Robotics and Autonomous Systems 62 (2014) 619–631

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

# Deploying a modeling framework for reusable robot behavior to enable informed strategies for domestic service robots

Frederic Siepmann [a,b,*], Leon Ziegler [a], Marco Kortkamp [a], Sven Wachsmuth [a,b]

[a] Applied Informatics, Faculty of Technology, Bielefeld University, Germany

[b] Central Lab Facilities, CITEC, Universitätstraße 25, 33615 Bielefeld, Germany

ARTICLE INFO

*Article history:*
Available online 22 November 2012

*Keywords:*
Robot behavior modeling
Informed strategy
Attention mapping
Visual search
Service robot

ABSTRACT

For the development of multi-purpose robots that can operate in a wide range of different situations we will need sophisticated behavioral building blocks to compose the desired performances. In this paper we argue for a behavior modeling framework which provides specific behavior interfaces for implementing robot *skills* that stay close to evaluation cycles and to rich fused sensory data. This way we ensure reusability and facilitate what we call *Informed Strategies*. As a specific application we deploy the framework to an object search task for a domestic service robot. The presented behavior involves an attention mapping mechanism based on 2D and 3D visual cues. We show the advantages of the proposed approach by conducting an evaluation in a real-world apartment scenario as well as by successfully taking part in the RoboCup@HOME competition.



## 1. Introduction

Advancements in various research areas over the last decade, such as SIFT [1] or the Kinect sensor [2], have been found to be very important for robotics and have generated a number of systems that incorporate these advancements and capabilities and leverage them in the real world. This implies that today robotic systems obtain even more of their abilities through the combination of different software components from different areas. One of the most important areas for autonomous robots is perception. To be able to communicate with humans and interact with the environment, robots not only need to perceive their surroundings, they also have to interpret the current scene. This ability becomes even more important for more complex scenarios, such as domestic service robots.

Service robotics, which currently may be one of the most complex scenarios for robots, is a growing field of interest, and with all the gained capabilities it has become feasible to autonomously perform tasks in regular domestic home environments with a robotic system. A basic task for such robots is for example "fetch and carry", where the robot has to fetch a known object from another place and deliver it to the human. Variants of this task can be found in actual robot competitions, like the "mobile manipulation challenge" at ICRA 2010, the "semantic robot vision challenge" or the "RoboCup@HOME" competition. We have seen a growing number of robots competing in RoboCup@HOME, not playing soccer but following persons, searching for objects, grasping objects and many other tasks in a home-like environment. Now the next step is to move on from fetch and carry and alike scenarios to more complex tasks. At the lowest level of complexity a robot task may assume that, for example, a map, and the positions of an object and of a person are known in advance. At a higher level of complexity the robot needs to be able to act more autonomously and interpret the environment; for example, first it has to search for the object and then find the person it should be delivered to. Despite the fact that the single *skills* needed to perform such tasks are well established, robots frequently fail if they need to show them in a real-world scenario. The system integration aspect to efficiently combine *skills*, which typically is underestimated, is one factor why robots are failing in complex environments. Another factor is that robots need to acquire knowledge from the environment and exploit semantic information of the scene to be able to robustly perform in the real world. Arkin [3] describes a robot as "[…] *a machine able to extract information from its environment, and use this knowledge to move safely, in a meaningful and purposive manner*". He constructs robot behavior from building blocks, consisting of the knowledge of how to act in and how to perceive the environment as well as the computational processes underneath. In this regard we distinguish in this paper between *informed* and *uninformed* robot behavior. Uninformed robot behavior refers to mainly reactive behavior in which the robot facilitates information mainly processed from a

* Corresponding author at: Central Lab Facilities, CITEC, Universitätstraße 25, 33615 Bielefeld, Germany. Tel.: +49 5211062950; fax: +49 5211066488.

*E-mail addresses:* fsiepman@TechFak.Uni-Bielefeld.DE (F. Siepmann), lziegler@TechFak.Uni-Bielefeld.DE (L. Ziegler), mkortkam@TechFak.Uni-Bielefeld.DE (M. Kortkamp), swachsmu@TechFak.Uni-Bielefeld.DE (S. Wachsmuth).

0921-8890/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2012.10.013

single source. This can be for example a behavior for obstacle avoidance, where the distances to the obstacles (for example measured with a laser sensor) are taken into account to guide the robot. In contrast, an informed behavior is not purely reactive and introduces additional information by interpreting the environment where data from many sources can be processed. The focus of this paper, the combination of a modeling framework for the system behavior and the component interaction underneath, will account for the problem of effectively combining robot *skills* as well as for acquiring and facilitating this extra information about the environment.

This capability of a robot to acquire and exploit additional information about the surroundings and utilize it for actions and behavior is already used in some places. The challenge is that often an additional acquiring step is needed: for example, analyze the current scene or record a map. The complexities of the new scenarios tend to produce complex and hard-coded robot behavior that is difficult to maintain and difficult to reuse. What we want to achieve with the *informed strategies* or *informed robot behavior* is to acquire the additional information on the fly, without an extra analyzing step, and produce reusable behavior code that can be improved over time. For that matter our framework is available as open-source software.[1] Since this kind of behavior must involve all available *sensor* inputs and generated higher level information, e.g. laser data as well as the positions of a person, it is crucial to embed these *strategies* into the overall system architecture to enable simple and flexible construction of new robot behavior.

In this paper, we will concentrate on two parts of this challenge. How can informed *strategies* be embedded into the system architecture to easily generate informed robot behavior and what methods do we use to acquire knowledge from the environment? We will show how we were able to improve the search behavior of our robot using informed search *strategies* and how we combined the various information from different components into one coherent system behavior.

The exemplary search behavior uses a target-directed search *strategy* for known objects in unknown rooms. It combines (i) a two-step object recognition approach that applies a color-based top-down attention filter as a first step, (ii) the exploitation of scene geometry for the extraction of appropriate places for objects in a room, and (iii) a SLAM-like probability grid approach that dynamically builds a semantically annotated map of the environment. Finally, we will present insights from user studies and a detailed evaluation of the search behavior that we have conducted in a real-world apartment with our robot.

### 1.1. Related work

That system architecture and integration of software components from many different research areas is especially crucial for the development of multi-purpose robotic systems is widely accepted. Early works such as that of Brooks in 1986 [4] or of Gat in 1997 [5] investigated the architectural principles for complex robotic systems. Based on that, the work of Nesnas [6] and colleagues proposes a *Coupled Layered Architecture for Robotic Autonomy* (*CLARAty*), which is a domain-specific robotic architecture that focuses on the reduction of the overhead of developing custom robotic infrastructure, e.g. middleware, and on integrating components from various domains that can be reused on different hardware platforms. They distinguish two layers of components, the *Functional Layer* and the *Decision Layer*, which comprehend certain properties that structure the architecture and ease up the interaction between components. The Functional Layer facilitates an object-oriented system decomposition to achieve reusability whereas the Decision Layer is tightly coupled for planning, scheduling, executing and monitoring actions. They do not explicitly model the system behavior or the interaction strategies with a user or the environment.

As far as planning is concerned, robots that are acting in the real world have to deal with environments that make planning particularly difficult: there is limited knowledge about the current state of the world and there are also very dynamically changing environments. This problem, where "classical" AI planning methods fail, has been addressed for example by Brenner and Nebel [7], who introduce their continual planning approach that enables the agent to decide why and when to switch between the planning phase and acting. Hence it is possible to postpone parts of the planning and first actively gain more information for later planning steps.

It is also widely accepted that humans utilize some kind of attention system when inspecting a scene to guide the visual focus through saccades to interesting parts of the field of view for fixation [8]. This fact also leads to a non-uniform resolution of details in the scene. Areas that have been focused are perceived more accurately than areas that were only covered by peripheral vision. One of the most noted approaches to implement a pure stimulus-driven bottom-up attention control system was provided by Itti and Koch [9]. A variant of this approach (see [10]) is used by Meger et al. [11] on their mobile robot "Curious George" for a visual searching task. As a further cue for visual attention they use a horizontal surface finding algorithm developed by Rusu et al. [12]. Their approach makes behavioral decisions based only on the information from the current field of view. They do not implement an integration step to map the attentive regions over time in a spacial way.

In [13], Shubina and Tsotsos argue for using attentive cues that optimize the search process of a robot. They propose a greedy algorithm that considers the cost and effect of different actions. Various kinds of a priori knowledge are utilized, like objects in spatial proximity (also proposed by [14,15]), saliency knowledge (see e.g. [16]), or spatiotemporal constraints (see also [17,18]). As they only address the problems of "Where to look next" and "Where to move next", a temporal and spacial integration of the attentive cues is not implemented in their approach.

A probabilistic approach utilizing semantic knowledge about the environment to find persons in a mobile robot's surrounding is proposed by Stückler et al. [19]. Their system fuses multiple sensory data to build up spacial probability maps for person location hypotheses. The results are improved by prior information from semantic scene knowledge about the spacial context. This is similar to our attention mapping approach which is described in Section 3.1. Additionally our approach is coupled more closely with the behavior modeling (see Section 2.2).

## 2. Reusable behavior modeling and informed strategies

Complex robot behavior requires a number of different software components to enable the robot to properly act in a domestic environment. To integrate these different kinds of components and methods nowadays a number of existing robotic frameworks assist the developers with hardware abstraction for code-driven development and with middleware functionality, for example Player/Stage [20], YARP [21], XCF [22] or ROS [23]. With these frameworks it is possible to deploy autonomous robots in different scenarios and applications. The variety of frameworks though indicates that neither any existing modeling approach nor available libraries can cover all aspects necessary in robotic applications. The resulting variety of frameworks poses a difficult challenge for developers of multi-purpose robots: how does one model, implement and improve robot behavior that can be reused

[1] http://opensource.cit-ec.de/projects/bonsai.