17 17 12-03 03 Erik Sandewall Department of Computer and Information Science www.ida.liu.se/∼erisa/ www.ida.liu.se/

1 Laboratory, Demonstrator Project, Partners and Affiliates

The Laboratory. The Laboratory for Cognitive Autonomous Systems (CASL) is a research group within the Department of Computer and Information Sciences at Linköping University. Participation in the WITAS project is currently the major activity of CASL. The following are the present CASL members (November, 2003):

```
Erik Sandewall, professor, group leader
Malin Alzen, M.Sc., software engineer
Peter Andersson, M.Sc., graduate student
Karolina Eliasson, M.Sc., graduate student
Coynthia Hernandez, technician
Susanna Monemar, M.Sc., software engineer
Tobias Nurmiranta, M.Sc., graduate student
Heike 'Joe' Steinhauer, M.Sc., graduate student
Anna Maria Uhlin, administrator
```

Malin Alzén and Susanna Monemar are on short-term employments for the WITAS project. For percentage of time accounted to WITAS, please refer to the WITAS personnel report.

The DOSAR group in WITAS The contribution by CASL to the WITAS system and demo is mainly through the development of the DOSAR Dialogue System, an operator's-assistant system that enables an operator to interact with the WITAS UAV. Almost all the lab's members participate in the DOSAR effort. In addition we have the participation of the following partner groups and persons:

Skövde. The Distributed Real-Time Database Group at the University of Skövde cooperates through the following persons:

```
Sten F. Andler, professor, 10% WITAS
Jonas Mellin, lecturer, 10% WITAS
Marcus Brohede, graduate student, 80% WITAS
Henrik Grimm, technician, 50% IDA
Sanny Gustavsson, graduate student, 80% KK
Gunnar Mathiason, graduate student, 80% CUGS
```

The WITAS participation of the Skövde group is co-financed by other organizations. The percentages in the table specify the part of time for WITAS participation, followed by the source as follows: IDA = the C.Sc. department in Skövde, KK = the Knowledge and Competence Foundation, CUGS = the National Graduate School for Computer Science.

NLPLAB. M.Sc Genevieve Gorrell is a graduate student in the Natural Language Processing Laboratory (NLPLAB) in Linköping. She has participated as an advisor to Susanna Monemar (CASL) concerning the use of the Nuance speech recognition system.

The following cooperation is mentioned since Dr. Yi is an affiliate on the basis of common background and interests, and because cooperation in the coming years is being discussed.

Karlstad/FOI. Dr. Choong-ho Yi did his licentiate degree in Linköping before the start of WITAS. He pursued his Ph.D. studies while working at the University of Karlstad for several years, and joined FOI (Research Institute of National Defense, Stockholm) in 2001. He defended his Ph.D. in Linköping in 2002. (Additional details below).

In this memo, section 2 describes my long-term research agenda for cognitive autonomous systems and work done during 1997 through 2002 within WITAS according to that agenda. Section 3 describes the research by the graduate students that are now in CASL, and section 4 describes the research by the partners and partner groups. Section 5, finally, describes the work so far on the DOSAR system per se, mainly during 2003.

2 Research Agenda for Cognitive Autonomous Systems

I use *cognitive autonomous systems* (CAS) as the banner for research that addresses knowledge representation and especially cognitive robotics, but which in addition includes the architecture(s) in a systematic and integrated way, besides the logic(s) or other formal representation systems. The CAS approach attempts to find a concise characterization of the entire chain from sensing via decision to actuating while crossing that chain with considerations of concept-level memory and of interactions with the user.

2.1 Long-term workplan

The full treatment of the CAS research topic requires a combination of systems work and theoretical work; it also requires enough time (hours per week) during a sufficiently long period (years) due to the size and complexity of the topic. The stable funding and the organizational context of WITAS has offered me an opportunity to do exactly this, first as individual research and recently with students, but throughout as a long-term plan. That plan is divided into four overlapping stages, as follows:

1. Choice of a logic (syntax, semantics, and more) to be used as the formal base. This was achieved through my monograph 'Features and Fluents' that was published in 1994, together with some later extensions that have been collected under the name *Cognitive Robotics Logic*, CRL.
2. Characterization of important functions in cognitive autonomous systems, as well as important aspects of their architecture, using the CRL. The main period for this work has been from 1996 to 2002.
3. Choice of a software infrastructure that is appropriate as a platform for implemented, cognitive autonomous systems. The main period for this work started in 2001, and I consider the work to be essentially completed with the version of the Software Individuals Architecture (SIA) that was obtained in the summer of 2003, together with a library of practical modules for reasoning, communication, and presentation that has been built up during the last years.
4. Use of the above for an actual system, so that the concepts and designs could be put to test, and revised if necessary. The WITAS dialogue system, for speech-centered dialogue between the UAV and an operator on the ground, is now used for this purpose.

The agenda has always included the building of an actual system as one of the stages, but the choice of a dialogue system for this purpose happened fairly late. Previously I thought that a simple ground robot, of one of the common types, would be appropriate. Using the WITAS UAV system itself has not been considered, since it is too large, complex, and difficult to be addressed in the CAS framework in its present form. As WITAS evolved it became clear however that the dialogue task will fit quite well into the general agenda.

Year 2003 has been dominated by the beginning of phase 4 and the direct implementation work, resulting in the current DOSAR system that was part of the main WITAS result demonstration in October, 2003. This implementation is now (November/December 2003) undergoing a 'cleanup'. I expect that work during 2004 and onwards will be soundly balanced between formal work on one hand, and extensions to and experiments with the actual system on the other hand, which will also make the distinction between these overlapping stages more diffuse.

The present section will describe work in the first three stages of this overall plan. Its last stage, concerning the DOSAR system, will be described in sections 3 and 4, and in particular in section 5.

2.2 Cognitive Robotics Logic (CRL)

I propose to use Cognitive Robotics Logic (CRL) as a basis for those tasks of modelling, or knowledge representation that are needed in a robotic system such as in WITAS, or indeed in any kind of cognitive autonomous system. This logic is based on my research monograph 'Features and Fluents' from 1994, where I proposed an underlying semantics for logics of actions and change, and used it for assessing the range of applicability of a number of nonmonotonic logics (more precisely: nonmonotonic entailment conditions) for actions and change. The underlying semantics was based on a kind of simulation of worlds at hand, formally expressed, and on a notion of agenthood and actions in that 'simulated' world.

Continuous and discrete levels of actions. One intended topic is missing in that monograph, namely, an account of the relation between the continuous and the discrete levels of actions in the context of a robot. Work in this direction was started but not completed in time for the book, and therefore the simulated world in the book is defined entirely on the discrete level, using transitions in a finite space of states at timepoints usually represented as integers. Actions have extended duration in time, but everything is discrete.

When the book had been completed, and also during Phase I of WITAS, I returned to some of the questions that had not been addressed in the book, including this one. Relating the discrete and the continuous levels ought to be done using my previously (1989) published approach to embedding differential equations in logic. A fairly straight-forward application of the method from the 1989 papers made it possible to relate the continuous-level and the discrete-level descriptions of robot move actions within the logic. Please see section 4.3.1 of the accompanying publications-list report.

The term 'Cognitive Robotics Logic' (CRL) was introduced following these and a few other extensions of the logic, compared to what was defined in 'Features and Fluents'.

2.3 Functions and Architecture in CAS

Double Helix Archicture. The work on the two description levels for actions was incomplete in the sense that it did not model the machinery of invoking, executing, and finishing or interrupting actions. It was limited to modelling the performance within one action once it had been invoked, and it also had some other limitations. The first step towards a comprehensive model of the full action machinery was taken by the Double Helix Architecture, DHA, which is work that is still in progress. DHA is an abstract architecture (not an actual software system) for a robotic system that uses CRL formulas in its internal state and as messages between its main parts. It has some characteristics in common with standard three-level architectures, but it differs from them in the sense that it models the passage of time explicitly using a synchronous model (state update at discrete timepoints at fixed intervals).

The relationship between DHA and CRL is in fact twofold: DHA uses CRL formulas on its cognitive level for representing current state as well as messages but, at the same time, the semantics of actions in CRL is explicated using the DHA. In that sense the DHA is a further development of the set-theoretic structure that is used for the 'simulation' of actions in the Features and Fluents work.

It is intended that this characterization shall be useful both for defining a sufficiently rich semantics for actions and change (sufficiently rich for the needs of robotic applications, that is), and as a guideline for future implementations. The relationship between the DOSAR simulator and the DOSAR dialogue manager has taken some ideas from the DHA, but there is not a full correspondence. We intend to continue the work by contrasting an idealized concept, such as the DHA, with the grounding in reality that emerges from software development of nontrivial size.

Goal-directed behavior. One of the most significant functions in a cognitive agent is its goal-directed behavior. We follow a non-standard approach to characterizing goal-directedness which is simpler than the customary BDI approaches (BDI = Belief, Desire, Intention).

The idea is as follows: rather than introducing 'goal' as a modality, I consider goals as high-level actions. For example, the goal of getting to the airport is viewed as an action 'go to airport' that is to be broken down into a sequence of sub-actions corresponding to the plan towards the goal. As the plan is carried out, it may occur that some of its constituent actions fail, in which case the continued achievement of the goal will require to 'try again' or 'try something else'. The *actual* achievement of the goal may therefore consist of a concatenation of several initial fragments of plans up to the point of failure, followed by a completed plan that was successful. Any such concatenation of plan-fragments and plans is a rational way of achieving the goal, modulo the fact that actions can fail. The research question I addressed was how such rational goal-achieving sequences can be characterized through an extension of the logic used in the 'Features and Fluents' book.

Both of these topics (the characterization of actions, and of goal-directedness) required certain basic extensions of the logic, such as the notion of limited applicability of actions and the distinction between successful and failed performance of an action. Furthermore, the definition of composite actions (sequential or concurrent ones) that was present already in the 'Features and Fluents' book needed to be generalized in order to work correctly when both limited applicability and the success/fail distinction had been introduced.

The decision step. When it is in its planning stages or planning mode, a cognitive autonomous system can consider alternative courses of action. Once a course of action has been chosen, it goes into plan execution mode. The problems of planning and of plan execution have been studied extensively, not least by logic-based methods. The crucial *decision* step where the system commits to one of several possible alternatives, is certainly well studied from other points of view, but it is not clear how one can integrate those accounts into a logic-based 'model' for the entire cognitive autonomous system. An approach to this problem has been developed to the point of there now being a completed but unpublished manuscript.

The publications from the work described here are listed in section 4.3.3 of the publications-list report.

2.4 Software Infrastructure for CAS

Software Individuals. I am taking a radical approach to the question of the software infrastructure for a cognitive autonomous agent, using the label **software individuals**. This is a design for aggregates of data and programs that are capable of long life (months, years), capable of modelling their own structure, of self-modification and of knowledge acquisition from peers, and last but not least, capable of non-trivial reproduction where cognitive information is transmitted.

This work started in response to the question "if we are going to build actual artificial intelligences, and not merely build conventional software that has been doped with some AI components, then what requirements is that going to make on the underlying software techniques?" My answer was that it requires a combination of longevity and high flexibility: longevity in order to provide sufficient time for learning and other forms of knowledge acquisition, and flexibility as a necessary requirement for exactly those properties.

I propose in addition that the contemporary types of software artifacts: operating systems, databases, ordinary application programs as developed using mainstream programming languages, etc., do not necessarily represent the best and final fauna of software species. In particular, when addressing the question of appropriate AI infrastructure, I wanted to break out of those classical categories because they seemed to be too restrictive.

Software individuals are therefore a different kind of species. They are self-modifying systems, which in itself is anathema to the contemporary doctrines of programming languages. They integrate 'programs' and 'data' to a maximal extent. Indeed the abstract structure of a software individual is a labelled tree where each of the nodes may 'contain' a variety of data objects, each of which can be a collection of elementary data (a set of triples), a set of formulas, or an algorithm expressed in the host programming language, to mention a few of the options. The most significant, however, is the capability of a software individual to spawn descendants, where the latter inherit some of the structure of the parent, and then go off to acquire more structure using a knowledge acquisition script that came with them as they were spawned.

The formal definition of software individuals is quite concise. In addition there is a compact practical implementation where the abstract (static) structure is mapped on a combination of file directories and files in a few formats.

The software individuals architecture is therefore a kind of platform: the original or 'kernel' individual provides some basic services, and it is able to reproduce, but it does not contain applications. Indeed it is important that the kernel individual shall be as simple and concise as possible, but at the same time it must be capable of serving as a platform supporting various applications.

The development method has therefore been as follows. An initial design for the software individuals kernel was made several years ago, and its basic capabilities were verified. Then it has been used as a platform for a number of applications, including also in the DOSAR project. This usage gave feedback on the basic design, which has been revised several times. The next-to-latest design of the kernel is the one that is used in the current DOSAR dialogue system and DOSAR simulator. The latest design of the kernel is based on the experience from that usage, but actually moving the DOSAR systems over to the latest kernel has waited until after the main WITAS demo in October, 2003.

The plan is that a revised DOSAR dialogue system based on the latest SIA kernel shall be useful when we proceed from dialogue between one operator and one UAV, to the network situation with several operators and several UAV's. Each autonomous operator's assistant shall then be a software individual, reproduction is used when the network is extended, and knowledge transfer between individuals becomes a standard operation. This is for year 2004.

Web resource objects. In network-centered applications it will not be sufficient to present information to the operator using speech and pictures; there will also be a need to present structured information in the form of tables with or without animation. We have developed a facility, the Web Resource Definition Language (WRDL) that provides a way of defining such information and its associated layouts using a 'functional' style. Although first developed for purposes outside WITAS, it seems likely that this technique will be of use in the continued development of the dialogue system as well.

2.5 On the relation between theory and system

As described above, our agenda for Cognitive Autonomous Systems integrates theory (formal treatment) and work with actual software systems. In principle I view the latter aspect as the primary one. It would be a mistake (and it is not an uncommon mistake, I think) to assume that the formal treatment must come first, that a 'deeper' account (more difficult to produce and to understand) is necessarily more relevant and powerful, and that the job of an implementation is to verify what has already been designed. To my mind it is better to let the realities and the intuitions from experimental implementations be the guide, and to view the first stage of formal treatment as a way of abstracting from the actual system in the interest of structure and generality. One may need to iterate this process in order to arrive at a duo consisting of a formally expressed model for the system or domain at hand and a viable implementation that is consistent with the model (in the engineering sense of 'model'). Only then is it meaningful to proceed with technically more sophisticated theoretical work that uses the chosen model.

2.6 Completion and publication

The following is the completion and publication status and the scenario for forthcoming publication, for the stages of the CAS research agenda that were defined above:

1. Choice of a logic: Cognitive Robotics Logic, CRL. Stable publications exist, in particular in the form of the reference article published in the ETAI.
2. Characterization of important functions in cognitive autonomous systems, as well as important aspects of their architecture, using the CRL. These can be described and published one by one, which has also been done, but the strong validation of this work is only obtained when the same functions have been implemented in the framework of an actual system. The present plan is to demonstrate this within the DOSAR platform, using both the simulator and the dialogue system, in such a way that finished articles can be submitted during 2004.
3. Choice of a software infrastructure. The first report of the Software Individuals Architecture was published in 2001, and a second one in 2003. However these are reports along the way. A conclusive report of SIA and its full use is not likely until 2005, although additional partial reports should be possible.
4. Use of all the above in the actual DOSAR system. I aim for working on a monograph during 2005, resulting in publication during 2006. That monograph should be accompanied by one or more summary articles at the same time.

The publications in these categories are listed in the separate memo "Dialogue Technologies and Cognitive Autonomous Systems. Overview and Publications".

2.7 General issues of methodology and publication

Looking back at the publication situation for our agenda, I am not entirely satisfied: although a fair numbeer of results have been published, we still have uncomfortably much in the pipeline, or only in conference papers. One possible analysis is that I have taken on too broad a problem and spread too thin as a consequence. Without dismissing that analysis, I think it is also worthwhile to take the opposite perspective: if one can presuppose that our topic is worth addressing and the agenda is reasonable from the point of view of the investigation per se, then to what extent is it supported or discouraged by the generally accepted methodology and the standard publication procedures in our field?

My interest in this question is a reflection of my engagement in scientific journal activities, in particular the Electronic Transactions on Artificial Intelligence (ETAI)[1] which I started in 1997, and being one of the two Editors-in-Chief of the Artificial Intelligence Journal (AIJ) since 2001. The questions of research methodology and publication policies for systems-type research are of considerable interest for both of these journals, and the various parts of the WITAS project have provided a lot of illustrations to those general problems.

The contemporary system of research publication and evaluation favors producing a steady stream of articles that add new results to an established approach without upsetting it too much. The CAS agenda aims at a comprehensive account which by necessity will be fairly large, and which must be motivated not only by its formal properties but also by its conceptual coherence and by reference to an implementation. Therefore the natural presentation for it will be as a monograph. This is further accentuated by the choice of fairly non-standard approaches in several parts of our work in this area.

Publication of successive papers on specific parts of the approach and the results therein is also necessary of course, for a number of good reasons. Then a second albeit smaller problem arises because of the proposed view about the relation between actual systems and formal treatment that I defended above. The problem is that the first generation of results tends not to appeal a hundred percent to either the theoretically minded or the implementation minded reviewer. The results are naturally seen as 'preliminary' on both sides, even if they are concrete and well motivated results from work during an extensive period of time.

It is a matter of interpretation whether this problem is due to the character of the problem, requiring long delays between start of work and arriving at acceptable publications, or whether it indicates a fault in the contemporary research methodology in our field.

[1] Incidentally, the WITAS project has given some support to the running of the ETAI during these years.

3 Graduate student research in CASL

I proceed now to brief descriptions of the Ph.D. and licentiate topic areas of the CASL graduate students, mentioned here in the order that they have joined the group. Notice, however, that a large part of the time for Tobias, Karolina, and Peter during 2003 has been devoted to their graduate-school courses and to the implementation effort for DOSAR. Their work on the topics described in this section is therefore in its beginning stages.

3.1 Heike (Joe) Steinhauer

Joe Steinhauer is a graduate student in CASL but she has not participated actively in WITAS. Her research interest is in the modelling of the behavior of automobiles, both alone and in aggregates such as queues and 'convoys'. Queues occur before traffic lights, for example, and 'convoys' are sequences of cars that drive after each other at limited distance on a highway. This domain is chosen as a good example of common-sense spatio-temporal knowledge representation and reasoning.

It is easy to see how results in this area could be used in a future extension of the WITAS system, provided however that the vision system or an alternative sensing system can provide the required observations.

3.2 Tobias Nurmiranta

Tobias' interest is in the design of persistent software systems, that is, software systems that will be operational instantly when the computer is turned on. This requires a new approach to the relation between, and use of volatile and non-volatile memory. He is implementing a system on the 'systems software' level that borders over the traditional roles of operating system and programming language, and where paging and garbage collection in the new structures is one of the key issues. At this point he has come quite far with the design and implementation, and he has started writing on what will be the first article about his work.

Not too far down the road, I hope there will be a way of relating Tobias' work to the software individuals work described above, in particular since the present implementation of the formal software individuals design should really be seen as a primitive way of emulating the individuals using directory structures. Each individual is implemented at present as a directory with all its subdirectories, recursively. Tobias's work has the persistence goal in common with the software individuals, and promises to provide a better foundation for them than their present implementation which is based on a heavy use of conventional directory structures. However, this topic will only be considered after Tobias has completed his licentiate degree.

Beside his thesis work, Tobias has also helped with a number of support efforts for the WITAS dialogue systems and DOSAR, in particular the NGUI (new graphic user interface), the video capture system, and interfacing the WITAS-Stanford dialogue system to the W2K middleware.

3.3 Karolina Eliasson

Karolina's thesis topic concerns long-term memory in a cognitive autonomous system. Given that an autonomous operator's assistant operates during a long period of time (months, at least) in an actual field mission, the question arises by what criteria it is going to 'forget' actions that it has performed and information that it has received from its operator and also from other agents in the network. Such forgetting must of course be done with respect to the conceptual frame of reference of the agent and of the application.

Furthermore, 'forgetting' type reduction and restructuring of an agent's long-term memory has many ramifications for the dialogue system. One obvious case is when the system has 'forgotten' a certain piece of information, but it still 'knows' that it 'knew' about it before. This type of situation needs to be represented in dialogue. Another example is when recall is possible but only with a time-delay, and the agent has to keep the conversation alive in a natural manner while it searches its memory.

Karolina has made big contributions to the present DOSAR systems, in particular for the long-term memory facility using a MySQL relational database. The present facility is integrated into the dialogue, but there is no 'forgetting' yet. Another contribution was for a capability estimator whereby the DOSAR system is able to answer questions whether the helicopter is able to perform a particular action, such as flying to a particular point, within certain restrictions with respect to flying time, fuel consumption, or other. This implementation work provides at the same time a beginning for her thesis research.

3.4 Peter Andersson

Peter's research interest is in the network and communication structure in a network of autonomous agents, with applications for the type of network that is envisioned in the continued CASL and DOSAR research, that is, a network where the nodes are autonomous operator's assistants, autonomous vehicles such as UAV's, and autonomous sensor stations. It is assumed that geographical information, referring to the actual physical position of each network node including operator nodes, is an important aspect of the information being handled.

Particular emphasis is placed on the question how the characteristics of the network and its member nodes (both static and dynamic characteristics) influence the dialogue between operator and operator's assistant.

In the context of DOSAR, Peter has implemented the D2W middleware that mediates between the WITAS-Stanford dialogue system, the DOSAR dialogue system, the DOSAR simulator, and the ground-station facilities of the WITAS autonomous UAV.

4 Research by Partners and Affiliates

This section will briefly describe the research in partner and affiliate groups outside Linköping.

4.1 Skövde

4.1.1 Research Profile

The **DeeDS Executive** is the focus of interest of the Distributed Real-Time Database Systems Group at the University of Skövde. DeeDS is a distributed active real-time database, whose main motivation is to avoid sources of unpredictability, in particular disk accesses, network accesses, and distributed transactions.

Traditionally, databases store their data on disk to ensure durability. However, since disk accesses are one order of magnitude slower than main memory, this is not acceptable for real-time systems with stringent data access times. To avoid this unpredictability, all data in DeeDS are stored in main memory.

Another source of unpredictability is due to the distribution. Accessing data over the network, just as with local disk access, is an order of magnitude slower then fetching from main memory, and in the case of a partitioned network it is impossible. DeeDS therefore provides virtual full replication. Virtual full replication means that all data needed by applications are present locally, but unused data can be discarded.

Distributed databases are faced with unpredictable commit protocols, for example, distributed commit, or 3 phase commit (3PC). DeeDS addresses this problem by allowing local commits and by handling any inconsistencies via its replication protocol that guarantees eventual consistency.

Many real-time systems have reactive behaviour that can be modeled by the active functionality found in DeeDS as event-condition-action (ECA) rules. An ECA rule reacts on the occurence of an event, and if the corresponding condition evaluates to true, it performs a specified action.

In summary, DeeDS is a distributed database that has all its data in main memory. Data are (virtually) fully distributed to avoid remote access and enable local commits. In addition, an important feature is the active functionality provided by the ECA rules, which provides a convenient way of representing the reactive behavior of a real-time system in the database. Distributed real-time simulation is one kind of application that benefits from using a distributed active real-time database such as DeeDS, especially if there are hardware-in-the-loop situations.

4.1.2 Common Interests

The WITAS cooperation with the DeeDS group in Skövde is motivated from the point of view of going to network scenarios in the coming years. When looking at different, alternative ways of providing communication between agents and shared databases for them, a distributed database such as DeeDS is one possible approach.

In order to bring some experience with this technology into WITAS, the DeeDS group has implemented a pilot application, namely a visualizer for the DOSAR simulator that is used when developing the DOSAR dialogue system. The visualizer allows one to see the road structure, buildings, and vehicle movements (cars and helicopter) in the simulated world at the same time as the dialogue proceeds. DeeDS is used as the communication medium between the DOSAR simulator and the visualizer.

4.2 Choong-ho Yi (Karlstad/FOI)

4.2.1 Research profile

The licentiate research of Choong-ho Yi (before WITAS) concerned assessment results for logics of actions and change in the presence of concurrent actions. He then moved to the University of Karlstad, and became interested in the use of logics of actions and change for characterizing business processes and, in particular, the relation between general business goals (such as maximizing profit, or satisfying customers) and concrete choice of actions. Concurrently with work in this applied area, he also developed extensions to a logic of actions and change (a descendant of 'Features and Fluents') that captures the expressiveness of the Universal Modelling Language, UML. This work was done with Erik Sandewall as the advisor, and resulted in a Ph.D. that was defended in Linköping in 2002. Meanwhile he had moved from Karlstad to the Swedish Defence Research Agency, FOI, in Stockholm.

4.2.2 Common Interests

Choong-ho's Ph.D. research is akin to the work reported under 'Cognitive Robotics Logic' (subsection 2.2). The forthcoming extension of the WITAS dialogue-system work to network scenarios matches well with Choong-ho's interests in his present employment, and increased cooperation is being discussed based on our common interests in earlier work.

5 The DOSAR Dialogue-System Project

5.1 Project goals

The DOSAR[2] project addresses the problem of high-level operator dialogue with one or more UAV's, in particular as part of the WITAS project. It is planned for a period of at least three years, 2003-2005. The system goal of DOSAR within that period is to develop a system with the following characteristics:

- Communication about the physical actions of the robots, including both on-going actions, past actions, and intended future actions.
- Communication using spatial concepts, referring to the location of the UAV itself and to the objects that it observes on the ground.
- Multi-initiative dialogue resulting from the fact that both the operator and the UAV itself may initiate dialogue threads.
- Close integration with the behavior and the dynamic parameters of the UAV and its on-board system, in particular, the video camera.
- Network awareness: dialogue shall occur in a network containing numerous dialogue agents and numerous UAV:s and other robotic systems.
- Longevity: each dialogue agent shall be capable of operating during a long time, at least for months, and shall maintain a memory of events during that period.

Research goals of the DOSAR dialogue project include the study and development of software architecture principles for robotic dialogue agents, and the study of knowledge representations for such agents including logics of spatiotemporal phenomena, action, and change.

5.2 Target system

We aim at an overall system design along the following lines. There is a *network* consisting of a potentially large number of *agents*, including in particular *robots*, *dialogue agents*, and *information repositories*. The term 'robot' is used to include unmanned vehicles of several kinds as well as stationary sensory equipment with local capability to interpret data. The 'dialogue agents' communicate with human operators using a 'presentation system'; each dialogue agent serves exactly one operator. The term 'information repositories' includes geographical information systems for use by the robots and dialogue agents, for example.

We assume furthermore that there is an agreed *inter-agent language* within the network that is adequate for the needs of the dialogue agents, and that the robots are *dialogue enabled* in the sense that they can communicate in terms of the inter-agent language. In this way the robots can receive instructions for actions, report

[2] DOSAR was initially an acronym for 'Dialogue-Oriented Simulation And Reasoning', but more recently for 'Dialogue-enabled Operator Support Agent for Robots'.

on the progress of those requests, and obtain information that they need for their tasks. We foresee several levels of dialogue enablement in the robots.

Normally, each user works with her or his dialogue agent, and each user agent serves one particular user. The *external dialogues* in the network are those between users and their respective dialogue agents, whereas the *internal dialogues* are those where dialogue agents and robots are the only participants. From the user's point of view it is the external dialogues that are of direct interest, but dialogue agents must be able to participate in internal dialogue in order to carry out various tasks. In the simplest case one dialogue agent communicates with one robot, but the generalization is also needed.

The primary topics of dialogue shall be (a) the physical actions of the robots, (b) objects and events in the world that the robots observe, (c) objects and events in the world that the users observe or otherwise know about. This includes commands by users to robots, reports from robots to users, and mediated communication from user to user. Conceptually it also includes, for example, users giving background information that is relevant for robots in carrying out their tasks, users obtaining information about events and conditions in the past, and multi-party dialogue for the purpose of planning and resource allocation.

We aim for external dialogues that are conducted in spoken natural language and accompanied by visual interaction (pictures, video clips, diagrams) for both input to, and output from the dialogue agent. Typed natural language, using e.g. a keyboard, may be used as a fallback option, for example if the environment is too noisy for speech.

The information that a dialogue agent has, or has access to, is not only useful for answering direct questions about that particular piece of information. It is also useful linguistically, for example for identifying the object that is referred to in a descriptive phrase, or for identifying a timepoint that is characterized by reference to a past event, and for which additional information is sought.

5.3 Target system requirements

Compared to other research on AI-based dialogue systems, our particular application imposes strong requirements with respect to real time and concurrency. Multiple initiative situations are abundant, in particular because the robot (the UAV) may initiate conversation threads, but also because the operator may request actions that take non-trivial time, and interactions may be initiated by both the operator and the robot in the course of those actions. From the scientific point of view, this means that we have here a nice application for studying some of the major topics of interest in dialogue system research.

Some other topics of current interest are less important in our case. In particular, we can make the assumption "what you say is what you mean", at least as long as we stay with direct command situations. If the operator asks the helicopter "can you fly to the high school in four minutes", then she probably means exactly that, and it is not likely that this is a request to fly to the high school. At present we therefore do not address the topic of inferring the operator's real intentions.

The ontology of the application domain is also a topic of considerable interest in the dialogue community at present. In the case of the WITAS system, the ontology is quite restricted: we have to deal with roads, buildings, road vehicles, and only a few other types of objects. The limitation is primarily set by the vision system. DOSAR uses a simple two-level type hierarchy, where there may be a few different subtypes of ground vehicles (cars, schoolbuses, firetrucks, ambulances, etc) and similarly a few, mutually disjoint subtypes of buildings. This is anyway more than what other parts of the WITAS system can handle at present, but compared to ontologies in general it is not very challenging.

One important problem that we do address is how different elements in the dialogue relate to different points in the overall architecture for the operator's assistant. Later on this must be expanded to understanding how they shall relate to the architecture of the robot and the structure and contents of the overall network. This appears to be a key issue in the engineering of practical robotic dialogue systems.

5.4 Design choices

A number of design decisions were made at the beginning of this project, and during its first eight months we have started to implement them with two time horizons in mind: participation in the major WITAS demo during month 10 of this project, and obtaining the best results from an overall project duration of three years. The following are some of the basic decisions that have guided our design at the outset.

1. Dialogue agents shall be capable of representing and performing 'actions' in general, and not be seen only as conversation agents. Their notion of actions shall include both actions by themselves, actions by the robots, actions by other dialogue agents, and actions by the users. It shall cover both physical actions, such as movements, communication actions, and deliberation actions such as planning. - This choice is motivated by conceptual economy and as a way of obtaining maximal generality from a limited number of primitive concepts.

2. Dialogue agents shall be based on a software platform that is also adequate as a basis for some simple robotic agents, and for simple robotic agent simulators. Up to a certain point these shall be able to share the same software; above that point they have different additional facilities. It is *not* however intended or expected that this platform be adequate for all kinds of robotic agents, and in particular it is by no means supposed to be adequate for an autonomous UAV with its very high complexity and its strong real-time software engineering requirements. - This choice is a fairly natural consequence of point 1.

3. The action concept mentioned in item 1 shall be implemented already in the common software platform. It shall be a clean implementation of the concept of actions in a logic of actions and change, and specifically the Cognitive Robotics Logic (CRL). - This choice is made in order to have a clear correspondence between theory and systems, with the resulting opportunity to put theoretical results into practical use, but also to facilitate feedback from the implementation experience to the theory.

4. The common agent platform shall be time-aware, that is, it shall represent time and duration on all levels from the sub-second level to the level of months and years. In particular, all actions shall be characterized as starting and ending at well-defined timepoints, and time is modeled as metric and discrete with a fixed frequency. (At present the base frequency is 2.5 Hz so each timestep is 0.4 seconds). - This is motivated both by the strict real-time character of the helicopter control system with which the dialogue system shall interface, and by the structure of several major logics of actions and change.

5. Concurrent activities are required both in our UAV simulator and in our dialogue system. The following approach, which is standard in many simulation systems and real-time systems, is used for obtaining concurrency in our agent platform. The computation is organized as a periodic process with the base frequency mentioned above. In every such cycle the system goes through a list of *tasks* and executes the cycle procedure belonging to the task, provided that the task is *enabled*. Each task is associated with an enabling procedure, a cycle procedure, a disabling procedure, and an error handling procedure. Tasks may be enabled and disabled dynamically, and the cycle procedure is only applied for enabled tasks. - This choice is well adapted to the needs of a simulator and for the lower levels of robot software, but it is also quite appropriate for elementary dialogue behavior, in particular if the real-time aspect is important in the dialogue.

6. In addition to what was said in point 5, there will also be a facility for asynchronous concurrency with processes that can not easily be reduced to short timestep computations, such as complex deliberations.

7. Software exchange between agents is important, where 'software' means both programs and data. A dialogue agent that has acquired a particular behavior shall be able to communicate it to other user agents. A dialogue agent shall also be able to 'teach' a robot how to execute a particular type of request. - This choice is motivated both by software engineering considerations, in a broad sense of software engineering, and by artificial intelligence considerations. We think that the ability to communicate knowledge to peers and to acquire knowledge from peers ought to be an important feature of intelligent systems in general.

Several of these choices were supported by our earlier work within WITAS. The synchronous update and round-robin execution of tasks was first set up for the DOSAR simulator, and later on we found that it was a promising choice for the dialogue agent as well. The software exchange goal was adopted based on the experience with the software individuals architecture.

5.5 The present system: major parts

Our first-generation system has been designed with two goals in mind: to have one working, albeit limited system by October, 2003, and to lay the ground for the continued work. We have been strongly influenced by what we learned from the WITAS-Stanford dialogue system, and in fact we have chosen the performance of that system as a bar that we should match, as far as possible, even in our first system. We did not however choose to proceed with the actual software from Stanford, and instead we have started over with our own design. The seven design choices that have just been described were the major reason for starting over, since we thought it would be difficult to implement them on the basis of the WITAS-Stanford system. We also felt that the reimplementation effort would be of manageable magnitude, given the size of the group and the software base we already had, and that it would be a learning experience in itself. The results so far seem to confirm those assumptions.

The present system consists of the following communicating parts:

1. The DOSAR dialogue agent, further described below, that is capable of dialogue with the user.
2. The DOSAR simulator.
3. A MySQL database that is used by the dialogue agent for medium-term and long-term storage of system history, including observations and actions.
4. The D2W switch that can be used to connect the DOSAR user agent with the DOSAR simulator and with the DOR (Dynamic Object Repository) of the WITAS UAV system. The simulator and the DOR look exactly alike from the point of view of the user agent, communication can be directed to one or the other.
5. The specialized, real-time GIS (geographical information system) of the WITAS UAV system. It is also connected to the D2W switch.
6. The Nuance speech interpretation system, using our own grammar and Java-coded wrapper and interface. This system sends text versions of input sentences to the dialogue agent.
7. The Festival speech generation system. The dialogue agent sends text versions of sentences to this system to be pronounced.
8. The NGUI (new GUI) graphical user interface, with capabilities for displaying video clips as requested by the dialogue agent.
9. A videoclip server that receives a composite video stream and stores it in MPEG format as clips of moderate size, for example 20 seconds per video clip, while also keeping track of the clock time for each clip. This server is used for recording the video stream that is sent from the helicopter to the ground station during flights. The dialogue agent can request playback of one or more continuous video clips that were recorded at a given time, in response to user requests, so that those clips are displayed on the NGUI screen.

Items 1, 2, 3, 4, 8, and 9, as well as the grammar for item 6 have been developed in our group. Items 1 and 2 are written in Common Lisp, the others in Java, C or C++. The system structure and interconnections are shown in figure 1, as follows. The DOSAR dialogue agent communicates with the D2W switch and the NGUI using a KQML-like message format, with the Nuance system using a specialized format, and with Festival using simple DOS-level system calls. It communicates with the 'memory' database using the software interface between Allegro Common Lisp and MySQL. The D2W switch provides translation between the DOSAR, KQML-like messages and the CORBA-based interfaces to the DOR and the specialized GIS, whereas no translation is needed for the link to the DOSAR simulator. The same message format also allows the dialogue agent to direct the NGUI to obtain and display particular video clips from the videoclip server.

5.6 The present ontology

The world that is used in the DOSAR system at present, including both the simulator and the dialogue system, contains the following types of objects on the ground. Buildings and roads have fixed positions and are known to both the UAV GIS system and the dialogue system from the beginning of the dialogue. For cars, the position varies as a function of time. The simulator knows the exact position of each car, of course, but the dialogue system only gets to know about the position of a car when it comes within the field of vision of the camera in the (simulated) helicopter. (Car positions are not obtained when the dialogue is run against actual UAV flights). Each car has an identifier, and this identifier is used by both the simulator and the dialogue manager, so they can communicate in terms of unique names for cars.

Containers are like buildings, except that they are not known initially by the dialogue system; their existence and location becomes known to the dialogue system by either of two methods: the helicopter 'sees' the container, or the operator states that there is a container in a certain position. Usually the position of a container is expressed relative to the position of another object, for example "in front of the hospital". Bodies (intended as bodies of injured or dead persons, in a disaster scenario) are similar to containers, except that they can also change position within the duration of a scenario.

When containers and bodies are introduced by the operator using speech input, then the dialogue introduces its own, local identifier for the object in question. The local identifier shall of course be identified with the 'objective' identifier used by the simulator when an object of these kinds is observed and reported by the helicopter. The system attempts to make this identification using an optimistic spatial adjacency criterium. This is a first instance of a general and important representation problem.

In summary, there are the following main types for ground phenomena: roads, buildings, vehicles, containers, and bodies. Buildings and vehicles can have one level of subtypes, for example 'hospital' or 'school bus', and they can have a color property. Buildings can also have a property for the facade material (brick, plate, etc), and they can have a few attached objects such as the roof and the front door, which in turn can have color and material properties. The same applies for containers. This is a quite limited ontology, of course, but it is enough for the dialogue and more than enough for the actual WITAS UAV.

The UAV is modelled as consisting of the vehicle per se and the video camera. The UAV has the elementary parameters for position, orientation, and movement, and the camera has parameters for orientation relative to the helicopter, and for zoom. The simulator allows the dialogue system to control the movement of the helicopter as well as the camera parameters. In addition it can have a few other payloads that are entirely passive, as discussed in subsection 5.8.

Only a subset of this ontology is available when the DOSAR system communicates with the WITAS helicopter. In this case the dialogue system can inquire about the helicopter parameters, but not influence them, and it can control the camera parameters or the zoom. Observations of phenomena on the ground are not provided in symbolic form from the UAV to the dialogue system.

5.7 The present dialogue agent

The DOSAR system that was used in the demonstration at Revinge in October, 2003 was a first version which has capabilities in the following categories. Naturally it does not have complete language or cognitive capability in any of them, but the list does indicate what may be expected after extending the system with additional cases. Throughout, the term 'the UAV' refers to the DOR and/or the DOSAR simulator:

First-level features i.e. basic interactions

- Receive user requests for UAV actions and send those requests to the UAV for execution
- Monitor the execution of requested actions on the basis of progress reports from the UAV, and report back to the user
- Receive user queries for UAV status, query the UAV if necessary, and report to the user
- Receive observation reports from the UAV and report back to the user, with a capability for aggregating several elementary observations into less detailed summaries
- Honor requests to display video clips pertaining to a particular time or a particular past event that was observed by the UAV. (The event recognition is not part of our system)

Second-level features

- Identify the referents of some descriptions of objects, use them in the interpretation of sentences, and query the user if a description is ambiguous or not understood
- Manage the situation when several things need to be said at the same time by the dialogue agent
- Honor policy requests with respect to the interaction, for example, to report or not to report certain types of observed events
- Answer queries about past actions and events
- Receive and honor requests for the UAV to execute a plan in the sense of a sequence of more elementary actions
- Honor questions whether the UAV is able to perform a certain action within given restrictions on e.g. duration or fuel consumption

This dialogue agent is based on the agent platform that is also used by the DOSAR simulator, which means that time-awareness and a notion of actions are present already within the platform. In each platform cycle it may receive stimuli as speech input from the user, as messages from the UAV, or as progress reports from actions that are internal to the dialogue agent and that may last for several timesteps.

Spoken input sentences go through a conventional sequence of speech to text transformation (in Nuance), parsing, semantic transformations including identification of referents, and response. Such a sequence is considered as an action within the dialogue agent, and is recorded as such. The action may be completed, with success or with failure, or it may be incomplete e.g. if a referent can not be identified successfully. In the latter case, the continued interaction for resolving the problem is considered as additional actions, not part of the present one. The speech-to-text transformation may take several timesteps for lengthy sentences, but the subsequent steps for parsing etc can usually all be performed within one timestep.

The final step in the processing of an input sentence is to offer the internalized sentence to a set of message 'consumers' each of which specifies how 'eager' it is to process the sentence; the consumer with the highest degree of eagerness gets the call. The resulting decision may be to send a message to the UAV, or to register an output 'protosentence' (skeleton for a sentence) as an intended output phrase.

Messages from the UAV to the dialogue agent may be progress reports for actions, reports on observed events, or answers to queries sent to it by the dialogue agent. Such messages are also offered to message 'consumers', like for sentences that were input from the user. The use of such consumers provides a local and specialized rule-base facility.

Since the dialogue agent is based on the robotic agent platform, it is natural for it to maintain information about the current state of the UAV. The CRL logic and its implementation allow separate representation of the actual UAV state and the dialogue agent's belief about that state, but that possibility is not in use for the dialogue at present.

The dialogue with the dialogue agent is characterized by multiple initiatives, and therefore by what is often called concurrent 'threads' in the dialogue. Our system does not assign sentences to such threads. Instead, the short-term memory structure contains both the communication acts themselves (input and output phrases) and a higher-level structure that represents relations between actions, such as 'communication act B is an answer to communication act A'. Formally, these relations are realized within a first order logic, but using objects each of which is effectively the reification of a communication act. The reportoire of such metalevel relations is extensible, and is not restricted to binary relations. This opens the way for representing more complex situations, for example, when one phrase serves to merge several 'threads', or to close one 'thread' while at the same time proceeding on another one.

5.8 Current activity: planning for the UAV patrol

The DOSAR system that was used for the October demo was equipped with a number of 'handles', that is, points in the architecture where more advanced handling of messages could be plugged in. It is part of our strategy to first design a base system that is as simple as possible while still providing the functionality for simple dialogues, and then to extend it using different plug-ins into those handles.

The current (November, 2003) activity with respect to plug-ins is concentrated on the definition, use, and revision of plans in the operator's assistant system. For the concrete dialogue scenarios we use the *UAV Patrol World* which is defined as follows. Consider a helicopter UAV that resides in a small town where police and other emergency services are scarce, and where the UAV is used to substitute for them to the extent possible. It has its base on the roof of the tallest building in the town, and can come into action when called, for example by the robbery alarm in a local store. The following are the services that it provides:

- Fly to the site of a robbery and obtain pictures of cars leaving the site.
- Fly to the site of a road accident and obtain aerial photographs of the site, for use in the forthcoming investigation of the accident by police and by insurance companies. Also, identify if some of the cars there is on fire.
- Deliver a fire extinguisher to the site of a road accident if needed because some of the cars involved there is on fire and no one on site has an extinguisher.
- Fly to a building on fire in order to obtain pictures that are transmitted to fire squads.
- Look for a stolen car (it happens from time to time that someone "borrows" a car, drives it over some distance, and then just leaves it).
- Look for cars that are parked in violation of current regulations, in particular along major roads. If found, obtain a photograph where the license plate of the car is visible, except if it is an emergency vehicle (police car, ambulance, or vehicle belonging to the fire squad)
- Locate over the road outside the town's elementary school around the time of school breaks, and check that cars passing there respect the speed limit.
- When the temperature is just below freezing and air humidity is high, fly over the highway passing the town and check whether the road is icy. (This can be determined from the air using a specialized instrument on board).

To meet these needs the UAV has a few alternative payloads, namely:

1. A camera for still images and video, controllable by the UAV
2. A fire extinguisher mounted in a parachute; the UAV is able to drop it at a place where it is needed
3. An instrument for determining ice cover on a road
4. An IR instrument capable of identifying fires

5. A laser-based instrument for determining the velocities of passing cars

The UAV can carry either the camera plus one of the three instruments, or just the fire extinguisher. It can change payload by going back to its base; there is a machinery there that will mount and dismount these payloads on the request of the helicopter system. The number of available fire extinguishers is limited, and is known to the operator's assistant system.

We wish to provide dialogue support for scenarios of the following type:

- The operator defines a mission for the helicopter, consisting of a few tasks that need to be performed. These tasks are not necessarily specified in the optimal order of execution, and auxiliary actions that are needed in order to be able to perform the tasks need not be mentioned explicitly.
- The operator's autonomous assistant schedules the order of performing the requested tasks. While doing so it may also need to add enabling tasks, such as the task of returning to base in order to mount and unmount payload.
- The operator assistant system instructs the UAV to perform the actions one by one according to the plan. For each action that the UAV reports has been completed, the assistant informs the operator.
- While the mission is being performed, additional information may arrive to the operator's assistant that causes it to revise the current plan. New problems may require attention, and there may be reports to the effect that existing problems have gone away 'by themselves'. For example, a fire after a car accident may have been put out by the people on the site, or it may be reported that an apparent accident actually was not a problem.

An example of a dialogue scenario in the UAV Patrol World is found in the Annex (subsection 6.3). At this point (end of November) we have implemented the basic machinery for defining, executing, and revising plans, and this machinery has also been connected to the dialogue manager using the handles that were mentioned at the beginning of this subsection. Additional work is needed, but with this domain and this type of dialogue scenarios we can identify and address a number of interesting problems at the intersection between dialogue, planning, and cognition in general.

5.9 Conclusions and future plans

In summary, our system design provides an effective machinery for performing simple dialogues, and at the same time it contains 'handles' that open the way for dealing with more complex dialogue situations. Our system does not give any particular attention to implicit requests and implicit commitments in the communication; it basically assumes that 'what you say is what you mean'. Addressing implicit requests and commitments is on the long list of future topics, of course, but it does not seem to be among the most pressing needs for the case of robotic dialogue.

One major purpose of the current DOSAR system is of course to serve as the platform for the thesis work of several of the graduate students in CASL, in particular for Karolina Eliasson and Peter Andersson, but in fact also for Joe Steinhauer.

Continuations of the current activities may also be obtained through M.Sc. projects, guest researchers, and in other ways. I consider the following problems to be important in the sequel:

- Better handling of failed communication acts in the sense just described
- Dialogue situations with multiple dialogue agents and multiple robots
- Use of spatial and spatiotemporal concepts in the dialogue

Increased use of Cognitive Robotics Logic in the system is one of the means that I intend to use for addressing these problems.

In addition, we plan to further pursue the work on the two abstract architectures - the Software Individuals Architecture and the Double Helix Architecture - by relating them to the presently implemented system as well as to its continued development.

6 Annex: Transcripts of Dialogues

This annex first contains transcripts of the two dialogues that actually occurred at the demonstration in Revinge in October, obtained from tape recordings made at the time. (The tape recordings are also available as `.wav` files). There were two dialogues in succession, one that was run against the simulator, and then one that was run against the WITAS UAV during actual flight. They are exchangable since the D2W middleware provides a transparent, equalizing interface. However, the actual-flight dialogue is restricted since the UAV was set to only accept camera movement commands from the dialogue. The fly-to commands were blocked for security reasons. Therefore, fly-to commands occur only in the dialogue against the simulator.

From the overall architecture point of view, camera movement commands and helicopter control commands are handled by the same type of messages and along the same message paths, until they reach their respective task procedures in the UAV. Because of this, and because the helicopter-command dialogues have been verified in simulation mode, no additional work is needed on the dialogue-system side for enabling the operator to direct the helicopter flight by voice command.

After this, there is a third dialogue that was run on November 30 and that illustrates some of the new developments during November, in particular the planning and plan execution facility. This example is offered as an illustration of the state of the work at the time of submitting this report, but not as an example of completed work. The dialogue about changes of the plan needs to be further improved, and the connection between the actual system and the (also ongoing) formal work on goal-directed behavior and on decisions (see section 2.3) needs to be explained as well.

6.1 Revinge-0310-#01: Dialogue with DOSAR simulator

Comment: it can be seen from this dialogue that the correct sequencing of output phrases in the presence of concurrent dialogue threads, does not work entirely correctly. This problem did not use to be there before and is probably due to a last-minute bug before the demo.

```
>> take off
<< the helicopter is about to start taking off
<< it is in the midst of taking off
<< it has completed taking off

>> fly to the church
<< the helicopter is about to start flying to the church

>> tell me about the buildings
<< ok
```

The helicopter tells us about his observations when he sees a building. At the moment no building is in sight, but there will be later in the dialogue.

```
>> does the helicopter see any cars?
<< no the helicopter does not see any cars

>> what is the current velocity?
<< the velocity is three meters per second

>> tell me about the buildings
<< ok

>> what is the current position?

<< the helicopter is beginning to see the church
```

The helicopter tells us that he sees the chuch, answering to the command "tell me about the buildings".

```
<< the helicopter has completed flying to the church

<< the helicopter is two meters from the church
```

The helicopter now answers the question "what is the current position?".

```
>> does the helicopter see any bodies?
<< no the helicopter does not see any body.

>> fly to position three hundred and forty slash minus seventy
<< the helicopter is about to start flying to three hundred forty
   slash minus seventy
<< it has completed flying to three hundred and forty slash minus
   seventy

<< the church is out of sight for the helicopter

>> dont tell me about the buildings
<< ok

>> fly to the school
<< i have four schools to choose between. which school do you mean?

>> which schools can i choose from?
<< you can choose between a red wood school and a black brick school
   and a green wood school and a green plate school

>> the green plate one
<< the helicopter is about to start flying to the plate green school

>> tell me about the cars
<< ok

>> what is the current position?
<< the helicopter is one hundred four meters from the plate green
```

```
   school

>> what is the altitude?
<< the altitude is twenty meters

>> what is the current position?

<< a red car is becoming visible to the helicopter
```

The helicopter explains that he sees a red car, answering to the command “tell me about the cars”.

```
<< the helicopter has completed flying to the plate green school

<< the helicopter is two meters from the plate green school
```

Answer to “what is the current position?”

```
>> can the helicopter fly to the boathouse under four minutes

<< the red car is out of sight for the helicopter
```

Answer to “tell me about the cars” This answer and the question before are being said simultaneously.

```
<< the helicopter can do it within the time
```

Answer to “can the helicopter fly to the boathouse under four minutes?”

```
>> there is a body west of the boathouse
<< ok

>> fly to it
<< do you mean fly to it?

>> yes
<< the helicopter is about to start flying to a body


>> what was the time when the helicopter took off?
<< the helicopter has seen several such actions but the latest one
   was at fifteen hours, twelve minutes and thirty five seconds

>> has the helicopter seen any cars today?
<< the helicopter has seen a red car

>> what is the position?
<< the helicopter is sixteen meters from the plate green school

<< the helicopter has completed flying to a body

>> does the helicopter see a body
<< the helicopter sees a body
```

```
>> fly home and land
<< do you mean fly home and land?

>> yes
<< the helicopter is about to start flying to three hundred ten
   slash minus one hundred forty

>> how much fuel does the helicopter have left?
<< pardon, please say again

>> how much fuel does the helicopter have left?
<< it has eighteen thirty one (1831) liters of fuel left
```

6.2 Revinge-0310-#02: Dialogue with WITAS UAV in flight

This dialogue demonstrates that the dialogue system is able to access the on-board system of the WITAS UAV for the purpose of obtaining parameters of the helicopter state and for issuing commands (here only camera commands).

The demo in Revinge also showed the use of the video during the dialogue. The transcript only shows this indirectly through the occurrence of requests to see a particular video segment. There is a parallel path for the video signal that is transmitted down from the helicopter to the ground, converted to mpeg in the dialogue system, and stored so that the operator is able to request either to see what the video camera sees at the same time, or to see a flashback from a certain timepoint or earlier event.

The dialogue itself is able to refer to timepoints for flashback both by absolute time, relative time ("fifteen minutes ago"), and by referring to an earlier event. However, since no events were reported during the flight, only relative-time reference could be demonstrated.

```
>> show me the video
<< ok

>> rotate camera ten degrees down
<< the helicopter is about to start rotating the camera
<< it has completed rotating the camera

>> show me the video
<< ok

>> show me the video from two minutes ago
<< here it comes

>> show me the video
<< ok

>> rotate camera forty five degrees right
```

```
<< it is about to start rotating the camera
<< it has completed rotating the camera

>> show me the video
<< ok

>> what is the current camera elevation?
<< the elevation is minus twenty eight degrees

>> rotate camera twenty degrees down
<< it is about to start rotating the camera
<< it has completed rotating the camera

>> show me the video
<< ok

>> show me the video
<< ok

>> show me the video
<< ok

>> what is the current camera azimuth?
<< the azimuth is one hundred sixty five degrees

>> rotate the camera one hundred and sixty five degrees left
<< it reports could not send message to camera action error
<< ok

>> show me the video
<< ok

>> show me the video
<< ok

>> show me the video
<< ok

>> rotate camera ten degrees down
<< it reports could not send message to camera action error

>> rotate camera ten degrees down
<< it has completed rotating the camera

>> show me the video

<< it is about to start rotating the camera
```

This should have come before "it has completed rotating the camera", but somehow this sentence didn't get there in time.

```
<< ok
```

Answer to “show me the video”

```
>> what is the current camera elevation?
<< the elevation is minus eighty five degrees

>> rotate camera thirty five degrees up
<< it is about to start rotating the camera
<< it has completed rotating the camera

>> show me the video
<< ok

>> what is the current camera azimuth?
<< the azimuth is minus forty four degrees

>> rotate camera forty five degrees right
<< it is about to start rotating the camera
<< it has completed rotating the camera

>> show me the video
<< ok

>> show me the video from five minutes ago
<< here it comes

>> show me the video from three minutes ago
<< here it comes

>> show me the video from four minutes ago
<< here is comes
```

6.3 UAVPatrol-0312-#01: Planning and scheduling a mission

This scenario is an example from work in progress, and is intended to give an impression of our system at present (first days of December, 2003). The ’UAV Patrol’ domain has been combined with a facility for planning and scheduling dialogue that works as follows. The operator first specifies a number of problems that need to be dealt with, without regard to order. The system then groups the problems (considered as ’goals’) by priority, and furthermore in groups of goals that use the same payload in the helicopter so that it is preferable to perform them together. It then expands the resulting, partially ordered sequence of goals to a corresponding plan for the helicopter, in the sense of a sequence of actions for it, taking the preconditions for the various actions into account. Finally, on command, it starts executing the plan and reports on the completion of steps in the plan.

The progress report may be omitted for some actions, for example when an ’unmount' action is directly followed by a 'mount' action: provided that the 'unmount' action succeeds, it is not necessary to report it separately.

Instructions to add or cancel goals may arrive while the plan is being executed. A subsequent replanning request establishes a new plan that respects the changes and which of course also omits those goals that have already been achieved.

The following is an example of actual dialogue with the system that illustrates these basic facilities.

```
>> we shall plan mission nov-30
<< ok what are the problems and the tasks at this point

>> it is time to check speed at the school
<< ok

>> check ice on road along lidingovagen
<< ok

>> there is an accident at the pharmacy
<< ok

>> schedule the mission
<< ok i have determined a schedule for this mission

>> start mission nov-30
<< ok
<< the helicopter has arrived to the pharmacy
<< it has completed obtaining pictures of the accident

>> cancel checking speed at the school
<< ok i have cancelled that task

>> reschedule the mission
<< ok i have determined a new schedule for this mission

>> proceed mission nov-30
<< ok
<< the helicopter has arrived to the base
<< it has completed unmounting present payload and mounting
     the ice sensor
<< it has arrived to lidingovagen
<< it has completed checking for ice on the road
<< mission nov-30 has now been completed
```

What happened here is that the dialogue system first made a plan involving actions for all the three given goals, and started executing it by feeding commands to the UAV one at a time. When the UAV had achieved its first goal and was on its way back to base in order to retool for the purpose of checking the speed limit at the school, the system received the command to cancel that goal. The dialogue system let the UAV proceed to the base anyway, but now for the purpose of retooling for ice detection. A revised plan for the only remaining goal was produced and executed step by step.

The planning and scheduling dialogue facility has been implemented as a plug-in using handles in the existing base system. It is intended that other extensions can be added in the same modular way, and one of the outcomes of the work will be a feedback on that hypothesis. The present example was run entirely within the DOSAR dialogue manager and using an elementary state transition simulator for updating the world state both at planning time and at execution time. However, the message interfaces are consistent with those used by D2W, so they are compatible with the UAV ground station and with the main DOSAR simulator.

Facilities for explaining the purpose of particular actions, and for inferring the need for adding and cancelling goals when new observations arrive, are next in turn for the continued work.

The point with this exercise is not the planning and scheduling per se, since those procedures do not offer any particular difficulties. Instead, it is the ensuing dialogue facilities that are of interest. The planning and scheduling routines have a number of aspects that add to the complexity of the domain: grouping tasks according to urgency and payload, adding and cancelling tasks, temporal constraints, subsumption relations between payloads, and so on. This complexity, although moderate, results in a corresponding complexity of the dialogue, and this is what is interesting for the project. In general, I feel that dialogue complexity is derived from domain complexity, and the key to managing dialogue is to 'attach' dialogue facilities to a domain model. The present exercise provides a testbed for trying out that approach.