An Agent Architecture for the Implementation of Spoken Dialogue Systems in the Home Environment Domain

Jose F. Quesada[1] and J. Gabriel Amores[2]

[1] Department of Computer Science and Artificial Intelligence
University of Seville
jquesada@cica.es
[2] Department of English
University of Seville
jgabriel@us.es

Abstract. This paper presents an architecture for the implementation of spoken dialogue systems. The main constraints for the application of spoken dialogue systems to the home environment domain are described, as well as the main characteristics of all the agents involved. The architecture allows an efficient treatment of complex linguistic phenomena (reference and anaphora resolution, quantification, task accommodation, undo operations, ...) and isolating the low level features of the devices (by means of the Device Manager) from the high level modules in charge of the Natural Language Understanding, Dialogue, Action and Knowledge Management.

1 Introduction

The DHomme project[3] has addressed the theoretical challenges in language understanding and dialogue management for controlling and querying multiple networked devices from inside or outside the home.

One of the main challenges of this research trend was the integration of multiple technologies, mainly, speech recognition and synthesis, natural language understanding, dialogue management, knowledge specification, representation and inference, action management, and device control.

This paper describes one of the main results obtained from the project: an agent architecture (specifically implemented in OAA [Cheyer & Martin 2001]) for the design and implementation of spoken dialogue systems. This architecture was applied to the implementation of a spoken dialogue system in the home environment. Nevertheless, it was designed as a general architecture independent

[3] DHomme (Dialogues in the Home Environment Environment) has been funded by the 5th Framework Program of the Europoean Union, under the Information Society Technologies Programme: http://www.ling.gu.se/projekt/dhomme. The authors would like to thank all the participants in the project for their discussions and contributions on different topics covered by this paper.

of the domain, and is currently being used in the implementation of a spoken dialogue system in the automatic telephone operator domain[4].

Section 2 describes the agents in the global agent architecture proposed taking into account the design constraints for the global system. Next, section 3 concentrates on the Knowledge, Dialogue and Action Manager and presents the strategy for the representation of the semantic structure of a house and the algorithm for reference resolution. Also, this section analyses the linguistic coverage of the spoken dialogue system by means of an example. Finally, section 4 connects the future work with the promising research trend on Ambient Intelligence.

2 The Agent Architecture: Functional Description

2.1 Design Constraints

In some sense, the agent architecture designed is a consequence of the main constraints imposed by the intented goals of the project:

- **Standards**: The interface should *leverage existing or incipient standards* in Home Automation and spoken language technology.
- **Plug and Play**: The interface should support a 'Plug and Play' protocol, in the sense that *new devices can be added to the network* without requiring user configuration of the spoken language components.
- **User control**: The interface should allow the user to *query, control and program* individual devices, or collections of interacting devices.
- **Language localisation**: The interface should support *language localisation* by device manufacturers and service suppliers.
- **Upgrading**: The interface should allow *upgrading* for increasingly intelligent devices.

2.2 Overall Description of the Agents Architecture

Figure 1 shows the agents included in the proposed architecture:

Speech Recogniser This module is in charge of the translation from the acoustic signal corresponding to the spoken user's utterance to a string of words. Both ViaVoice and Nuance speech recognisers have been integrated into the system. An OAA wrapper has been developed for each recogniser.

Synthesiser The goal of this agent is to convert the string of words generated by the dialogue manager into a spoken output. Similarly, an OAA wrapper has been developed for the integration of Festival's speech synthesiser into the system.

[4] This work is been carried out as part of the Siridus project (Specification, Interaction and Reconfiguration in Dialogue Understanding Systems), funded by the IST (Information Society Technologies) Programme of the 5^{th} Framework Program of the European Union: `http://www.ling.gu.se/projekt/siridus`

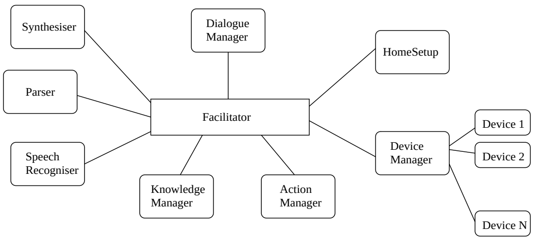

Fig. 1. Agent Architecture

Parser and Semantic Interpreter The goal of this module is to convert the n–best string of words obtained by the Speech Recogniser into a set of dialogue moves, and incorporate some contextual interpretation during the process. This module transforms each utterance into a list of (one or more) feature structures according to the DTAC protocol [Amores & Quesada 2000].

Dialogue Manager The dialogue manager's task is to interpret the utterances produced by the user taking into account three sources of information:

- The previous dialogue, represented internally as an ordered list of dialogue information states, each one being a DTAC structure.
- The current state of the external system, that is, the state of the set of devices controlled by the system, obtained from the Knowledge Manager.
- The specification of the dialogue model, that is, the set of rules which specify the behaviour of the dialogue system [Quesada, Torre & Amores 2000].

First, the dialogue manager will have to match the DTAC representation of the incoming dialogue move with the rules of the dialogue model. Next, the rule triggered by the DM will indicate the set of expectations and actions to be completed as part of its execution. Actions may include linguistic output (for instance, when the user asks the system about some information, or when the dialogue manager detects underspecification or ambiguity), queries to the knowledge database (in order to determine the current state of the different devices, or to disambiguate device descriptors) and instructions sent to the Action Manager.

Knowledge Manager This agent contains both the static (structure of the house, general ontology, ...) and dynamic (installation, configuration and state of the devices) knowledge involved in the task.

The Knowledge Manager is in charge of storing and organising the information related to the devices installed in the house regarding their localisation, characteristics, and relations with other devices. This information is organised as a semantic network.

Action Manager The Action Manager controls and coordinates the different devices and offers a common and general interface to the rest of the dialogue system.

Also, the Action Manager translates the high level abstracted action specification into a low level action rule which takes into account the relations between the device descriptors and their physical identifiers.

Finally, the Action Manager incorporates new complex actions to the dialogue system as a whole. That is, the Action Manager may provide new global actions as a combination of different primitive actions. For instance, at the device level there is no way to execute the action *Turn on all the lights in the outside* as there is no "all the lights in the outside" device. By using the information stored in the Knowledge Manager, the Action Manager may execute this action as a combination of different primitive actions.

HomeSetup This agent is in charge of the following tasks: Simulation of a home environment (figure 2), Device Installation (the user may install and configure the set of relevant devices, including information regarding their type, location, and attributes or descriptors), Simulation of the device functionality, and Monitoring of simulated and real devices.

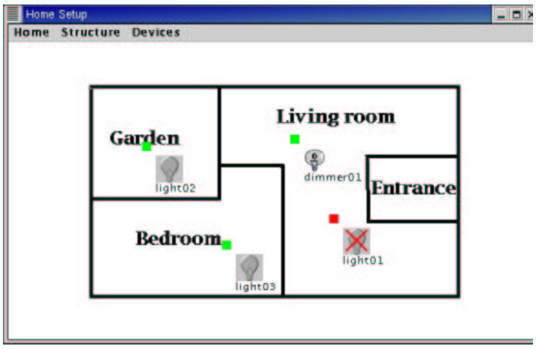

Fig. 2. HomeSetup GUI: The Simulation of a House

Device Manager The HomeSetup agent permits the control of real devices through the Device Manager agent.

This agent, implemented in C, allows us to make independent the design and implementation of the dialogue manager and configuration of devices from the technological infrastructure used for the control of the actual devices.

In its current version, the Device Manager agent is capable of controlling devices through the X10 and Lonworks protocols. However, the advent of new standards for the control of real devices would only require the introduction of those standards in this agent, keeping the rest of the implementation intact.

3 Knowledge, Dialogue and Action Management

3.1 Knowledge Management: From Graphical Representation to Semantic Graphs

Figure 2 shows a complete home simulation scenario (provided by the HomeSetup agent) in which we may observe the spacial organisation of the house, along with the devices currently installed in it and their status (on, off or unplugged).

Next, we analyse a real spoken dialogue with the system which will help to explain its capabilities.[5]

Initial State: At the beginning (correponding to the situation represented in Figure 2), there are 4 devices installed: an unplugged light at the living room (light01), a dimmable light switched off at the living room (dimmer01), and two lights in the garden (light02) and the bedroom (light03), both plugged and turned off.

The HomeSetup GUI allows the user to configure the devices, incorporating information about the type of the device, its location and properties (descriptors). Let us assume that we have associated the following characteristics of the 3 plugged devices in the example above:

1. Light01: living_room, light, small
2. Dimmer01: living_room, dimmer, blue
3. Light02: garden, light, yellow, big
4. Light03: bedroom, light

This information will be represented by the Knowledge Manager as a semantic graph as shown in Figure 3.

3.2 Reference Resolution

Let us consider that the user prompts the system to turn on all the indoor lights:

- **U(1)**: Could you please turn on all the indoor lights?

The Natural Language Understanding agent (Parser and Semantic Interpreter) will obtain the following formal representation (using the DTAC protocol) for the previous utterance:

[5] U(n) and S(n) represents the n-th utterance by the user and system, respectively. Comments are written in italics.

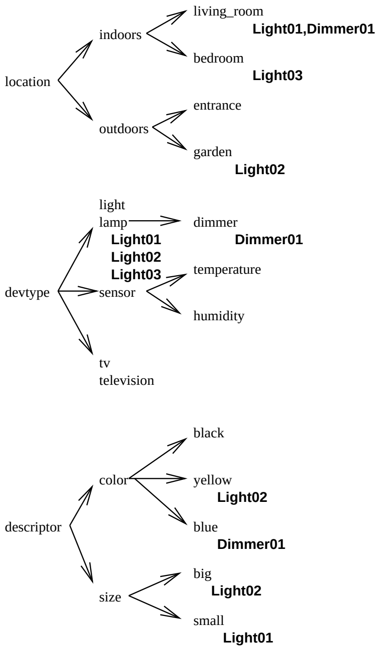

Fig. 3. The knowledge representation of a house

$$\begin{bmatrix} DMOVE & : specifyCommand \\ TYPE & : CommandOn \\ ARGS & : [DeviceSpecifier] \\ CONT & : \\ DeviceSpecifier & : \begin{bmatrix} DMOVE & : specifyParameter \\ TYPE & : DeviceSpecifier \\ ARGS & : [DeviceType, Location] \\ CONT & : \\ QUANT & : all \\ DeviceType & : \begin{bmatrix} DMOVE & : specifyParameter \\ TYPE & : DeviceType \\ ARGS & : \\ CONT & : light \end{bmatrix} \\ Location & : \begin{bmatrix} DMOVE & : specifyParameter \\ TYPE & : Location \\ ARGS & : \\ CONT & : indoor \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Upon receiving this structure, the Dialogue Manager detects that it is a command of type CommandOn (turn on) which must be executed over one or more devices (there is a quantification mark).

In order to resolve the device, the Dialogue Manager sends a request of type k-DevRes (knowledge-based Device Resolution) to the Knowledge Manager module, passing the information known about it:

k-DevRes(location:indoor, devtype:light)

The Knowledge Manager traverses the semantic graph (Figure 3) looking for the values for the each of the attributes above. That is, it searches for *indoor* and *light*. Each of these searches obtains the union of the devices hanging from that node, and all its descendents (thus, the search in the light node obtains also the devices of type dimmer, since dimmer has been defined as a daughter or subtype of light).

In a first phase, then, it obtains the following result:

k-DevRes-SolveNode(indoor) = (Light01,Light03,Dimmer01)
k-DevRes-SolveNode(light) = (Light01,Light02,Light03,Dimmer01)

In a second phase, it obtains the intersection of the searches performed on each node, and deletes the unplugged devices (in this case, Light01). Thus, the reference resolution of this query will contain Light03 and Dimmer01.

k-DevRes(location:indoor, devtype:light) = {Light03, Dimmer01}

Finally, the Dialogue Manager will send to the Action Manager the command SwitchOn applied to the devices Light03 and Dimmer01. In turn, the Action Manager will update the state of the devices in the HomeSetup simulation, and will send an instruction to the Device Manager to control the real devices.

So, the new situation is: Light01: unplugged, Light02: off, Light03: on, Dimmer01: on.

3.3 Dialogue Management: Complex Linguistic Phenomena

The previous example has illustrated the use of quantification as part of the specification of devices. Furthermore, the dialogue manager supports a wide range of linguistic phenomena. Next, we illustrate this functionality continuing with the example above in the state obtained in the previous section.

Natural Command Languages, Underspecification and Expectations

- **U(2)**: Turn off.
- **S(1)**: What device would you like to turn off?
- **U(3)**: The one in the bedroom.

In this example, the system detects that in order to turn off a device, the user must supply the descriptor of the actual device. This functionality is implemented as an expectation of the dialogue manager ([Quesada, Torre & Amores 2000]).

As a result the system turns off the bedroom light (Light03).

Queries, Anaphora and Multiple Commands

- **U(4)**: Is there any light turned on in the house?
- **S(2)**: Yes, the dimmer light in the living room is on.
- **U(5)**: Turn it off, please, and turn the garden light on.

The dialogue manager allows up to 12 different types of queries. Sentence U(4) illustrates one of these possibilities. Sentence U(5) includes multiple commands and also an anaphoric expression (turn it off) which must be solved taking into account the dialogue history. As a result of this subdialogue, the system should turn off Dimmer01 and turn on Light02.

Other advanced linguistic phenomena covered by the system include error repairs, positive and negative task accommodation, undo operations, conjuntions of device specifiers, default interpretations, plug and play (dynamic incorporation of new devices or change of the characteristics of the devices installed, even change of location).

3.4 Action Management and Device Control

In order to illustrate, analyse and evaluate the whole system in a more natural scenario we have designed, configured and installed a small house prototype. The only difference with respect to a 'real' house is its size, since our prototype had to be small enough for its manipulation and transportation.

The prototype consists of a set of lights installed over a board, in which the floor plan of a house has been drawn.

Figure 4 shows a front photograph of the prototype, with labels (in Spanish) for each 'room' in the house.

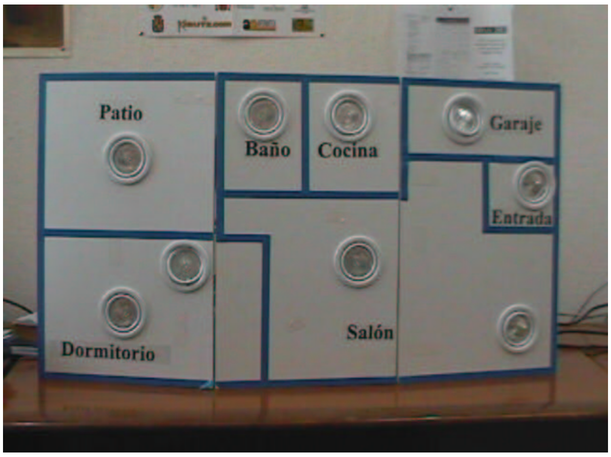

Fig. 4. A house prototype with real devices

4 Conclusion and Future Work: Towards Ambient Intelligence

This paper has concentrated on the description of an agent architecture for the design and implementation of spoken dialogue systems, along with its application to a real scenario in the home environment.

The main characteristics of the system are:

- It includes a dialogue management system which allows flexible interaction using natural language commands.
- It has been implemented over a distributed agent architecture.
- It allows speech input and output.
- It has been especially designed for the control of devices in a home environment.
- The system supports Plug and Play functionality.

Although in its current version the system supports a reduced number of control commands over real devices, the design and implementation of agents has been carried out having in mind future demands imposed by more intelligent devices.

Even though the generic goal underlying the goal of Ambient Intelligence [Ducatel et al 2001] was beyond the specific goals to be achieved by the D'Homme project, there exist points of coincidence between both approaches.

Perhaps the most relevant point in the Ambient Intelligence declaration lies in the necessity of achieving the control of devices through natural language and flexible communication with them. Obviously, the incorporation of more intelligence in the devices will necessarily involve more sophistication in the linguistic modules. Nevertheless, a critical study and evaluation of the agent architecture proposed does not foresee a serious *a priori* incompatibility with such linguistic functionality. Rather, the use of a machine translation engine as a lexical, syntactic and semantic analysis module ensures the capability of dealing with complex natural language expressions.

On the other hand, the HomeSetup and Knowledge Manager agents are not limited either by the current functionality of the devices being controlled in the current version. Perhaps the HomeSetup should evolve into an agent with more spatial coverage (something like an AmbientSetup).

Finally, the agent which would require more modifications is the Action Manager, which should support a more sophisticated set of actions. Furthermore, in an Ambient Intelligence environment, the proactive behaviour of devices is a crucial feature.

References

[Amores & Quesada 2000] Amores, J. G., Quesada, J. F. 2000. *Dialogue Moves in Natural Command Languages.* Deliverable 1.1. Siridus project.

[Amores *et al* 2001] Amores, J.G., Berman, A., Bos, J., Boye, J., Cooper, R., Ericsson, S., Holt, A., Larsson, S., Milward, D., Quesada, J.F. *Knowledge and Action Management in the Home Device Environment.* Deliverable 4.1. DHomme project. October 2001.

[Bos 2001] Bos, Johan. DORIS 2001: Underspecification, Resolution and Inference for Discourse Representation Structures. In Patrick Blackburn and Michael Kohlhase, editors, *ICoS-3, Inference in Computational Semantics*, pages 117–124, 2001.

[Boye *et al* 2001b] Boye, Johan, Ian Lewin, Colin Matheson, James Thomas, and Johan Bos. 2001. *Standards in Home Automation and Language Processing.* Deliverable D1.1. D'Homme project. November 2001.

[Cheyer & Martin 2001] Cheyer, Adam; Martin, David. (2001). The open agent architecture. *Journal of Autonomous Agents and Multi–Agent Systems*, **4(1/2)**, 143–148, March 2001.

[Ducatel et al 2001] Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman, J-C. (compilers). 2001. *Scenarios for Ambient Intelligence in 2010.* Information Society Technologies Advisory Group and Institute for Prospective Technological Studies (Joint Research Center). `http://www.cordis.lu/ist/istag.htm`

[Eckert & Strube 2001] Eckert, Miriam; Strube, Michael. 2001. Dialogue Acts, Synchronising Units and Anaphora Resolution. Journal of Semantics 17(1). pp 51-59.

[Fraser & Thornton 1995] Fraser, N.M. and J.H.S. Thornton. Vocalist: A robust, portable spoken language dialogue system for telephone applications. In *"Proc. of Eurospeech '95*, pages 1947–1950, Madrid, 1995.

[Giachin & McGlashan 1996] Giachin, E. and S. McGlashan. 1996. Spoken Language Dialogue Systems. In Corpus-based Methods in Language Processing edited by G. Bloothooft and S. Young, Kluwer, The Netherlands, 1996.

[Kamp & Reyle 1993] Kamp, Hans, and Uwe Reyle. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT.* Kluwer, Dordrecht, 1993.

[Larsson & Traum 2000] Larsson, S. and D. Traum Information State and dialogue management in the TRINDI Dialogue Move Engine Toolkit. In *Nat.Lang. Engineering*, volume 6, 2000.

[Lemon et al 2001a] Lemon, Oliver, Anne Bracy, Alexander Gruenstein, and Stanley Peters. 2001. "A Multi-Modal Dialogue System for Human-Robot Conversation", In proceedings NAACL 2001.

[Lemon et al 2001b] Lemon, Oliver, Anne Bracy, Alexander Gruenstein, and Stanley Peters. 2001. The WITAS Multi-Modal Dialogue System I", In proceedings EuroSpeech 2001.

[McGlashan & Axling 1996] McGlashan, S. and T. Axling. 1996. Speech Interfaces to Virtual Reality. In the Proceedings of the International Workshop on Speech and Computers, St. Petersburg, Russia, 1996

[Milward 2001] Milward, D., Knight, S. (eds). *The Initial D'Homme System Architecture* Deliverable 2.1. D'Homme project. May 2001.

[Palomar & Martinez-Barco 2001] Palomar, M., Martnez-Barco, P. 2001. Computational Approach to Anaphora Resolution in Spanish Dialogues. Journal of Artificial Intelligence Research 15 (2001), pp. 263-287.

[Quesada, Torre & Amores 2000] Quesada, J. F., Torre, D. & J. G. Amores. 2000. *Design of a Natural Command Language Dialogue System* Deliverable 3.2. Siridus project. December 2000.

[Quesada et al 2001] Quesada, J. F., J. G. Amores, J. Bos, S. Ericsson, G. Gorrell, S. Knight, I. Lewin, D. Milward, M. Rayner. 2001. *Configuring Linguistic Components in a Plug and Play Environment.* Deliverable 3.1. D'Homme project. October 2001.