



# An Information State Based Dialogue Manager for a Mobile Robot

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

IEETA, Departamento de Electrónica, Telecomunicações e Informática  
 Universidade de Aveiro, 3810-193 Aveiro, Portugal

{marcelo, lsl, ajst}@ieeta.pt

## Abstract

The paper focuses on an Information State (IS) based dialogue manager developed for Carl, an intelligent mobile robot. It uses a Knowledge Acquisition and Management (KAM) module that integrates information obtained from various interlocutors. This mixed-initiative dialogue manager (DM) handles pronoun resolution, is capable of performing different kinds of clarification/confirmation questions and generates observations based on the current knowledge acquired. An evaluation of the DM on knowledge acquisition tasks is shown.

**Index Terms:** human-robot interaction, natural language understanding, dialogue system, knowledge representation, information state system

## 1. Introduction

“Integrated Intelligence” identifies an approach to building intelligent artificial agents in which the integration of key aspects, such as linguistic communication, reasoning, reactivity and learning is considered [8]. Personal robots, service robots capable of performing useful work in close cooperation/interaction with humans, are one example of these agents.

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 the robot prototype Carl was developed [7]. The software architecture of Carl, which is based on the Open Agent Architecture (OAA) [4], 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. State transitions specify which actions should be taken 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 [9] integrates information obtained from different interlocutors, even if they are contradictory, and provides replies to received questions. The dialog management approach used until now had several limitations. It followed a mostly user-initiative strategy and it lacked capabilities for ambiguity resolution, clarification of misrecognized sentences and confirmation of sentences with low recognition confidence.

This paper focuses on the dialogue manager developed to address these limitations. It uses the Information State (IS) approach to dialogue systems [10]. It supports mixed initiative dialogues, the use of pronouns and 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 observations related

to the information just acquired and it is also able to give informative answers

Others approaches have been used. ASIMO is a conversational service robot. Nakano et al [5] 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, hence 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.

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

## 2. Dialogue Manager

### 2.1. 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:

1. Reactive – it should react appropriately to an utterance
2. Dialogue context – for reference resolution, for pronoun use and for selecting the most suitable action
3. Reference resolution – so far, for understanding pronouns
4. Ellipse – for understanding incomplete sentences
5. Pronoun clarification – when the reference resolution is not able to resolve one or more pronouns
6. Pronoun use – when speaking, it should use as much pronouns as it can to make the dialogue smooth
7. Confirmation – when confidence level of the ASR is low
8. Observation – when receiving new information, it should make observations if it is possible
9. Informative – when answering, it should be as much informative as it can, avoiding just “no” answers

10. Grounding – system should make clear that it understood the user<sup>1</sup>
11. 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!	2
Carl:	Hi! I'm the robot Carl.	1
Carl:	What's your name?	11
User:	Jim.	2, 4
Carl:	What can I do for you?	11
User:	Is Paul in IEETA?	2
Carl:	No, he is in the department.	1, 9, 2, 6
User:	Tony lives in Spain.	2
Carl:	Did you say that Tony lives in Spain?	1, 7
User:	Yes.	2
Carl:	Really? Daniel lives in Spain too!	1, 8
User:	Does he like you?	3, 2,
Carl:	Who do you mean?	1, 5
User:	Tony.	2, 4
Carl:	Yes.	1, 9
User:	Bye.	2
Carl:	Goodbye.	1

## 2.2. Information State Approach

Dialogue Systems can be divided into [2]: Finite State Systems, Frame-Based Systems and Advanced Systems, which includes Belief, Desire, Intention (BDI) Models, Markov Decision Process (MDP) and Information State Systems.

From all these types, 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 has the following main components:

- information state (IS) – identification of the 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 management, dialogue initiative and turn, user information, referenced objects (referents), questions performed by the system, events and robot control.

<sup>1</sup>currently, grounding is done by explicit display of ASR output on the screen

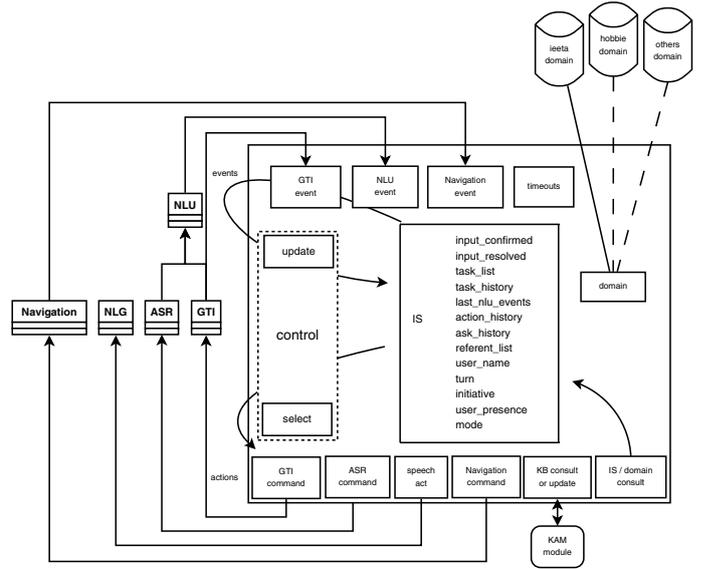


Figure 1: Dialogue manager design overview

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 [6]). If it is a shallow analysis and the ASR confidence is above 30%, a *clarify\_shallow* task is added.

If it is a deep analysis and the ASR confidence is between 30% and 60%, a *confirmation* task is added so the information can go to the field *input\_confirmed*. If the confidence is above 60%, the information goes directly.

When there is a valid information on the field *input\_confirmed*, 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.

## 2.3. 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

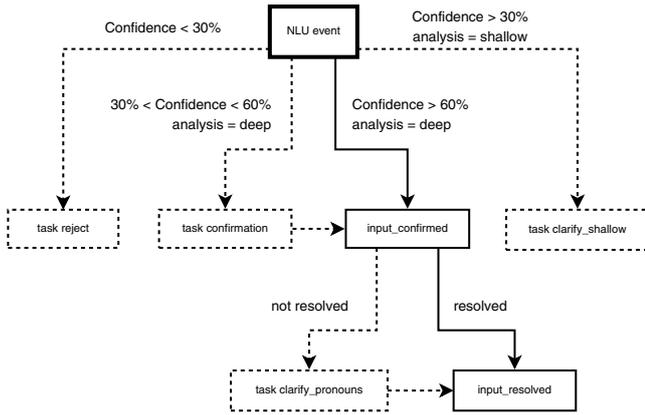


Figure 2: Update flow after a NLU event

**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
    until ProgramStateValue ← get_IS_value(program_state)
  until ProgramStateValue = stop
end

```

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. It uses a simplified version of the algorithm described in [3].

Rule : updateReferentList  
 Conditions : { valid\_IS\_value(input\_resolved)

Effects : {  $\left\{ \begin{array}{l} \text{Input} \leftarrow \text{get\_IS\_value}(\text{input\_resolved}) \\ \text{ListValue} \leftarrow \text{get\_IS\_value}(\text{referent\_list}) \\ \text{NewList} \leftarrow \text{update\_referent\_list}(\text{ListValue}, \text{Input}) \\ \text{set\_IS\_value}(\text{referent\_list}, \text{NewList}) \end{array} \right.$  (1)

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 supports contradictory information (details in [9]), 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\_observation*. This one tries to generate an observation 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 an observation, the plan is to use pronouns

if we can (by calling *use\_pronouns*) and then send a message to NLG agent with the semantics of the observation. 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, +RefList, +Sem)

Operations :  $\left\{ \begin{array}{l} \text{task}(\text{get\_user\_name}(\text{User})), \\ \text{action}(\text{kb\_update}(\text{User}, \text{RecConf}, \text{Sem})), \\ \text{action}(\text{kb\_observation}(\text{Sem}, \text{Obs}, \text{ObsConf}, \text{Status})), \\ \text{if\_then\_else} \\ \quad \text{Status} = \text{ok}, \\ \quad [\text{action}(\text{use\_pronouns}(\text{User}, \text{RefList}, \text{Obs}, \text{ProObs})), \\ \quad \text{oaa\_action}(\text{nlg}(\text{Obs}, \text{ProObs}, \text{ObsConf}))], \\ \quad [\text{oaa\_action}(\text{nlg}(\text{ack}))] \end{array} \right.$  (2)

**3. Experimental Results**

Several experiments were carried out to evaluate the described system. In the first experiment, the spoken language understanding (SLU) capabilities, i.e. the ASR/NLU integration, is evaluated. In this experiment, one speaker read aloud 41 sentences, repeating 3 times each one, in a total of 123 sentences with 312 relations. As ASR system, Nuance 8.0 with trigrams language model was used. This is the experiment reported in [9], now repeated to compare the current system with the previous one. Table 2 shows the results.

Two performance measures are used. Precision is the number of correct relations extracted by the system divided by the total number of relations extracted (= *Right/(Right+Wrong)*). Recall is the number of correct relations extracted divided by the total number of correct relations that could have been extracted (= *Right/Target*). The system had a Recall of 88.2% and a Precision of 90.7%, which is better than the previous results of 84.3% and 80.9%, respectively.

One of the sentences was not recognized by the ASR, probably because the experiment was not carried out by a native English speaker. Otherwise, the results could have been even better. Furthermore, the user did not correct minor ASR errors, such as confusion between articles “the” and “a”. This has increased the number of wrong relations in sentences in which this implied a different semantic.

Table 2: Semantic Analysis Results.

Relation	Target	Right	Wrong	Precision	Recall
Association	87	79	10	88.8 %	90.8%
Composition	21	12	1	92.3%	57.1%
Name	93	89	2	97.8%	95.7%
Instance	39	38	25	60.3%	97.4%
Subtype	21	21	0	100.0%	100.0%
Function	51	44	0	100.0%	86.3%
Total	312	283	38	88.2%	90.7%

The second part of the first experiment was carried out to test the performance of the integrated SLU/DM/KAM system. As in [9], an “Utopian” state of the knowledge base was created, using the correct semantics of the same 41 sentences of the previous test. Then, 41 questions based on the original phrases were made. Of course the system could respond to all of them, but since the confidence of the answer depends on the number

of statements that supports it, the average confidence is considerably lower than 100%.

To finish the first experiment, we have made the same 41 questions to the base that actually was produced after the first recognizer test. Table 3 shows the results of the comparison between reality and utopia. If we take the confidence result of reality (37.3%) and divide by the respective result of utopia (45.5%), we have a good measure of the efficiency of the system (82.1%). Performance numbers (97.6%) indicate the percentage of questions that could be answered. Again, the numbers are better than previous system, 67.2% and 90.2%, respectively <sup>2</sup>.

Table 3: *SLU/DM/KAM results - utopia vs reality.*

Data	Questions	Answers	Performance	Confidence
Utopia	41	41	100.0%	45.5%
Reality	41	40	97.6%	37.3%
Efficiency			97.6%	82.1%

A second experiment was made to analyze the internal update flow after a NLU event and to test the pronoun resolution capability of the dialogue manager. For this purpose, 10 pairs of sentences were selected. Half of the first sentences of the pairs contain two person/place names and the others contain a single name. The second sentence of a pair always contains a single pronoun referring to a name in the first one. The sentences were repeated until they reach 3 times the *input\_confirmed* field.

Table 4 shows the flow of the NLU event to the IS field *input\_confirmed*. In 84.4% of the cases, the information goes through the task confirmation, which confirms 71.6% of them (60.4% of total). Only in 2.1% of the events, it goes directly to the field. This reveals it is been used conservative thresholds in the NLU flow. The reason is to protect the knowledge base from misrecognized sentences.

Table 4: *Update Flow after a NLU event.*

Flow	Number	Percentage
Direct	2	2.1%
Task confirmation - yes	58	60.4%
Task confirmation - no	23	24.0%
Task reject	9	9.4%
Task clarify_shallow	4	4.2%

From the *input\_confirmed* to the *input\_resolved* field, 91.7% (55 out of 60) of the flow goes directly and the rest goes through the task *clarify\_pronouns*. The ASR on this experiment had a word error rate of  $11.26\% \pm 27.82\%$ .

Since the focus was on the pronoun resolution, the sentences were simpler than on the previous experiment and the semantic extraction was perfect for all the relations included. Table 5 shows the analysis of the integrated SLU/DM/KAM system on the second experiment. One can see that it has a high performance and confidence efficiency.

Despite the small size of the test, the results show that new dialogue manager is able to handle pronoun resolution and has improved the performance of system on knowledge acquisition.

#### 4. Conclusion and Future Work

The paper presented an IS based dialogue manager developed for Carl, an intelligent mobile robot. It uses a Knowledge Ac-

<sup>2</sup>part of the improvement is due to a bug correction, since with that bug corrected, the performance would have been 92.7%

Table 5: *SLU/DM/KAM results - second experiment.*

Data	Questions	Answers	Performance	Confidence
Utopia	20	20	100.0%	48.9%
Reality	20	20	100.0%	44.1%
Efficiency			100.0%	90.3%

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

The experimental results presented show that the new DM has improved the performance of Carl on semantic analysis and also on knowledge acquisition. When compared to the previous system, it has better precision and recall values, 9% and 7.5% higher, respectively. On knowledge acquisition, it was 22.1% more efficient.

The results also demonstrated the new DM is able to handle pronoun resolution. The system had a perfect semantic extraction and a performance of 100% on the second integrated SLU/DM/KAM experiment.

Future work includes a deeper evaluation of the dialogue system that addresses all the requirements mentioned on section 2.

#### 5. 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] D. Jurafsky and J. H. Martin. *Speech and Language Processing*. to be published, 2nd edition, 2007.
- [3] S. Lappin and H. J. Leass. An algorithm for pronominal anaphora resolution. *Comp Ling.*, 20:535–561, 1994.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] L. Seabra Lopes. Carl: from situated activity to language level interaction and learning. In *IROS*, volume 1, pages 890 – 896, Lausanne, Switzerland, 2002.
- [8] L. Seabra Lopes and J. H. Connell. Guest editors' introduction: Semisient robots– routes to integrated intelligence. *IEEE Intelligent Systems*, 16(5):10–14, 2001.
- [9] L. Seabra Lopes, A. Teixeira, M. Quinderé, and M. Rodrigues. From Robust Spoken Language Understanding to Knowledge Acquisition and Management. In *Interspeech*, pages 3469 – 3472, Lisbon, Portugal, 2005.
- [10] 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.