new demands. Since our goal here is to give an overview of the project, and due to the format of the present conference proceedings, it is not possible to describe the specifics of the chosen formal system, and we have to refer to forthcoming, more extensive publications. Please use the WITAS webpage `www.ida.liu.se/ext/witas` for an up-to-date list of publications.

5 Connections Between Dialogue System and UAV

Both generations of dialogue managers (the one developed at Stanford, and the one presently developed at Linköping) will be connected to the *WITAS system* for the demonstrations. However, for safety reasons this can not be a direct connection, and there must be checkpoints in-between. Technically, it works as follows. The UAV communicates with a *UAV ground station* using three independent communication channels. First, there is a basic control channel for direct remote-control of the UAV, which is provided by the manufacturer and which is essential for safety reasons as a backup. Secondly, there is a two-directional data link for downloading measurement data and for uploading commands to the helicopter. Thirdly, there is a video downlink so that a person on the ground can see what the helicopter sees. (The computer vision system of the UAV is on board, however, like all its other control systems).

The dialogue system, on the other hand, consists of a *mobile terminal* that the operator can carry with him or her, and a *dialogue ground station* that is located adjacent to the UAV ground station. The mobile terminal is a laptop or tablet running the dialogue system software, which may be both the *WITAS Dialogue System* from Stanford and the more recent *DOSAR system*, together with the necessary audio and radio equipment and power supply. The dialogue ground station is a cart carrying two computers, one for real-time video recording and one for interfacing and for supervision of experiments.

When the operator issues a command to the helicopter, then the command phrase is interpreted in the mobile terminal; a command message is sent to the dialogue ground station, from there on to the UAV ground station where it is checked by a supervisor, and only if it passes that check is it actually transmitted to the UAV for execution.

The downloaded video signal is recorded in the dialogue ground station, in such a way that it can be referred to in the subsequent dialogue; the operator can request to see the video segment containing a particular event that is referred to verbally.

This relatively complex configuration is required in order to deal first of all with the application as such and the safety issues during test flights, but also because of the diversity of signal types and the bandwidth limitations. It leads to several interesting problems of synchronization and other time management, in particular in the continued development of the dialogue system. Consider, for example, the situation where an operator wishes to identify a moving object by pointing into a running video that is shown on the display screen.

Besides the transmission of commands from the dialogue system to the UAV, there are also several other facilities in the interface between the *WITAS system* and the dialogue systems. These are presently being implemented for the interface to the *DOSAR System*, and the interface to the *WITAS Dialogue System* will follow. There is a flow of observations from the UAV to the dialogue manager which uses the same communication lines as the commands, but in the opposite direction. Also, some of the knowledge representation technologies that were mentioned in section 1, are made accessible to the dialogue managers via the interface. This applies, in particular, for the specialized, real-time GIS system, but also for the dynamic object repository (DOR). The DOR contains dynamically updated descriptions and properties of objects on the ground that are observed by the helicopter, and may also be used for storing properties of the camera system and other subsystems within the helicopter itself.

6 Future Plans

The first author of this article foresees two important tasks in the continued work on the DOSAR dialogue system for WITAS, besides completing the integration and the final project demonstration. The continued modeling of paradigmatic dialogue situations and how they relate to the application model is a major consideration. In the architecture used by these systems, incoming phrases have to be attached to the dialogue move tree which in turn is used for determining the system's responses. The attachment rules for the Stanford system have been published in [22]. The structure of these attachment rules will be a topic of further study.

Our second goal is methodological. For continued progress in this research area, it will be important to have a means of comparing the architectures and approaches that are used in different systems, as implemented by different research groups. The techniques that have been developed for comparing and assessing different approaches to reasoning about actions and change [24] may be applicable for this domain as well.

7 Acknowledgemenets

The research reported here was supported by the Knut and Alice Wallenberg Foundation. We acknowledge the contributions of the following persons who have participated in the work on the WITAS dialogue systems,

at Stanford: Anne Bracy, Alexander Gruenstein, and Laura Hiatt;

in Linköping: Malin Alzén, Peter Andersson, Genevieve Correll, Karolina Eliasson, Coynthia Hernandez, Tomas Johansson, Felix Koch, Susanna Monemar, Egil Möller, Krister Nilsson, and Tobias Nurmiranta.

References

1. V. Akman, S. Erdogan, J. Lee, V. Lifschitz, and H. Turner. Representing the zoo world and the traffic world in the language of the causal calculator (to appear). *Artificial Intelligence*, 2003.
2. Peter Bohlin, Robin Cooper, Elisabet Engdahl, and Staffan Larsson. Information states and dialog move engines. *Electronic Transactions in AI*, 3(9), 1999. Website with commentaries: www.etaij.org.
3. Brady Clark, Oliver Lemon, Alexander Gruenstein, Elizabeth Owen Bratt, John Fry, Stanley Peters, Heather Pon-Barry, Karl Schultz, Zack Thomsen-Gray, and Pucktada Treeratpituk. A general purpose architecture for intelligent tutoring systems. In *Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems*. Kluwer, 2003.

4. P. Doherty, J. Kachniarz, and A. Szałas. Using contextually closed queries for local closed-world reasoning in rough knowledge databases. In *[23]*. 2002.
5. P. Doherty and J. Kvarnström. TALplanner: An empirical investigation of a temporal logic-based forward chaining planner. In *Proceedings of the 6th International Workshop on Temporal Representation and Reasoning (TIME'99)*, 1999.
6. P. Doherty and J. Kvarnström. TALplanner: A temporal logic based forward chaining planner. *Annals of Mathematics and Artificial Intelligence*, 30:119–169, 2001.
7. P. Doherty and J. Kvarnström. TALplanner: A temporal logic based planner. *Artificial Intelligence Magazine*, Fall Issue 2001.
8. P. Doherty, W. Łukaszewicz, A. Skowron, and A. Szałas. Combining rough and crisp knowledge in deductive databases. In *[23]*. 2003.
9. P. Doherty, W. Łukaszewicz, and A. Szałas. Computing strongest necessary and weakest sufficient conditions of first-order formulas. *International Joint Conference on AI (IJCAI'2001)*, pages 145 – 151, 2000.
10. P. Doherty, W. Łukaszewicz, and A. Szałas. Efficient reasoning using the local closed-world assumption. In A. Cerri and D. Dochev, editors, *Proc. 9th Int. Conference AIMSA 2000*, volume 1904 of *LNAI*, pages 49–58. Springer-Verlag, 2000.
11. P. Doherty, W. Łukaszewicz, and A Szałas. Cake: A computer-aided knowledge engineering technique. In F. van Harmelen, editor, *Proc. 15th European Conference on Artificial Intelligence, ECAI'2002*, pages 220–224, Amsterdam, 2002. IOS Press.
12. P. Doherty, W. Łukaszewicz, and A Szałas. On mutual understanding among communicating agents. In B. Dunin-Keplicz and R. Verbrugge, editors, *Proceedings of Workshop on Formal Approaches to Multi-agent Systems (FAMAS'03)*, pages 83–97, 2003.
13. P. Doherty, W. Łukaszewicz, and A. Szałas. Tolerance spaces and approximative representational structures. In *26th German Conference on Artificial Intelligence.* Springer-Verlag, 2003.
14. Patrick Doherty et al. The WITAS unmanned aerial vehicle project. In *Proc. 14th European Conference on Artificial Intelligence, ECAI'2000*, pages 220–224. Wiley, 2000.
15. Patrick Doherty et al. A distributed architecture for intelligent unmanned aerial vehicle experimentation (to appear). 2003.
16. Joakim Gustafsson and Jonas Kvarnström. Elaboration tolerance through object-orientation. *Artificial Intelligence*, 2003.
17. Beth-Ann Hockey, Oliver Lemon, Ellen Campana, Laura Hiatt, Gregory Aist, Jim Hieronymus, Alexander Gruenstein, and John Dowding. Targeted help for spoken dialogue systems: intelligent feedback improves naive users' performance. In *Proceedings of European Association for Computational Linguistics (EACL 03)*, pages 147 – 154, 2003.
18. J. Kvarnström, P. Doherty, and P. Haslum. Extending TALplanner with concurrency and resources. In *Proceedings of the 14th European Conference on Artificial Intelligence*, pages 501–505, 2000.
19. Jonas Kvarnström. Applying domain analysis techniques for domain-dependent control in tal-planner. In *Proceedings of the Sixth International Conference on Artificial Intelligence Planning and Scheduling (AIPS-2002)*, pages 147 – 154, 2002.
20. Oliver Lemon, Anne Bracy, Alexander Gruenstein, and Stanley Peters. Information states in a multi-modal dialogue system for human-robot conversation. In Peter Kühnlein, Hans Reiser, and Henk Zeevat, editors, *5th Workshop on Formal Semantics and Pragmatics of Dialogue (Bi-Dialog 2001)*, pages 57 – 67, 2001.

21. Oliver Lemon, Alexander Gruenstein, Randolph Gullett, Alexis Battle, Laura Hiatt, and Stanley Peters. Generation of collaborative spoken dialogue contributions in dynamic task environments. In *AAAI Spring Symposium on Natural Language Generation in Spoken and Written Dialogue*, Technical Report SS-03-07, Menlo Park, CA, 2003. AAAI Press.
22. Oliver Lemon, Alexander Gruenstein, and Stanley Peters. Collaborative activities and multi-tasking in dialogue systems. *Traitement Automatique des Langues (TAL)*, 43(2):131 – 154, 2002. Special Issue on Dialogue.
23. S.K. Pal, L. Polkowski, and A. Skowron, editors. *Rough-Neuro Computing: Techniques for Computing with Words.* Springer–Verlag, Heidelberg, 2003.
24. Erik Sandewall. *Features and Fluents.* Oxford University Press, 1994.
25. Erik Sandewall. Cognitive robotics logic and its metatheory: Features and fluents revisited. *Electronic Transactions on Artificial Intelligence*, 2:307 – 329, 1998.
26. Erik Sandewall. Use of cognitive robotics logic in a double helix architecture for autonomous systems. In M. Beetz, J. Hertzberg, M. Ghallab, and M. Pollack, editors, *Plan-based Control of Robotic Agents.* Springer, 2002.
27. Erik Sandewall. What's in a decision. 2002. Submitted for publication.