Knowledge–based Reference Resolution for Dialogue Management in a Home Domain Environment

J. F. Quesada
University of Seville
jquesada@cica.es

J. G. Amores
University of Seville
jgabriel@us.es

Abstract

Several features of spoken dialogue systems require advanced reference resolution strategies. This paper presents a knowledge–based strategy for reference resolution implemented over an agent architecture which differentiates the Knowledge and Action Managers from the Dialogue Manager. The strategy, in conjunction with the ability of the Dialogue Manager to manipulate the dialogue history, has been applied to a home environment domain to solve successfully different phenomena such as quantification, anaphoric expressions and coordination.

1 Introduction

Reference Resolution plays a crucial role in spoken dialogue systems:

- The cognitive structure and content of linguistic utterances incorporates multiple phenomena that require specific strategies to solve their reference. For instance, anaphoric expressions (Eckert & Strube 2001; Palomar & Martinez-Barco 2001), underspecification (Bos 2001), presupposition (Hulstijn 1996; Lemon et al 2001b), or quantification (Cooper 1993).
- In application–oriented dialogue systems (Giachin & McGlashan 1996) there is a representation gap between the logical form manipulated by the semantic interpreter and dialogue manager components, and the interface to the real elements controlled at the application level (Kamp & Reyle 1993; Fraser & Thornton 1995; Larsson & Traum 2000)
- Multimodality adds additional constrains to the reference resolution problem, such as the use of anaphoric (Lemon et al 2001a; Lemon et al 2001b; Lemon et al 2001c), or even deictic anaphoric expressions (Eckert & Strube 1999).

This paper describes the knowledge-based device resolution (k-DevRes) strategy which has been implemented in the Spanish demonstrator developed by the University of Seville within the D'Homme project (DHomme Project 2001). This strategy may be characterised as a *Knowledge–based Device Resolution Algorithm*, given the central role played by the Knowledge Manager module in the resolution process.

In the context of a home environment with a large number of devices, one of the most important tasks to be carried out by the modules in charge of contextual interpretation and dialogue management involves the identification of the device or devices which are the object of a command.

There are two main challenges at this level.

1. First, Device Resolution is a task in which several components or agents are usually involved (Milward 2001; McGlashan & Axling 1996). Even though the semantic analysis module

provides the formal representation of the user's instruction, the identification of the device or devices to which the desired command has to be applied requires both static (structure of the house, etc.) and dynamic information (connected devices, their characteristics, state, etc.) which is stored in the Knowledge Manager. On the other hand, taking into account the previous dialogue history, the Dialogue Manager may apply different disambiguation strategies.

2. Second, and related to the previous point, Device Resolution must incorporate the resolution of phenomena such as anaphoric expressions, question accommodation, quantification, etc. This, in turn, imposes a higher degree of coordination and integration between the different modules involved.

Section 2 analyses the main agents involved in the resolution of devices by means of a simple example. Plug and Play functionality is achieved as a consequence of the agent architecture and the communication interface designed between them. The main function at this level is situated between the Dialogue Manager and Knowledge Manager agents involved in the resolution of devices, which is studied in section 3.

The strategy, in conjunction with the ability of the Dialogue Manager to manipulate the dialogue history has been applied to a home environment domain to solve successfully different phenomena such as quantification, anaphoric expressions, coordination, etc.

Specifically, the information returned by the Device Resolution function allows the detection of ambiguity and the design of collaborative subdialogues for its resolution. Furthermore, this strategy has been applied to solve disambiguation subdialogues, error repairs, reaccommodation, task accommodation, default actions, and undo operations.

2 Knowledge Specification, Knowledge Management and Action Management

This section describes the main features of the spoken dialogue system (Quesada et al 2001) in which we have implemented the reference resolution strategy:

- 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. That is, specific agents connect the dialogue management engine with the modules of speech recognition and synthesis.
- It has been especially designed for the control of devices in a home environment.
- The system supports Plug and Play functionality, both weak and strong. That is, the design of the agent architecture permits the dynamic incorporation of new devices and the automatic reconfiguration of the Action Manager, Knowledge Manager and Dialogue Manager agents.

From the point of view of the device resolution, the agent architecture includes a specific agent for the specification of the domain knowledge (the HomeSetup agent), which permits the design of a semantic structure for a house, or, in general, for the environment of use of the system. The semantic structure is represented by means of a tree, which is internally stored as a graph. Three main nodes are distinguished in the semantic graph: location, devtype and descriptor, which correspond to the Location, DeviceType and Descriptor types of the representation formalism used (Quesada et al 2001).

The GUI included as part of the Home-Setup agent allows the association of real devices installed in the house with their semantic structure. Let us consider, for instance, the following four devices with their corresponding characteristics (location, device type and descriptors):

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

Figure 1 shows the semantic graph represented in the Knowledge Manager, which includes synonyms, and the devices associated with each node in the graph.

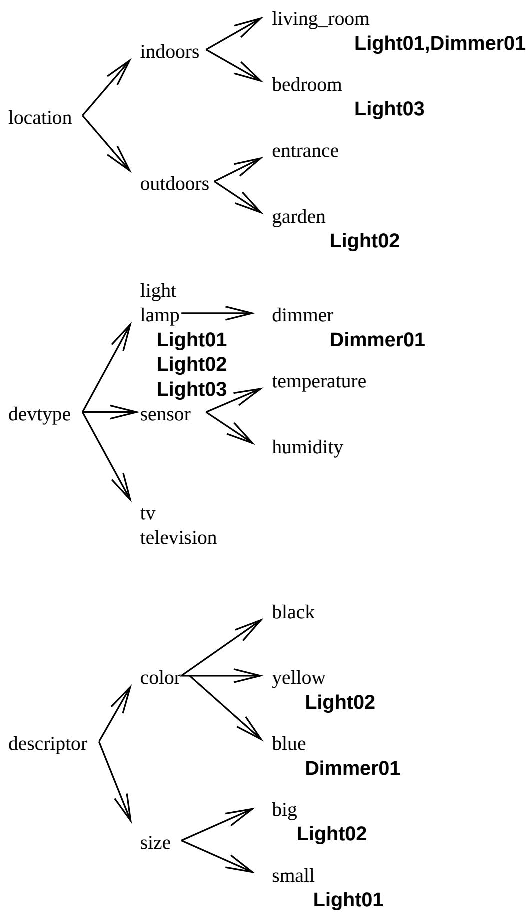

Figure 1: Semantic graph, including nodes, synonyms and devices.

2.1 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 formal representation (using the DTAC protocol) for the previous utterance (Figure 2).

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 1) 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.

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

Figure 2: Formal (DTAC) representation of the utterance: *Could you please turn on all the indoor lights?*

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.

2.2 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.

2.2.1 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).

2.2.2 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).

2.3 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 3 shows a front photograph of the prototype, with labels (in Spanish) for each 'room' in the house.

3 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.

Figure 1: A house prototype with real devices

- 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, J. G., Quesada, J. F. 2000. *Dialogue Moves in Natural Command Languages.* Deliverable 1.1. Siridus project.

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

Cooper, Robin. 1993. Generalised quantifiers and resource situations. In P. Aczel and D. Israel and Y. Katagiri and S. Peters editors, *Situation Theory and its Applications, v.3*, pages 191–212, CSLI and University of Chicago, Stanford, 1993.

http://www.ling.gu.se/projekt/dhomme

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, Miriam; Strube, Michael. 1999. Resolving Discourse Deictic Anaphora in Dialogues. In EACL '99 (pp.37-44).

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

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, 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.

Hulstijn, J. 1996. Presupposition accommodation in a constructive update semantics. In Durieux, G., Daelemans, W., and Gillis, S. (eds.), Proceedings of CLIN VI. University of Antwerp.

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

Lemon, Oliver, Anne Bracy, Alexander Gruenstein, and Stanley Peters. 2001. Information States in a Multi-modal Dialogue System for Human-Robot Conversation. In Proceedings Bi-Dialog, 5th Workshop on Formal Semantics and

Pragmatics of Dialogue, pages 57 - 67, 2001.

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

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

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, D., Knight, S. (eds). *The Initial D'Homme System Architecture* Deliverable 2.1. D'Homme project. May 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, J. F., Torre, D. & J. G. Amores. 2000. *Design of a Natural Command Language Dialogue System* Deliverable 3.2. Siridus project. December 2000.

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.