robOCD: Robotic Order Cups Demo – An Interactive Domestic Service Robotics Demo

Stefan Schiffer, Tobias Baumgartner, Daniel Beck, Bahram Maleki-Fard, Tim Niemüller, Christoph Schwering, and Gerhard Lakemeyer

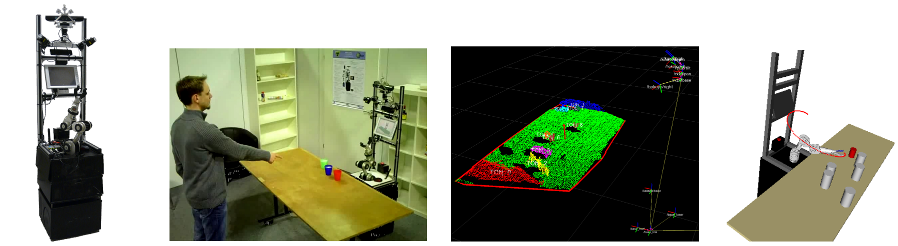
Knowledge-based Systems Group,
RWTH Aachen University, Aachen, Germany
(schiffer,gerhard)@kbsg.rwth-aachen.de

Abstract. This paper describes an interactive demonstration by the AllemaniACs' domestic service robot Caesar. In a home-like environment Caesar's task is to help setting the table. Besides basic capabilities of an autonomous mobile robot it uses methods for human-robot interaction and it also has a sophisticated high-level control that allows for decision-theoretic planning. We use this demo to illustrate the interplay of several modules of our robot control software in carrying out complex tasks. The overall system allows to perform robust reliable service robotics in domestic settings like in the RoboCup@Home league. Also, we show how our high-level programming language provides a powerful framework for agent behavior specification that can be beneficially deployed for service robotic applications. The system was showcased and evaluated repeatedly, most notably at a national RoboCup competition and at an international conference.

1 Introduction

In RoboCup, apart from making agents and robots play soccer, there are also competitions in rescue scenarios as well as in domestic service robotics settings. While in the soccer leagues quick decision making, cooperation, and teamplay are crucial, the challenge in the RoboCup@Home league is to build a robust robotic system that can safely operate in a human environment and that can interact with humans. As the complexity of the tasks to solve in domestic settings rises, so increases the benefit a robot has from using sophisticated means for decision-making and deliberation. The high-level control of Caesar is based on the language Readylog [1], a variant of the logic-based language Golog [2] which combines explicit agent programming as in imperative languages with the possibility to reasons about actions and their effects.

In this paper, we present a demo application of Readylog in a domestic setting to showcase its benefits and its applicability. After we briefly introduce our robot we sketch the domestic service robotics domain. Then we present the robotic order cups demo (robOCD) before we conclude.



(a) Caesar (b) User pointing at a cup (c) Scene perception (d) Manipulation

Fig. 1. Our robot Caesar, the robOCD scenario with a user giving instructions and perception and motion planning in simulation for manipulation in an extended setup.

2 The AllemaniACs Domestic Service Robot Caesar

An increasingly popular application domain for autonomous mobile robots is domestic service robotics (DSR) where robots perform assistive tasks in a domestic setting. A competition that focuses on these kinds of applications is RoboCup@Home [3]. Apart from the demands that are commonly put on autonomous mobile robots, a service robot in a domestic setting must meet additional requirements with respect to interactivity, robustness, and accessibility. What is more, it can be more helpful in complex assistive tasks if it also features a sophisticated high-level control.

Our robot Caesar, shown in Figure 1(a), is designed and was built to operate in human-populated environments in domestic scenarios. It should be helpful around the house, assisting elderly or disabled people with their daily activities. Caesar meets all the basic requirements put on an autonomous mobile robot, that is, it can navigate in its environment safelyand it can localize itself reliably with high accuracy in known environments. Further it is able to detect and recognize people and objects and it can also manipulate objects with its robotic arm. Of particular importance for the demo we discuss in this paper are its robust speech recognition [4] and a component for gesture recognition [5]. Those components are orchestrated in the Fawkes robot framework [6] to form a robust assistive robotic system.

Above all the low-level components mentioned so far and a mid-level for basic skills of the robot [7], Caesar has a logic-based high-level control that allows for deliberation and flexible decision-making. We use Readylog [1], a dialect of the robot programming and plan language Golog [2]. Golog is based on Reiter's version of the situation calculus [8] which is a sorted second-order language to reason about dynamic systems with actions and situations. Readylog features several extension to the original Golog language, most notably it allows for decision-theoretic planning in the spirit of DTGolog [9]. On Caesar we use an implementation of a Readylog interpreter in ECLiPSe-CLP,[1] a Prolog dialect.

[1] http://www.eclipseclp.org/

Algorithm 1: READYLOG program for the Order Cups Demonstration The terms p_1 to p_4 denote four positions on the table, while I_i and P_i are variables that hold the color of a cup at position i in the initial and the goal situation, respectively. $pos(C)$ returns the position of the cup with color C.

```
proc main,
  get_Initial_Order(I1, I2, I3, Init);        %% perceive initial order
  get_Goal_Order(P1, P2, P3, Goal);           %% inquire about goal order
  set(cup_pose(I1, I2, I3), (p1, p2, p3));    %% set initial cup positions
  sort_cups(P1, P2, P3, 4);                   %% start planner
endproc
proc sort_cups(P_1, P_2, P_3, H),
  solve(H, reward_cup(P_1, P_2, P_3),
    while(¬(p1 = pos(P_1) ∧ p_2 = pos(P_2) ∧ p_3 = pos(P_3))) do
      pickBest(cup, {red, green, blue},
        pickBest(to, {p_1, p_2, p_3, p_4}, move_cup(cup, pos(cup), to)))
    endwhile
  endsolve
endproc
```

3 Robotic Order Cups Demo

To demonstrate how the reasoning capabilities provided by READYLOG can be put to good use we now discuss a special helping task that CAESAR is able to perform: the *Order Cups Demo (OCD)*.[2] The robot's task in this demo is to help decorating a table. There are three differently colored cups (red, green, and blue) on the table. To complete the re-arranging they have to be put in a specific order. A human user is instructing the robot on the desired order by pointing to positions on the table and by simultaneously specifying which cup should be placed at that very position using speech. Figure 1(b) shows a user specifying the desired order of the cups by pointing.

Alg. 1 shows the READYLOG procedures used in the Order Cups Demo. Once CAESAR has collected all the necessary information it determines an execution strategy for the non-deterministic procedure *sort_cups* such that the desired arrangement of the cups is achieved eventually. This is done by means of decision-theoretic planning with respect to a reward function. In our scenario, CAESAR computes the re-ordering with a minimum number of movements.

Apart from constructs known from imperative programming languages READYLOG also offers less common non-deterministic constructs like **pickBest**. The former initiates decision-theoretic planning, the latter is the non-deterministic choice of argument. During planning, the logical specification of the dynamics in the world can be used to reason about the state of the world after executing a program, here that is a sequence of *move_cup* actions. Non-determinism is resolved by opting for those choices that maximize the reward function.

[2] A video of the demonstration is available at `http://goo.gl/7rEEY`.

The application described above shows the potential benefit of deliberation and that it can be integrated in the behavior specification of a robot very easily. Also, this kind of high-level control could be used for more complicated tasks such as planning the course of actions in daily activities of an elderly person. The decision-theoretic optimization can also be used for things like path planning. For example, in an extended version of the demonstration the robot drives around looking for different tables with cups on them, picks up cups from those tables and moves cups from one table to another. Then, the robot also uses its collision avoidance and navigation capabilities [10] to get around and remember table positions. This version also make use of pointing gestures to allow for a human user to select a specific cup to be moved somewhere else.

4 Conclusions

In this paper, we presented a demonstration where our interactive domestic service robot Caesar orders cups on a table. The demo integrates basic capabilities for autonomous mobile robots with methods for robust human-robot interaction. We showcased that the robot can perform complex tasks in domestic settings and how it can benefit from a powerful high-level control in such tasks.

References

1. Ferrein, A., Lakemeyer, G.: Logic-based robot control in highly dynamic domains. Robotics and Autonomous Systems **56** (2008) 980–991
2. Levesque, H.J., Reiter, R., Lespérance, Y., Lin, F., Scherl, R.B.: Golog: A logic programming language for dynamic domains. J Logic Program **31** (1997) 59–84
3. Wisspeintner, T., van der Zant, T., Iocchi, L., Schiffer, S.: Robocup@home: Scientific Competition and Benchmarking for Domestic Service Robots. Interaction Studies. Special Issue on Robots in the Wild **10** (2009) 392–426
4. Doostdar, M., Schiffer, S., Lakemeyer, G.: Robust speech recognition for service robotics applications. In: Proc. of the Int'l RoboCup Symposium 2008 (RoboCup 2008). Volume 5399 of LNCS., Springer (2008) 1–12
5. Schiffer, S., Baumgartner, T., Lakemeyer, G.: A modular approach to gesture recognition for interaction with a domestic service robot. In: Proc. Int'l Conf. on Intelligent Robotics and Applications. LNCS, Springer (2011) 348–357
6. Niemueller, T., Ferrein, A., Beck, D., Lakemeyer, G.: Design Principles of the Component-Based Robot Software Framework Fawkes. In: Proc. Int'l Conf. on Simulation, Modeling, and Programming for Autonomous Robots, Springer (2010)
7. Niemueller, T., Ferrein, A., Lakemeyer, G.: A Lua-based Behavior Engine for Controlling the Humanoid Robot Nao. In: Proc. Int'l RoboCup Symposium 2009 (RoboCup 2009). Volume 5949 of LNCS., Springer (2009)
8. Reiter, R.: Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems. The MIT Press (2001)
9. Boutilier, C., Reiter, R., Soutchanski, M., Thrun, S.: Decision-theoretic, high-level agent programming in the situation calculus. In: Proc. of the 17th Nat'l Conf. on Artificial Intelligence (AAAI'00), Menlo Park, CA (2000) 355–362
10. Jacobs, S., Ferrein, A., Schiffer, S., Beck, D., Lakemeyer, G.: Robust collision avoidance in unknown domestic environments. In: Proc. Int'l RoboCup Symposium 2009 (RoboCup 2009). Volume 5949 of LNCS., Springer (2009) 116–127