# A Dialogue Manager for an Intelligent Mobile Robot

Marcelo Quinderé, Luís Seabra Lopes and António J. S. Teixeira

IEETA, Departamento de Electrónica e Telecomunicações
University of Aveiro
3810-193 Aveiro, Portugal

**Abstract.** This paper focuses on a dialogue manager developed for Carl, an intelligent mobile robot. It uses the Information State (IS) approach and it is based on a Knowledge Acquisition and Management (KAM) module that integrates information obtained from various interlocutors. This mixed-initiative dialogue manager handles pronoun resolution, it is capable of performing different kinds of clarification questions and to comment information based on the current knowledge acquired.

## 1 Introduction

Personal robots, intelligent service robots capable of performing useful work in close cooperation/interaction with humans, are expected to be the next generation of robots. "Integrated Intelligence" [10] identifies an approach to building such agents in which the integration of key aspects is considered, including linguistic communication, reasoning, reactivity and learning.

This is the scope of CARL (*Communication, Action, Reasoning and Learning in Robotics*), a research project started in our institute in 1999, in the framework of which a robot prototype was developed, Carl [9]. The software architecture of Carl, which is based on the Open Agent Architecture (OAA) [6], uses a community of agents to handle general perception and action, display appropriate emotions through an animated face, process natural language and manage the robot.

Carl has been using a dialogue management approach based on finite state machines. In each predefined state, state transitions specify which actions the robot is supposed to execute under different conditions and which new states are reached. Several speech acts are supported including declarations (tell), questions (ask, ask_if) and commands (achieve). A Knowledge Acquisition and Management (KAM) module [11] integrates information obtained from different interlocutors, even if they are contradictory, and provides replies to received questions. The dialog management approach used until now has several limitations. It follows a mostly single-initiative strategy, since most dialogues are started by the user. On the other hand, it lacks capabilities for ambiguity resolution, clarification of misrecognized sentences and confirmation of sentences with low recognition confidence.

This paper focuses on the dialogue manager now being developed to address these limitations. It uses the Information State (IS) approach to dialogue systems [12]. The approach allows for mixed initiative dialogs. It also supports the use of pronouns and

Quinderé M., Seabra Lopes L. and J. S. Teixeira A. (2007).
A Dialogue Manager for an Intelligent Mobile Robot.
In *Proceedings of the 4th International Workshop on Natural Language Processing and Cognitive Science*, pages 191-200
DOI: 10.5220/0002431901910200

generates clarification/confirmation questions when the Automatic Speech Recognizer (ASR) confidence is low and/or the sentences are ungrammatical. Finally, it is able to produce comments on the information just acquired and it is also able to give informative answers

The paper is structured as follows. Section 2 presents different types of dialogue systems. Section 3 describes the developed dialogue manager. Section 4 presents an evaluation. Section 5 concludes the paper with reference to future work.

## 2 Dialogue Systems

Dialogue Systems can be divided into the following types [4]:

1. Finite State Systems – A finite state machine represents the dialogue, which means that every state transition has to be coded in the system. These transitions occur when the user gives the information the system was waiting for, usually a short phrase or even isolated words. Most Finite State Systems do not give much freedom to the user because the answers have to be given in a preset order. Besides, the user should not answer more than it was asked. As mentioned, our robot Carl also used until now a finite state approach. Although there are no constraint in the order of declarations/questions and the sentences can be quite complex, the approach is not flexible enough to address other problems, such as ambiguity, misrecognition and low recognition confidence.
2. Frame-Based Systems– These systems have frames with fields that need to be filled in order to allow a database query. The user is free to give as many answers as he wants, and the system is capable of handling that.
3. Advanced Systems – They are mixed-initiative, either the user or the system can have the control of the conversation.
   (a) Belief, Desire, Intention (BDI) Models – The majority of the dialogue systems that use BDI models [2] are plan based. For instance, if an agent needs an information, it can set a plan that includes asking something to get the missing information. Analogously, an agent that hears a question can infer why it was made.
   (b) Markov Decision Process (MDP) – in order to use them, a model that defines the behavior of the system is needed. For that, two methods can be used:
      – adjust the state number and policies to the minimum, build a system that explore the state space through random dialogues and then a model can be built from the created corpus.
      – develop a simulated user that interacts with the system a million times, then the system can learn from the corpus
   (c) Information State Systems – A dialogue context data structure, called *information state*, is kept and identifies what happened to the dialogue and also bases the dialogue manager decisions.

### 2.1 Some Dialogue Managers

**Florence** is a dialogue manager framework developed by AT&T [3]. This framework was made to support the development of a Spoken Dialogue System (SDS) with multiple dialogue strategies, instead of focusing on a single strategy, e.g. call routing or plan-based interaction.

Each dialogue can be guided by a different strategy, such as an Augmented Transition Network (ATN) strategy, which is a finite state machine extension, a clarification strategy or a rule-based strategy. The usual is ATN, which acts on the input and on the local context to control the interaction flow.

**ASIMO** is a conversational service robot. Nakano et al [7] propose a two-layer model for the behavior and dialogue planning in robots of that kind. They named their module MEBDP (*Multi-Expert-based Behavior and Dialogue Planning*), which is divided in: upper layer – a task planning layer responsible for decomposing a task into subtasks; lower layer – an expert action selection layer that performs the subtasks using experts. There are four types of experts: request understanding, information providing, physical action planning and information obtaining dialogue experts

**Jijo-2** is an office service robot that is able to communicate and learn about its environment [1]. The dialogue is handled by a finite state machine and so the robot's answer depends on its state. There are five kinds of tasks: consult database, database update, person identification, navigation and people calling.

Authors point out that the dialogue system's main problem is that the design of the state transition network is ad hoc and very task dependent. Another problem is the informal semantic representation of utterances, which makes it hard to extend the system to other tasks.

## 3 Description of the Dialogue Manager Developed

### 3.1 Requirements

The robot Carl should be able to carry on a dialogue like the one shown in Table 1. In order to do that, Carl's dialogue manager should have at least the following characteristics:

- Reactive – it should react appropriately to a user utterance
- Dialogue context – for reference resolution, for pronoun use and for selecting the most suitable action
- Reference resolution – so far used for understanding pronouns
- Ellipse – for understanding incomplete sentences
- Pronoun clarification – when the reference resolution is not able to resolve one or more pronouns
- Pronoun use – when speaking, it should use as much pronouns as it can to make the dialogue smooth

– Clarification – when confidence level of the ASR is low
– Comment – when receiving new information, it should make comments if possible
– Informative – when answering, it should be as much informative as it can, avoiding just “no” answers
– Grounding – system should make clear that it understood the user[1]
– Mixed initiative – both the user and the system should be able to have the initiative of the dialogue

**Table 1.** Fictitious dialogue with robot Carl.

| Speaker | Utterance | Requirement |
|---|---|---|
| User: | Hi! | dialogue context update |
| Carl: | Hi! I’m the robot Carl. | reactive |
| Carl: | What’s your name? | initiative |
| User: | Jim. | dialogue context update, ellipse |
| Carl: | What can I do for you? | give initiative |
| User: | Is he in the institute? | dialogue context update |
| Carl: | Who do you mean? | reactive, pronoun clarification |
| User: | Peter. | dialogue context update, ellipse |
| Carl: | No, he is in the department. | reactive, informative, dialogue context use, pronoun use |
| User: | Does he like you? | reference resolution, dialogue context use/update |
| Carl: | Yes. | reactive, informative |
| User: | Mary lives in Spain. | dialogue context update |
| Carl: | Did you say that Mary lives in Spain? | reactive, clarification |
| User: | Yes. | dialogue context update |
| Carl: | Really? Daniel lives in Spain too! | reactive, comment |

### 3.2 Information State Approach

From all the types of dialogue systems shown, an Information State (IS) approach was chosen to be used on the dialogue manager for being advanced, flexible, and able to address all the requirements mentioned before.

The developed dialogue manager is composed by:

– information state description – identification of the IS components
– events – external occurrences that evoke an information state update
– IS update rules – define when and how to update the information state
– action selection – defines which action to perform next
– control module – controls which update rules are applied and selects the next action

Carl’s software architecture is composed of following agents: Graphical and Touch Interface (GTI), Automatic Speech Recognizer(ASR), Natural Language Understanding (NLU), Natural Language Generation (NLG), Synthesis, Navigation and Manager.

Figure 1 shows an overview of the dialogue manager. One can see that it communicates directly to the agents: Navigation, NLG, ASR and GTI, as well as the KAM module. The IS includes fields to handle the NLU input, task managent, dialogue initiative and turn, user information, referenced objects (referents), questions performed by the system, events and robot control.

[1] currently, grounding is done by explicit display of ASR output on the screen

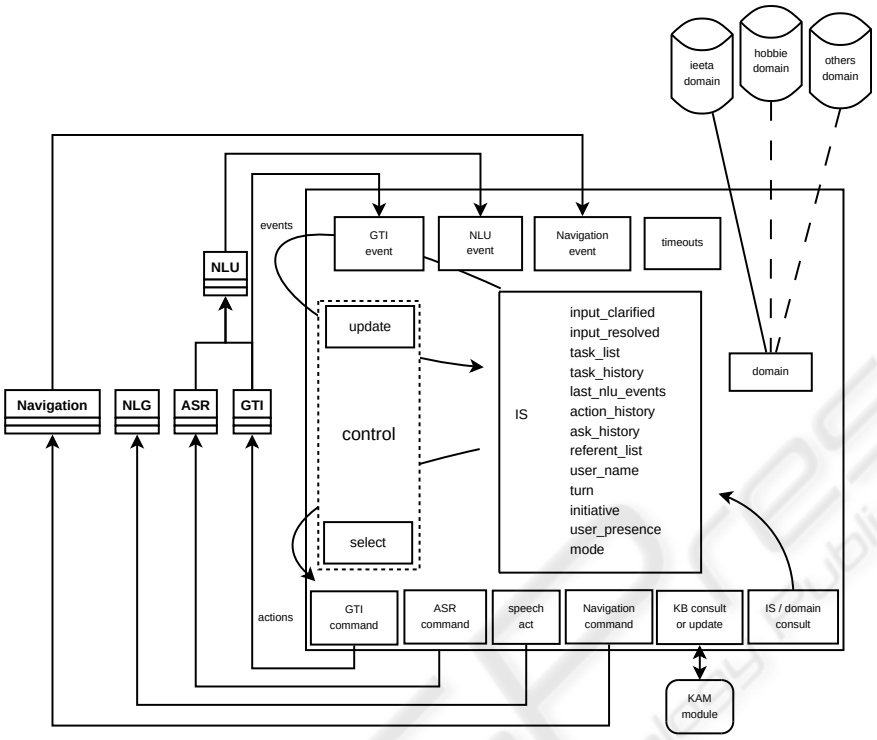


**Fig. 1.** Dialogue manager design overview.

Timeouts and events generated by the agents GTI, NLU and Navigation allow IS updates. The control module applies update rules (which can add tasks to the task list) and then select the next action to take (based on the current task list). Therefore, the behavior of the system is: receive events, update information state, select action and act.

Figure 2 shows the IS update flow of the NLU event. If the ASR confidence level is too low (less than 30%), the information is rejected and the task *reject* handles it. The semantic extraction can be shallow – performed by *Tilburg Memory Based Learner* (TiMBL) or deep – performed by *LCFlex* (details in [8]). If it is a *TiMBL* analysis and the ASR confidence is above 30%, a *clarify_timbl* task is added.

If it is a *LCFlex* analysis and the ASR confidence is between 30% and 70%, a *clarify* task is added so the information can go to the field *input_clarified*. If the confidence is above 70%, the information goes directly.

When there is a valid information on the field *input_clarified*, pronoun resolution is evoked. If all pronouns can be resolved, the information is put on the field *input_resolved*. If there is no referent for a pronoun or the system has two good choices, the *clarify_pronouns* task is added.

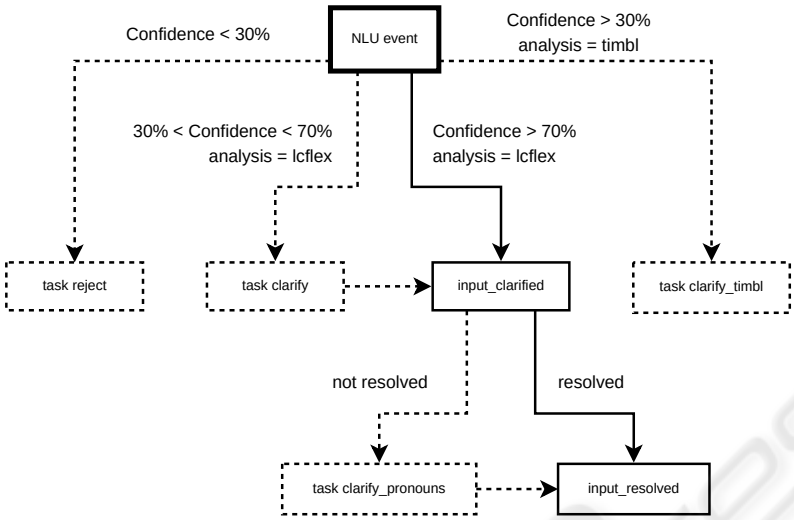


**Fig. 2.** Update flow of NLU event.

### 3.3 Dialogue Manager Control Algorithm

The dialogue manager control is performed by the algorithm 1. After receiving the events, the procedure *perform_state_update()* is called, which applies all the update rules that have the conditions satisfied. Note that the actions are performed while the system has the turn and, of course, actually has an action to perform. The IS field *turn* is updated when the user or the system perform a speech act.

Since tasks in task_list are sorted by priority, the control algorithm uses the first one to select the next action by calling the procedure *perform_action_selection(Task)*. The procedure *act(NextAction)* executes the action and updates the history.

One should note that the execution is finalized when the *program_state* field has a *stop* value.

The rule *updateReferentList* (1) is an example of an update rule. It is applied whenever there is a valid value in the IS field *input_resolved*. Its main effect is to update the IS field *referent_list* by calling the procedure *update_referent_list* with the current referent list and the semantics just received.

$$\begin{aligned} &\text{Rule} : \text{updateReferentList} \\ &\text{Conditions} : \{\ \text{valid\_IS\_value(input\_resolved)} \\ &\text{Effects} : \begin{cases} \text{InputValue} \leftarrow \text{get\_IS\_value(input\_resolved)} \\ \text{ReferentListValue} \leftarrow \text{get\_IS\_value(referent\_list)} \\ \text{NewReferentList} \leftarrow \text{update\_referent\_list(ReferentListValue, InputValue)} \\ \text{set\_IS\_value(referent\_list, NewReferentList)} \end{cases} \end{aligned} \tag{1}$$

The procedure *update_referent_list* is a simplified version of the algorithm described in [5] because the robot is designed to handle sentences much more simpler than the

**Algorithm 1**: control().

```
begin
    repeat
        receive_events()
        perform_state_update()
        repeat
            Task ← get_IS_list_head(task_list)
            NextAction ← perform_action_selection(Task)
            if NextAction ≠ null then
                act(NextAction)
                TurnValue ← get_IS_value(turn)
        until TurnValue = user ∨ NextAction = null
        ProgramStateValue ← get_IS_value(program_state)
    until ProgramStateValue = stop
end
```

ones addressed by the original algorithm. The main difference is the use of less salience factor types and filters. To support the reference resolution, a list of the referents mentioned in the dialogue are updated by the algorithm 2.

**Algorithm 2**: update_referent_list(ReferentList, Semantics).

```
begin
    foreach referent[i] in ReferentList do
        if referent[i].age = 3 then
            delete(ReferentList, referent[i])
        else
            referent[i].age = referent[i].age + 1
            referent[i].salience = referent[i].salience/2
    TempList ← extract_referents(Semantics)
    NewReferentList ← merge(ReferentList, TempList)
    NewReferentList ← sort_by(NewReferentList, salience)
    return NewReferentList
end
```

Each referent has a salience value associated, which is used to sort the list. These values are reduced to the half every time a new sentence is evaluated. This is to give priority to the recent referents. Other detail is that the list only keeps the referent mentioned in the last four sentences.

As a plan example, one can see the *store_info* plan (2). Basically, it stores information acquired from the user by calling the procedure *kb_update* of the KAM module. Since this module support contradictory information (details in [11]), it needs the user name to associate to every information it keeps, so the task *get_user_name* is the first step in this plan. After the information is stored, another procedure from the KAM

module is called, *kb_comment_info*. This one tries to generate a comment based on the semantics given and on the current state of the knowledge base. If it succeeds, *Status* is set to *ok*, otherwise, *not_ok*. If we do have a comment, the plan is to use pronouns if we can (by calling *replace_names_by_pronouns*) and then send a message to NLG agent with the semantics of the comment. Otherwise, the message is just an acknowledgement that the information was stored.

There is one task that generates questions to the user based on the current knowledge acquired. This task is added by a IS update rule when the system has the initiative on the dialogue.

```
Plan : store_info(+RecConf, +Semantics)
Operations : {
    task(get_user_name(UserName)),
    action(kb_update(UserName, RecConf, Semantics)),
    action(kb_comment_info(Semantics, Comment, CommentConfidence, Status)),
    if_then_else(
        Status = ok,
        [action(replace_names_by_pronouns(UserName, ReferentList, Comment, PronounComment)),
        oaa_action(nlg(comment, release-turn, _, _ , PronounComment, CommentConfidence))],
        [oaa_action(nlg(ack, release-turn, _, _ , _, _))]
    )
}                                                                                    (2)
```

## 4 First Results

A scenario was built with the purpose of checking if a dialogue like the one showed in Table 1 could really be carried on by the dialogue manager. In that dialogue, the robot showed that it already had acquired some knowledge, namely that: 1) Peter is in the department, 2) Peter likes Carl, 3) Daniel lives in Spain. So for the test, these facts were previously added to the knowledge base using the KAM module.

As the dialogue manager was just developed and is not fully integrated in the robot system yet, the output of the spoken language understanding (SLU) was simulated in this test. For each user utterance in the target dialogue a semantic representation was manually created. An ASR confidence of 75% was used, as well as a *lcflex* analysis label. Except on the “Mary lives in Spain” representation, in which a 50% confidence was used in order to force a clarification case.

Figure 3 shows the messages exchanged by the “simulated SLU”, the dialogue manager and the NLG. One can see that the dialogue was fully accomplished. In order to improve the comprehension of those messages, the target dialogue is shown again in Table 2, but this time with the respective agent and speech acts associated to each utterance.

## 5 Conclusion and Future Work

The development of a dialogue manager for the intelligent mobile robot Carl was presented. It uses the Information State (IS) approach to dialogue systems and it is based

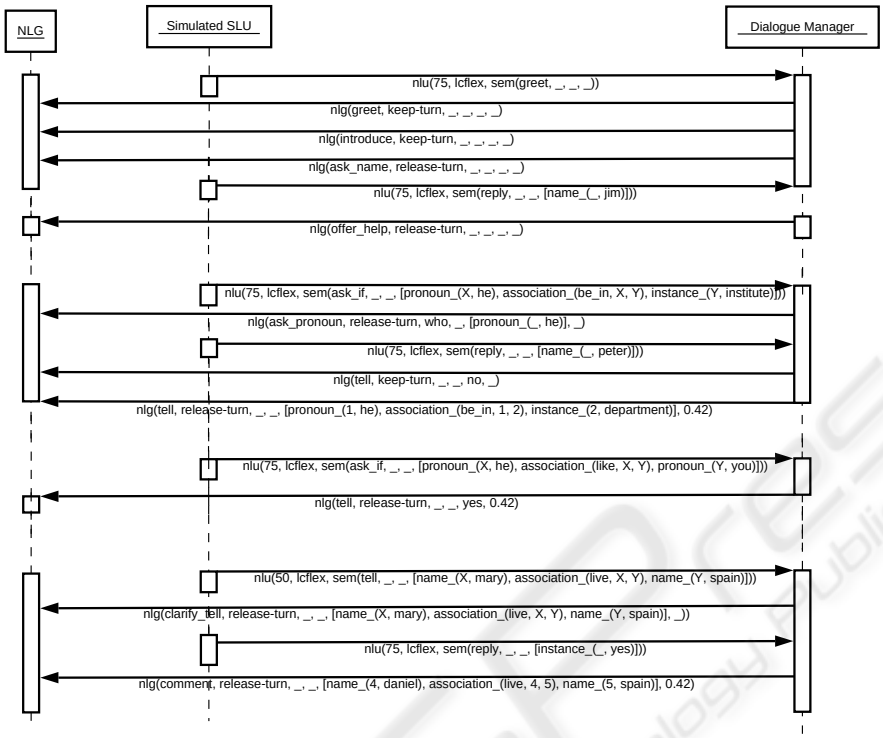


**Fig. 3.** Dialogue Manager Test.

**Table 2.** Fictitious dialogue with robot Carl.

| Speaker | Utterance | Agent | Speech Act |
|---|---|---|---|
| User0: | Hi! | NLU | greet |
| Carl0: | Hi! I'm the robot Carl. | NLG | greet, introduce |
| Carl1: | What's your name? | NLG | ask_name |
| User1: | Jim. | NLU | tell |
| Carl2: | What can I do for you? | NLG | offer_help |
| User2: | Is he in the institute? | NLU | ask_if |
| Carl3: | Who do you mean? | NLG | ask_pronoun |
| User3: | Peter. | NLU | tell |
| Carl4: | No, he is in the department. | NLG | tell, tell |
| User4: | Does he like you? | NLU | ask |
| Carl5: | Yes. | NLG | tell |
| User5: | Mary lives in Spain. | NLU | tell |
| Carl6: | Did you say that Mary lives in Spain? | NLG | clarify_tell |
| User6: | Yes. | NLU | tell |
| Carl7: | Really? Daniel lives in Spain too! | NLG | comment |

on a Knowledge Acquisition and Management (KAM) module that integrates information obtained from various interlocutors, even if they are contradictory. This mixed-initiative dialogue manager handles pronoun resolution, it is capable of performing different kinds of clarification questions and to comment information based on the current knowledge acquired.

Although the dialogue manager is not fully integrated in the robot system yet, a preliminary evaluation has shown that it is indeed capable to carry out complex dialogues. It should also be mentioned that it addresses all the requirements of a dialogue manager to be used on the robot Carl.

Current work includes the adjustment of the NLU and NLG agents to support the speech acts introduced by the new dialogue manager, which will allow the integration of the dialogue manager itself on the robot system.

## References


1. H. Asoh and et. al. Jijo-2: an office robot that communicates and learns. *IEEE Intell. Systems*, 16(5):46–55, 2001.
2. Michael E. Bratman, David Israel, and Martha Pollack. Plans and resource-bounded practical reasoning. In Robert Cummins and John L. Pollock, editors, *Philosophy and AI: Essays at the Interface*, pages 1–22. The MIT Press, Cambridge, Massachusetts, 1991.
3. Giuseppe Di Fabbrizio and Charles Lewis. Florence: a dialogue manager framework for spoken dialogue systems. In *ICSLP 2004, 8th International Conference on Spoken Language Processing*, Jeju, Jeju Island, Korea, October 4-8 2004.
4. D. Jurafsky and J. H. Martin. *Speech and Language Processing*. to be published, 2nd edition, 2007.
5. S. Lappin and H. J. Leass. An algorithm for pronominal anaphora resolution. *Comp Ling.*, 20:535–561, 1994.
6. D. Martin, A. Cheyer, and D. Moran. The Open Agent Architecture: a Framework for Building Distributed Software Systems. *App. Artif. Intelligence*, 13:91–128, 1999.
7. M. Nakano, N. Kanda, and et al. A Two-Layer Model for Behavior and Dialogue Planning in Conversational Service Robots. In *IROS*, pages 1542–1548, 2005.
8. M. Rodrigues, A. Teixeira, and L. Seabra Lopes. An Hybrid Approach for Spoken Natural Language Understanding Applied to a Mobile Intelligent Robot. In *NLUCS*, pages 145–150, Porto, Portugal, 2004.
9. L. Seabra Lopes. Carl: from situated activity to language level interaction and learning. In *IROS*, volume 1, pages 890 – 896, Lausanne, Switzerland, 2002.
10. L. Seabra Lopes and J. H. Connell. Guest editors' introduction: Semisentient robots– routes to integrated intelligence. *IEEE Intelligent Systems*, 16(5):10–14, 2001.
11. L. Seabra Lopes, A. Teixeira, and M. Quinderé. A Knowledge Representation and Reasoning Module for a Dialog System in a Mobile Robot. In *NLUCS*, pages 172 – 177, Miami, USA, 2005.
12. D. Traum and S. Larsson. The Information State Approach to Dialogue Management. In Jan van Kuppevelt and Ronnie Smith, editor, *Current and New Directions in Discourse and Dialogue*. Kluwer, 2003.