Improvement in Service Robot’s Interaction through Case Based Reasoning

Catalina Roncancio, Jose L. Rodríguez, Eduardo Zalama, Jaime Gómez G-B

Abstract—In order to enable tight interaction and close collaboration between human and service robots it is necessary that robots imitate the cognitive mechanisms of the brain and their role in competent activity control. That goal can only be achieved if robots are capable of observing their environment, reasoning about it, and autonomously deciding the most appropriate action in a given situation. We present a possible application in this broad area: “Sacarino” is a service robot designed to work as bellboy in a hotel environment as part of the ROSETEL project. In order to tackle the challenge of learning by interaction we define: (i) the scenarios in which our agent have to work, (ii) its prior knowledge in an episodic memory module to be used in a prime dialog interaction, and (iii) the proper architecture that will support intelligent behaviour suitable for the control of our robot. We will use frameworks like JCOLOBRI (Cases and Ontology Libraries Integration for Building Reasoning Infrastructures) to define the agent’s memory structure, taking a deeper look into the episodic memory of the system.

I. INTRODUCTION

Our current research interest is the development of a service robot that will safely coexist with humans, interactively communicate with them and successfully cooperate [1]. For this reason it is necessary that robots imitate the cognitive mechanisms of the brain and their role in competent activity control. The goal of our work is to provide reliable and highly integrated robotic platform which on the one hand allow the implementation and tests of various research and on the other hand the realization of service tasks in a hotel scenario. Indeed, this is a paradigmatic context of application combining both significant social and economic impact and challenging but realistic research objectives. Furthermore our work is aimed to achieve the general objective of social robotics research which is enable people who have never met a real robot to operate and interact with it.

Related work in the service robots interaction scenario has been carried out by [2],[3], the RoboCup@home competition\(^1\) and DESIRE [4] project just to name a few. In these projects we can see the efforts to create robots capable of performing various tasks in households or public domain, reliable enough for every-day use. These projects are based on many expert technologies from different robotic domains and different developers. “DESIRE” project proposes software engineering methods and tools that support integration and a new web portal for the distributed development and remote testing of service robots, integrating all software components on one common hardware platform in which more than 14 partners from universities, research institutes and industry supply components to that platform.

Inspired in works like those mentioned before and in [5], we define a framework for modeling System of Systems (SoS) architecting [6]. These SoS attempt to integrate several independent complex systems into meta-architectures. From many potential systems with differing objectives, a set must be selected to construct the meta-architecture for SoS. The selection of the set depends on the requirements, functionalities and capabilities desired from the SoS to achieve the common mission. This requires the creation of a meta-architecture that consists of core components that remain unchanged for a given period as other components that evolved through time.

Systems could contain subarchitectures for motor control, vision, action, planning, language etc. Each subarchitecture contains a number of processing components which share information via a working memory. And the knowledge that can be used within a subarchitecture is defined by a set of ontologies for that module. Relationships (not necessarily equivalence) between the entries in the ontologies in different subarchitectures are defined by a set of general ontologies [7]. Sloman’s H-Cogaff cognitive architecture is utilized as a blueprint to design our architecture.

II. SCENARIO

We envisage our robot being useful in typical hotel scenario, in which the robot is expected to be able to interact with hotel guests, staff and environment, changing its known world through physical action and linguistic discourse, while performing the requested services and tasks. Knowledge to be of interest comprises topological representations of the hotel space, models about relevant objects and functional spaces, and about different users (guest and reception desk).

The latter aspect is of particular relevance because a robot is considered to be a servant or companion interacting with various members of the hotel.

Our bellboy robot will be designed and developed so that it is capable of providing the following services and tasks:

- Accompanying the guests to their rooms.
- Explaining the services available in the room and the hotel (meals, laundry, etc).
- Carrying food, drinks, equipment and newspapers to the rooms.

\(^1\)www.robocupathome.org

Catalina Roncancio and Jose Luis Rodríguez are with Robotics, Computer vision and Biomedical Engineering Department at CARTIF Foundation. Parque Tecnológico de Boecillo Parc 205, Valladolid, Spain; e-mail:(catron, josrod)@cartif.es.
Eduardo Zalama and Jaime Gómez García-Bermejo are with the ISA Dep. at the University of Valladolid, Valladolid, Spain; e-mail:(eduzal, jaigom)@eis.uva.es.
Talking with guests in defined contexts, taking care of orders, providing useful information (tourist, weather information, news, etc.)

And sending/receiving messages to/from the reception desk.

These services and tasks are going to be defined by the guest or hotel staff by natural language, as the main way of communication for people. This human–robot interaction is based on a requesting and response method. In this interaction the exchange of information lets our robot to continuously improve its knowledge, acquiring new information and storing the most relevant aspects as episodes. Modeling this knowledge as episodes allows us to fit it as cases in a CBR (Case Based Reasoning) system. The implementation will be done by a KI-CBR (Knowledge Intensive-CBR) System.

CBR works both as a design architecture and as a tool for adapting the system to new unseen phrases. The result is a dialog system that is able to deal with phrases that have not been stored beforehand in the system. The new knowledge is stored and the system performance improves through time.

All the concepts including tasks and services that the robot is able to manage has to be defined inside a hotel ontology. The ontology could be defined as a tool that describes and delimits a specific knowledge domain. Our hotel’s ontology has the concepts shown in Figure 1. These concepts allow the agent to provide information about: the hotel facilities (reception, restaurant, entertainment), nearest turistic places, public transportation, weather forecast and its self-information.

Moreover like in Soar, the bellboy robot’s knowledge could be represented in the architecture as a sequence of decisions through a problem space. Some concrete examples are:

This list incorporates many different kinds of knowledge that our model must include: knowledge about things in the world (K1) and knowledge about abstract ideas (K2), knowledge about physical actions (K7) and knowledge about mental actions (K3 and K5), even knowledge about how to use the other kinds of knowledge (K4 and K6).

### III. ROBOTIC PLATFORM

Our current work in robotics uses a custom robotic platform called “Sacarino” developed by CARTIF foundation as show in Figure 2. Sacarino is an indoor battery-powered wheeled robot that can move forward, turn in place, or turn while moving. It has capabilities of perception and understanding of natural language and Human body activity. For development, we have made a 3D environment simulation. Our simulation attempts to be a test-bed of our approach so we can specify the kinds of information the agent must be able to obtain and process, without initially specifying exactly how these components are actually implemented. When controlling the robot, an agent runs on a laptop that sits on the robot and interfaces through an Ethernet port.

<table>
<thead>
<tr>
<th>Table I</th>
<th>A SMALL PORTION OF THE KNOWLEDGE NEEDED TO MODEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>K1</td>
<td>Knowledge of the objects in the environement.</td>
</tr>
<tr>
<td></td>
<td>e.g. Lobby, front office, elevator, rooms, bar.</td>
</tr>
<tr>
<td>K2</td>
<td>Knowledge of abstract events and particular episodes.</td>
</tr>
<tr>
<td></td>
<td>e.g. What the user had mentioned in a previous interaction with the bellboy robot.</td>
</tr>
<tr>
<td>K3</td>
<td>Knowledge of the rules of the scenario.</td>
</tr>
<tr>
<td></td>
<td>e.g. Interaction with the guest.</td>
</tr>
<tr>
<td>K4</td>
<td>Knowledge of objectives.</td>
</tr>
<tr>
<td></td>
<td>e.g. Give information of the hotel’s services, leisure, mealtimes and transport.</td>
</tr>
<tr>
<td>K5</td>
<td>Knowledge of actions or methods for attaining objectives.</td>
</tr>
<tr>
<td></td>
<td>e.g. Chase, Evade, Surround, tell jokes.</td>
</tr>
<tr>
<td>K6</td>
<td>Knowledge of when to choose actions or methods.</td>
</tr>
<tr>
<td></td>
<td>e.g. If the user does not need more information, stop the dialog and go away.</td>
</tr>
<tr>
<td>K7</td>
<td>Knowledge of the component physical actions.</td>
</tr>
<tr>
<td></td>
<td>e.g. How to express him (facial expressions) while performing the interaction.</td>
</tr>
</tbody>
</table>

Fig. 1. Hotel Concepts

Fig. 2. The general system architecture

The robot should be able to deal with a hotel environment and the wide variety of objects and activities encountered in it. Therefore, the robot must be designed under a comprehensive view so that a wide range of tasks (and not only a particular task) can be performed.

### A. Agent architecture

At the cognitive level, Sloman’s H-Cogaff architecture model [8] is selected because this architecture is modular and flexible to model different sub-components of SoS. The
architecture (see Figure 2.) consist of perception, central processing and actuation components. The central processing (Figure 3) is also a layered architecture which consists of reactive layer for immediate response to environment states, deliberative layer for long term reasoning and a meta-layer to control the activities of the lower layers.

Fig. 3. Cognitive agent architecture

The proposed architecture was implemented in a previous project [9], a mechatronics head with social capabilities. As we see in figure 3 the cognitive architecture consist of:

- Reactive level. The information coming from the perception module is processed at this level. Also, the information coming from higher levels is here interfaced to the actuator components, through the movement control module and the sound and speech generation module. The former provides all actuator synchronization towards visual tracking, sound tracking and gesture production (angry, sad, and happy); the latter provides speech from text and other sound effects (music etc.).
- Deliberative level: At this level, external stimuli (from sensors) are processed, and motivational stimuli are generated upon the internal needs of the robot. All these stimuli are processed through learning neural networks in the behaviour and learning module, so that the currently active behaviours of the robot are chosen. These behaviours are fed to both the meta-layer level, and the movement control module within the reactive level, for gesture and tracking movement production.
- Meta-Layer level: High-level commands are here interfaced to the lower levels according to the activated behaviour, the sensorial information and internal variables such as the date, time and even information retrieved from Internet (such as weather information). The robot profile is defined at this level (a bellboy, in our case) for dialogue planning. Dialogue is then executed according to the user requests, and the external and motivational stimuli of the robot. Thus, an interactive, bidirectional dialogue with the surrounding humans is maintained in defined context environments.

B. Software Construction

Our prototype system has been constructed by two main components: (1) a server system which can put the different modules altogether under control by messaging function; (2) an agent system AIE-Agent that can interact with a voice, with speech recognition is implemented by Loquendo Speech SDK [10]. The server system can send an XML query to specified client.

In order to tackle real-world challenges, we program a virtual scenario of the hotel (see Figure. 4) as a test-bed method to test the real settings and let non-expert users interact with it as a first approach of our project.

Fig. 4. The participant interacting with the agent

The user is engaged in interaction with the robot by means of verbal dialog, joint spatial exploration, and gestural reference, to mention only some relevant abilities.

In the next section we describe the memory model to manage of the concepts previously defined to generate the prior knowledge that our service robot should have, experiences about the services and tasks it has to accomplish.

IV. MEMORY MODEL

The representation of knowledge recalls that knowledge itself is not built into an architecture, in that it can change across domains and over time. One common tradition distinguishes declarative from procedural representations. Declarative encodings of knowledge can be manipulated by cognitive mechanisms independent of their content [11]. In contrast, procedural formalisms encode knowledge about how to accomplish some task. In general, procedural representations let an agent apply knowledge efficiently, but typically in an inflexible manner.

Yet another distinction [12] involves whether stored knowledge supports a semantic memory of generic concepts, procedures, and the like, or whether it encodes an episodic memory of specific entities and events the agent has encountered in the environment. Most cognitive architectures focus on semantic memory, partly because this is a natural approach to obtaining the generalized behavior needed by
an intelligent agent, whereas an episodic memory seems well suited for retrieval of specific facts and occurrences. However, methods for analogical and case based reasoning can produce the effect of generalized behavior at retrieval time, so an architectures commitment to semantic or episodic memory does not, by itself, limit its capabilities. Neither must memory be restricted to one framework or the other.

The memory structure in our cognitive agent represent patterns and situations. To support efficient reasoning about the current situation, while at the same time supporting extremely large knowledge bases we take a deeper look into the episodic memory.

A. Episodic memory

To remember any cognitive activity, the architecture must store the cognitive structures generated during that activity, index them in memory, and retrieve them when needed. The resulting content is often referred to as episodic memories. Inspired by [13], in terms of its primitive memories, we will work on the long-term memory (LTM) -where the knowledge that exists independent of the current situation is held, and its three independent representations: Procedural, semantic, and episodic.

In contrast to semantic memory, which contains knowledge independent of when and where it was learned, episodic memory contains memories of what was experienced over time. Providing the ability to remember the context of past experiences as well as the temporal relationships between them [14]. An episode is retrieved by the deliberate creation of a cue, which is a partial specification of working memory in a special buffer. Once a cue is created, the best partial match is found (biased by recency and working memory activation) and retrieved into a separate working memory buffer (to avoid confusion between a memory and the current situation). The next episode can also be retrieved, providing the ability to replay an experience as a sequence of retrieved episodes. Although similar mechanisms have been studied in case-based reasoning, episodic memory is distinguished by the fact that it is task-independent and thus available for every problem, providing a memory of experience not available from other mechanisms. Episodic learning is so simple that it is often dismissed in AI as not worthy of study. Although simple, one has only to imagine what life is like for amnesiacs to appreciate its importance for general intelligence [15]. In Soar they have demonstrated that when episodic memory is embedded, it enables many advanced cognitive capabilities such as internal simulation and prediction, learning action models, and retrospective reasoning and learning.

A CBR system allows us to model the episodic memory by cases. These cases are formed through the components that both overall and particular ontology domain provides us. Although various authors have applied knowledge level analysis to CBR systems, the most relevant work is the CBR task structure that was developed in [16] influenced by the Components of Expertise Methodology [17]. At the highest level of generality, they describe the general CBR cycle in terms of four tasks: Retrieve the most similar case/s, Reuse its/their knowledge to solve the problem, Revise the proposed solution and Retain the experience. These four tasks involved in this continuous cycle are so called the 4R cycle (see Figure 5). The Central point is the case-base, which is a repository of completed cases, in other words the memory. When a new problem arises it must be codified in terms of the feature vector (or problem description) that is then the basis for retrieving similar cases from the case-base.

We can find various technologies and framework available on the market to model CBR systems like MyCBR [18]. Where one of the most well known framework is JCOLIBRI [19]. Also we use the software tool Protege [20], to build and manage ontologies needed to develop our system to model episodic memory of the cognitive engine. An ontology determines what is of interest in a domain and how information about it is structured.

<table>
<thead>
<tr>
<th>Target Case</th>
<th>Problem description</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Similar found</td>
<td>Solution</td>
<td></td>
</tr>
<tr>
<td>Case Base</td>
<td>Problem description</td>
<td>Solution</td>
</tr>
<tr>
<td>Retain Experience</td>
<td>Proposed solution</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 5. The CBR Process. (Adapted from Aamodt and Plaza [16])

Case Based Reasoning is based on the idea of memory rather than explicit models. It would also seem to fit closely with how humans often solve problems, that is by means of analogy. This concept can help human users to trust CBR systems and, potentially, to better interact with them [21].

The CBR approach is to build integrated systems that combine cases specific knowledge with knowledge models of the general domain. Even though any CBR system relies on a set of previous specific experiences, its reasoning power can be improved through the use of general knowledge about the domain. Our approach to CBR is to build integrated knowledge based systems (KBS) that combine case specific knowledge (our scenario) with models of general domain knowledge.

The core of the JCOLIBRI architecture is CBROnto, an ontology incorporating common CBR terminology and problem solving knowledge. CBROnto serves as a domain-independent framework to design KI-CBR systems. It is both
an ontology including general terminology related to CBR systems, and an ontology of CBR tasks and methods.

In this work we use CBROnto, as a task/method ontology [22]. CBROnto specifies a modelling framework to describe reusable CBR Problem Solving Methods based on the CBR tasks it solve and the knowledge requirements needed to apply them.

JCOLIBRI helps to design KI-CBR systems that combine specific cases with various knowledge types and reasoning methods. The major problem associated with the knowledge intensive approach to CBR is the so called knowledge acquisition bottleneck. Our approach to knowledge acquisition [23] is based on reusing knowledge from an ontology library to create complex, multirelational knowledge structures to support the CBR processes. As the next step, the KI-CBR system should be able to take advantage of the acquired domain knowledge. JCOLIBRI views KI-CBR systems as consisting of collaborating knowledge components, and distinguishes different types of knowledge [24]. Ontologies describe the structure and vocabulary of the Domain Knowledge that refers to the actual collection of statements about the domain. Tasks correspond to the goals that must be achieved. CBROnto captures CBR semantically important terms, includes CBR dependent but domain-independent terms, and aims to unify case specific and general domain knowledge representational needs. A useful way to describe problem solving behavior is in terms of the tasks to be solved, the goals to be achieved, the methods that will accomplish those tasks, and the domain knowledge that those methods need. A description along these lines is referred to as a knowledge level description.

V. KI-CBR APPROACH

In our approach to KI-CBR modeling we have used an ontology of the hotel environment. This ontology is built on several key concepts such as the hotel reception, hotel staff, hotel services and different kinds of information we can provide to the robot (weather, transport, leisure). In figure 6 we show an ontology built in our project concerning to the restaurant theme.

Once we have developed our ontology, the next step is to define the cases that were part of the CBR system. These cases are formed by the concepts and actions built in the the ontology of the hotel. We can use the CBR system as long-term memory when we want to use the knowledge of objectives, knowledge previously defined in Table 1, as k4, to provide the information requested by the guest. These cases contain the relevant information of the concepts on which requests information. When a guest requests information about restaurants, we provide minimal information for the CBR retrieve cycle. For example we have indicated to the system that we would like dinner at an Italian restaurant and not too expensive. Therefore we have introduced for the case that kind of specialty restaurant and the restaurant’s desired price. The system will search to find restaurants that best suit the preferences of the hotel guest. The agent provides the host response at a later stage and keep the case in its database to complete the cycle CBR.

VI. PRELIMINARY TESTS

The prototype will be tested in a hotel with one real scenarios:

scenario 1 :
- The robot shall meet the guests in the entrance and:
  - Greet them.
  - Carry their baggage.
  - Use the lift.
  - Navigate to the room accompanying the guest.
  - Establish a dialogue with the guests informing about hotel services and other information requested.

While interacting in this scenario the user could requests any information inside the hotel ontology. In this first experiment the guest make a request about restaurants in the area and specifically on Italian restaurant. The process beginning with that request makes the agent uses its KICBR system to search in memory any appropriate restaurant to the guest petitions. The answer provided by the agent is determined by their own knowledge, that knowledge falls into the own
TABLE II
SURVEY TEST RESULTS

<table>
<thead>
<tr>
<th>User Id.</th>
<th>Satisfaction</th>
<th>Usefulness</th>
<th>Response Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>80</td>
<td>88</td>
<td>85</td>
</tr>
<tr>
<td>User 2</td>
<td>77</td>
<td>85</td>
<td>85</td>
</tr>
<tr>
<td>User 3</td>
<td>84</td>
<td>92</td>
<td>86</td>
</tr>
<tr>
<td>User 4</td>
<td>85</td>
<td>91</td>
<td>85</td>
</tr>
<tr>
<td>User 5</td>
<td>88</td>
<td>93</td>
<td>86</td>
</tr>
<tr>
<td>User 6</td>
<td>81</td>
<td>88</td>
<td>84</td>
</tr>
<tr>
<td>User 7</td>
<td>82</td>
<td>88</td>
<td>88</td>
</tr>
<tr>
<td>User 8</td>
<td>82</td>
<td>90</td>
<td>88</td>
</tr>
<tr>
<td>User 9</td>
<td>84</td>
<td>93</td>
<td>87</td>
</tr>
<tr>
<td>User 10</td>
<td>87</td>
<td>93</td>
<td>88</td>
</tr>
<tr>
<td>User 11</td>
<td>83</td>
<td>94</td>
<td>88</td>
</tr>
<tr>
<td>User 12</td>
<td>84</td>
<td>91</td>
<td>85</td>
</tr>
</tbody>
</table>

ontology. The final result of the CBR cycle could contain information like:

- Name, contact and address.
- Closing Periods, days and opening hours.
- Average meal duration is giving as information for the customer if he/she has some time constraints.
- Specific occasions, atmospheres and entertainment. Particularities to help the customer to choose a restaurant according to his/her motivations.
- Dishes, meals and beverages. The type Of Cuisine can help to choose a restaurant according to the cuisine or some particular dishes, or some particular meals and their prices.
- Ratings. The notation of the restaurant by professionals or people.

Some preliminary results obtained based of a test survey show the user acceptance of the system and how the final user feel in the interaction process with the system. (Table II and figure 7)

VII. CONCLUSIONS

In this paper we show the integration of different concepts of cognitive architectures through the study of different frameworks available in the community. We make use of JCOLIBRI framework to establish our starting point for our memory structure. Ontology’s definition let us to establish the agent world’s rules and otherwise the agent behaviour. Finally we define the prior knowledge of our agent needed to produce a correct behaviour. This work is a first stage development of the ROSETEL project to test some of the typical scenario in which our agent should interact, evolve and learn.

The impressive advance of research and development in robotics over the past years has led to the development of humanoid robots that are rich in sensory and motor capabilities and hence provide a suitable framework for studying cognition. Currently, the different disciplines related to the development of cognitive humanoids have usually been explored independently, leading to significant results within each discipline. However, the big challenge is how we can fit the different pieces of results to achieve complete processing models and an integrative system architecture, and how to evaluate results at system level rather than focusing on the performance of component algorithms.

ACKNOWLEDGMENT

This research is based upon works partly supported by the Science and Innovation Spanish ministry (Pr. Nb.DPI2008-06738-C02-01/DPI ), jan-2009 to dec-2011 and CARTIF foundation.

REFERENCES


