

# Meaningful Conversation with a Mobile Robot

Johan Bos, Ewan Klein, Tetsushi Oka

ICCS, School of Informatics, University of Edinburgh

2 Buccleuch Place, Edinburgh EH8 9LW

Scotland, United Kingdom

{jbos, ewan, okat}@inf.ed.ac.uk

## Abstract

We describe an implementation integrating a spoken dialogue system with a mobile robot, which the user can direct to specific locations, ask for information about its status, and supply information about its environment. The robot uses an internal map for navigation, and communicates its current orientation and accessible locations to the dialogue system using a topological map as interface.

## 1 Introduction

Most research on spoken dialogue has focused on humans talking to virtual agents, often only reached at the end of a telephone line. Interesting challenges and opportunities arise when the interlocutor is a physically embodied mobile agent—for example, a robot. When we enter into dialogue with a robot, we can talk about the physical environment that we share with the robot, and we get a palpable indicator of dialogue success when an utterance such as *go to the corridor* produces the desired effect. In short, spoken dialogue research with mobile robots opens up a new vista for human-computer interaction design which goes beyond the current preoccupation with visual interfaces. In this paper, we give a short overview of the kind of dialogue that can be held with Godot, our robot, together with the architecture and technologies that we have implemented.

## 2 The Spoken Dialogue System

The spoken dialogue system allows the user to move Godot by giving it commands, to ask about its current location, or to provide it with new information. This section explains how speech

processing, language modelling, language understanding and dialogue management have been implemented.

### 2.1 Speech Recognition

Nuance's speaker-independent speech recognition system ([www.nuance.com](http://www.nuance.com)) allows language models to be specified using Speech Grammar rules. Rather than writing these directly, we compile them from a unification grammar for English. Moreover, instead of adopting the Speech Grammar slot-filling paradigm for semantic interpretation, our grammar builds a sophisticated, compositional semantics involving  $\lambda$  terms which are passed as the value of a single slot for the recognised sentence. As a result the output of the speech recognition stage is a semantic representation, and no further parsing is required before handing it over to the dialogue manager. In order to reduce perplexity, different grammars are loaded at different states in the dialogue.

### 2.2 Natural Language Understanding

Discourse Representation Theory (DRT) (Kamp and Reyle, 1993) is used for meaning representation in the system. The current implementation covers a wide variety of linguistic phenomena, including context-sensitive phenomena involving anaphora, presupposition, quantification, and plural descriptions. One crucial benefit of DRT is that it supports inference, using a translation from Discourse Representation Structures (DRSs) to formulas of first-order logic. Inference helps to detect inconsistencies in the dialogue and assists in disambiguation.

Inference invokes standard theorem proving techniques in a context sensitive way. The DRS representing the dialogue is combined with a DRS containing information about the current situation (i.e., the position of the robot and currently ac-

cessible locations), and translated into first-order logic. This is combined with further background knowledge (frame axioms and axioms about temporal states, plus ontological information), and the resulting formula  $\phi$  is sent to both a model generator and a theorem prover. If the theorem prover finds a counter-proof, we treat  $\phi$  as inconsistent information; conversely, if the model builder finds a model for  $\phi$ , we use the model to deduce what actions need to be performed by the robot, or to answer questions posed by the user.

### 2.3 Dialogue Management

We adopt the approach to dialogue move engines developed within TRINDI (Traum et al., 1999), in which an agent’s information state is updated on the basis of observed dialogue moves, leading to the selection of a new dialogue move to be performed by the agent. Our notion of information state consists of `grammar` (the grammar currently loaded by the speech recogniser), `contact` (whether or not there is communicative contact with someone), `input` (the results of speech recognition), `nextmoves` (the next dialogue moves to be realised by the robot), `lastmoves` (the latest dialogue moves produced by the user), and `interpretation` (consisting of the DRS of the ongoing dialogue and a first-order model generated for it).

An update rule links preconditions to effects. The dialogue manager repeatedly computes the effects of those update rules whose preconditions are satisfied by the current information state. Preconditions are expressed in terms of current values in the information state, while the effects will change these values. The 26 update rules in our current system deal with establishing contact with the user, initiating clarification dialogues (when the recognition confidence score is below a certain threshold), answering questions, acknowledging requests and confirming or denying statements.

System output is generated from templates and synthesised by the Festival TTS system ([www.cstr.ed.ac.uk/projects/festival/](http://www.cstr.ed.ac.uk/projects/festival/)). Utterances are coded in SABLE format (an XML standard for speech synthesis markup) in order to assign appropriate prosodic contours.

## 3 The Mobile Robot System

In this section we describe the hardware of the robot itself, the internal map representation it uses for navigation and communication with the dialogue manager, and the navigation component.

### 3.1 Godot, the Robot

Godot is an RWI Magellan Pro mobile robot platform with an on-board PC running Linux (Fig. 1). It is cylindrical, about 50 cm high and 41 cm in diameter. Godot’s sensor equipment consists of 16 sonars, infrared sensors, and bumpers, an odometry component, and a colour video camera with a pan-tilt unit. The on-board computer is connected to the local network via a wireless



Fig.1: Godot.

LAN interface. Godot’s navigation system relies on sonars, infrared sensors and odometry, and not the bumpers or the camera. However, the camera is used as live feedback to the user (Fig. 2), who is able to look “through the eyes” of Godot while engaging in a dialogue.



Fig.2: Image Viewer

### 3.2 The Internal Map

Godot moves about in the basement of our department and uses an internal map for navigation. It has two levels of representation: a geometrical and a topological layer.

The geometrical layer uses an occupancy grid to represent occupied and free space in the environment. The topological layer is automatically constructed from the occupancy grid by subdividing the free space into distinct topological regions corresponding to rooms or parts of the corridor (Fig. 3). This is possible by creating a Generalised Voronoi Diagram (Latombe, 1991; Thrun, 1998; Theobalt, 2000).

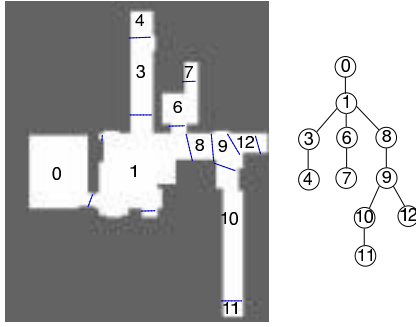


Fig.3: Representation of the Environment

The numbers in the geometrical map shown in Fig. 3 are identifiers of topological regions which can be seen as nodes of an undirected graph. There is a further layer of representation interfacing the map of the navigation system with a vocabulary of semantic symbols used by the dialogue system. This layer extends the topological map by associating semantic labels to regions. These descriptions can be arbitrarily complex. For instance, the DRS  $\lambda p.(\langle [x, y], [office(x), of(x, y), tim(y)] \rangle; p(x))$  is used as a label denoting Tim's office.

### 3.3 Navigation Module

The navigation module loops by reading sensory input and writing motor commands at regular intervals. The sensory input comprises the readings of the sonars, infrared sensors and odometry. The motor commands govern translational and rotational velocity of the robot, and pan/tilt/zoom of the camera unit. They are triggered by readings of the sensors or by commands communicated from the dialogue manager.

The behaviour of the navigation module is triggered by the last command from the dialogue manager. There are primitive commands such as *go(Distance, Speed)*, *turn(Angle, Speed)* and *look(Pan, Tilt)* as well as complex commands like *follow\_wall(Distance)* or *move\_to\_region(N)*. Since the navigation module accepts commands at any time step, it is possible to interrupt on-going behaviour. It is also possible to change the parameters of currently executing commands. Thus *set\_rotation(Angle)* can be used to change the robot's belief about its current orientation.

Moving to a particular grid cell or region requires accessing information about the environment that is stored in the internal map. The naviga-

tion module keeps track of the current position and orientation using odometry. It can detect walls using the readings of the infrared sensors and correct translational and rotational errors of the odometry by comparing them with the geometrical map.

The topological map is used to compute the shortest path from one region to another. This path is executed by setting subgoals for navigation, because there can be walls or obstacles between two regions. The position of the centre of a region lying along the path is also obtained from the topological map. Given the grid cell of the centre of the next region, the navigation module plans a path from the current location to the centre of the region on the geometrical map, by means of a distance transform path planner (Theobalt, 2000). After planning a path, one of the cells in the path is selected as the next subgoal. When the robot enters a new region, a new path is planned on both of the layers.

Motor commands are computed at every time step based on the current position and orientation, the location of subgoals and the sensor readings. The robot can avoid obstacles using the sonars and infrared sensors when it is moving to another region, since there is some latitude in how precisely it traces a path with respect to the grid cells.

## 4 Putting Everything Together

The dialogue system is implemented on top of the Open Agent Architecture (OAA). Godot, however, uses CORBA for inter-process communication. Fig. 4 shows how the two systems are implemented and combined into one working system.

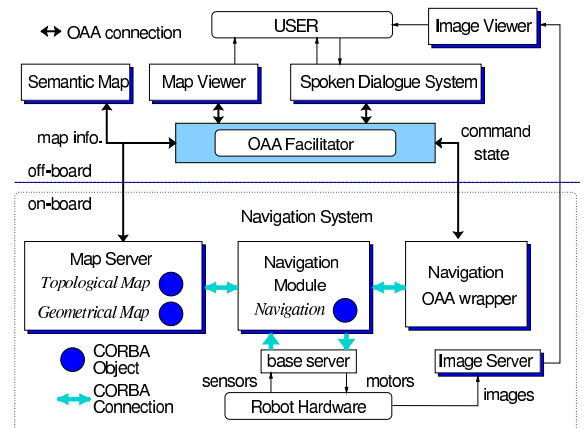


Fig. 4: System Architecture

## 4.1 The Dialogue System

The dialogue system is implemented on top of OAA 2.1.0, ([www.ai.sri.com/~oaa/](http://www.ai.sri.com/~oaa/)). OAA agents can run on different machines and even on different platforms, and they communicate with each other by posing *solvable*s via a facilitator.

The system contains OAA-agents for speech recognition (Nuance 8.0), speech synthesis (Festival), resolution (resolving ambiguities), inference (building models or finding counter-proofs), and dialogue management. There are two further agents for model building (MACE, [www-unix.mcs.anl.gov/AR/mace/](http://www-unix.mcs.anl.gov/AR/mace/)) and theorem proving (SPASS, [spass.mpi-sb.mpg.de/](http://spass.mpi-sb.mpg.de/)).

The system is coordinated by the dialogue manager, triggering OAA-solvable for speech recognition, speech synthesis and inference. It can also request information as the the robot's position and its current environment.

## 4.2 The Navigation System

The navigation system consists of three components which run concurrently on the on-board PC of Godot (Fig. 4). They are Linux processes which communicate with each other via CORBA objects. The map server stores Godot's internal representation of the environment as CORBA objects. Information stored in these objects can be retrieved by the navigation and dialogue modules via CORBA or OAA connections.

The navigation module has a CORBA object to store information that it shares with the dialogue system. The OAA-wrapper has access to this object and OAA agents in the dialogue system can communicate with the navigation module via the wrapper. The dialogue system can monitor the current state of the navigation system, e.g. the current location of the robot in the map, and can send commands to it.

## 4.3 Running the System

The system is distributed across a Linux laptop (the dialogue system) and the robot's on-board PC (running the navigation system). Users can start a new dialogue simply by addressing the robot, and the robot reacts in real time. Although we have not yet reached the stage of carrying out usability studies, we have held informal tests where visitors

to our department who are unfamiliar with Godot have controlled it and its camera with success.

## 5 Conclusions and Future Work

Although it is tempting to work with simulated 'embodied' agents, we believe that the real test of adequacy involves confronting the familiar problems of noisy sensor data and real hardware. Using the OAA architecture, we have developed an effective interface between natural language semantics and the robot control layer, thus enabling users to refer to locations in a natural way, rather than resorting to expressions like *go to grid cell 45-66* or *you are in region 12*. The framework is to a large extent domain-independent: a change of environment would only require a change of the internal map and possibly a new lexicon.

Future work will address the interpretation of vague expressions (*the end of the corridor*), metonymic expressions (*go to the door*, where an artifact is interpreted as a location), together with commands which require Godot to reason and talk about its current activities (*continue going to the kitchen*). We are further planning to extend the system with a face-recognition component to enrich the possibilities of natural interaction.

## Acknowledgements

Part of this work was supported by the EU Project Magicster (IST 1999-29078). We thank Nuance for permission to use their software and tools.

## References

- H. Kamp and U. Reyle. 1993. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Kluwer, Dordrecht.
- J. C. Latombe. 1991. *Robot Motion Planning*. Kluwer Academic Publishers.
- C. Theobalt. 2000. Navigation on a mobile robot. Master's thesis, University of Edinburgh.
- S. Thrun. 1998. Learning maps for indoor mobile robots. *Artificial Intelligence*, 99(1): 21–71.
- D. Traum, J. Bos, R. Cooper, S. Larsson, I. Lewin, C. Matheson, and M. Poesio. 1999. A model of dialogue moves and information state revision. Trindi Report D2.1.