Proc. Workshop "Dynamische Perception", Ulm, 2000, pp. 191-194, Akademische Verlagsgesellschaft/ infix 2000

A Behavior-Based Dynamic Control Architecture for a Mobile Shopping Assistant*

T. Wilhelm, H.-J. Böhme and H.-M. Gross

Ilmenau Technical University, Department of Neuroinformatics
89684 Ilmenau (Thuringia), Germany
{wilhelm, hans, homi}@informatik.tu-ilmenau.de

Abstract

This paper describes the control architecture for our interactive mobile shopping assistant PERSES (PERsonal SErvice System). In the context of behavioral robotics, we developed an arbitration system as variation of the dynamic arbitrator, which coordinates behavior selection utilizing a dynamical system with a set of conditions that define the position of the fixed points depending on the input, and the actual system state. The paper also includes a description of the arbitrators application to the PERSES scenario and an overview of our ongoing research.

1 Introduction

In this paper we present our dynamic arbitration system and its application to our ongoing research project PERSES (PERsonal SErvice System), which deals with the mainly vision-based interaction of a human user with a mobile service-robot operating as an interactive shopping assistant in a home improvement store. Our shopping assistant must be able to actively observe the operation area; to detect, localize and track a potential user; to contact her, interact with her continuously and navigate safely through the market while offering its services according to the actual situation.

Because of the complexity of the task, we adopt the behavior-based approach to robotics, which allows for a hierarchical decomposition of the problem into separate behavioral modules or subsystems. These behaviors can be developed incrementally and integrated successively. According to the definition given by Arkin [1] we consider *behavior* as: "Anything an organism does involving action and response to stimulation." Thus, a behavior needs not necessarily invoke observable actions in the environment but may modify internal representations, too. An external or internal stimulation together with the internal knowledge of the robot form the actual *situation context*, which is the trigger of one or more behaviors that are controlled by a dynamic arbitration system.

As formal framework for behavior selection we choose the so-called dynamic approach to robotics. The central part is a variation of the Dynamic Arbitrator, proposed by Bergener et al. [2] and consists of a dynamical system, defining conditions between behavioral modules. Dynamical systems seem to be the ideal tool to develop systems with complex

---

[*] Supported by a Thuringian Ministry of Science, Research, and Art Grant (B 611-98041, PERSES-Project)

behavior in a flexible but stable way. They can model time-continuous systems in abstract state spaces that develop in time, depending on the systems state and on a continuous input. Changes of the position of the system's fixed points produce bifurcations of the underlying differential equations and thus achieve the required flexibility in generation of behavior. Our idea is to use a hierarchy of arbitration systems working on multiple abstraction levels.

2 Arbitration System

The dynamical system consists of a differential equation and a set of conditions, that define its fixed points. The state $\mathbf{n}$ of the system is defined by

$$\tau \dot{n_i} = -n_i + \theta_{Input,i}(\rho_i) \cdot \theta_{Inhibition,i}(\mathbf{n}) \cdot \theta_{Precondition,i}(\mathbf{n}), \tag{1}$$

τ being the time scale of the arbitrator. For each behavior there exists one component n_i of the state vector $\mathbf{n}$. State variables take on continuous values between 0 and 1, depending on the conditions θ, that take on a value of 1 if complied with and 0 otherwise, so equation 1 has a stable fixed point at $n_i = 1$ if all the conditions are fulfilled and $n_i = 0$ otherwise. The state is the output of the arbitrator, where a value of 1 means, that behavior i is allowed to be active. The three conditions are
the **input condition**:

$$\theta_{Input,i}(\rho_i) = \rho_i \qquad \text{with} \qquad \dot{\rho_i} = \frac{(1-\rho_i)Q_i}{\tau_{\rho,2}} + \frac{\rho_i(Q_i - 1)}{\tau_{\rho,1}}, \tag{2}$$

where ρ_i is the input filter and Q_i is the binary situation context of module i, that states, whether it wants to be active ($Q_i = 1$) or not ($Q_i = 0$). When it is switched on, ρ_i follows on a slow time scale $\tau_{\rho,2} \gg 1$, while a deactivation is followed instantaneously on $\tau_{\rho,1} \simeq 1$. The input filter thus prevents behavioral oscillations in that a deactivated behavior can not be reactivated immediately after its situation context is switched on again.
The **inhibition condition**:

$$\theta_{Inhibition,i}(\mathbf{n}) = 1 - S\left(\sum_j \mathbf{I}_{i,j} \cdot n_j + \xi_t\right) \tag{3}$$

checks if any behavior j were active, that inhibits module i. This condition is useful for behaviors that work on the same hardware resources. S is a sigmoid with threshold 0.5 and steep slope. In the inhibition matrix $\mathbf{I}$ an element i, j is set to 1, if behavior j shall inhibit behavior i. The noise term ξ_t is needed to separate the dynamics in case the situation context is switched on for two competing modules at the same time.
The **precondition**:

$$\theta_{Precondition,i}(\mathbf{n}) = 1 - S\left(\sum_j \Phi_{i,j} \cdot (1 - n_j)\right) \tag{4}$$

tests the precondition matrix Φ for the activation of all the behaviors j that are supposed to be enabled before behavior i may be activated, thus defining behavioral dependencies. A behavioral sequence can be generated, when one module provides the situation context for the other. If multiple modules become active at the same time, their output needs to be combined, e.g. using a weighted sum or other methods of integration.

3 PERSES system architecture

3.1 Global system

The aspired architecture consists of three meta-behaviors: *User Localization*, *User Log In* and *Interactive Tour*, each utilizing its own arbitration structure (figure 1). They work sequentially, anyones output supplying the situation context of the next. The architecture can be seen as a state machine with the arbitrator and the situation context defining its transitions. When any subsystem finishes its task, it triggers the next one by delivering a package of data that is part of its situation context.

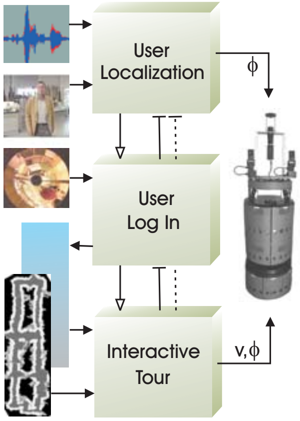

Figure 1: Global control architecture of our Personal Service System PERSES. The relations between the meta-behaviors are directly drawn, small bars indicating inhibition, while the cupped arrows show the situation context. Each meta-behavior inhibits its predecessors, which results in a sequential system structure, where the arbitrator generates smooth transitions between meta-behaviors. The internal knowledge, that is the global occupancy map of the operation field and the specific vision-based user model (bottom left), is learned over the course of a separate exploration run of the robot and during the user login phase, respectively.

3.2 Subsystem User Localization

We use a multi modal approach that integrates both visual and acoustic stimuli into the localization process (figure 2). The *Vision-based User Localization* detects movement in the omnidirectional view. For the acoustic localization of a potential user, we developed a biologically inspired model of binaural sound localization. The module *Acoustic User Localization* calculates the angle to the sound source from phase shift between the signals [4]. There is no arbitrator involved here, as the two modules work in parallel with no inhibitions or preconditions defined among them. The fusion module uses both cues to find the direction in which most likely a user is found and triggers the rotation of the robot towards the potential customer. The *User Verification* looks for skin color, head-shoulder contour and the structure of a human face, and combines the three cues to yield a robust measure if and where a user can be found [3].

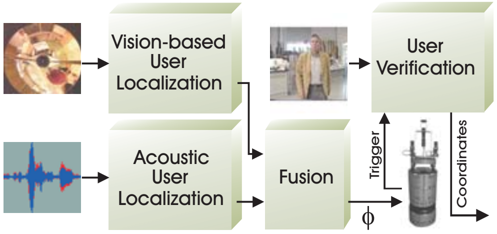

Figure 2: *User Localization* subsystem. The behavioral system consists of the cooperating behavioral modules, the fusion mechanism and the User Verification.

3.3 Subsystem User Log In

After a potential user has been detected, the *Learn User Model* behavior builds a model, suitable for tracking her, which is the situation context for the *User Dialog* (figure 3). It waits for a request for an article on the touchscreen and uses a database to guide the user through the dialog. The coordinates of the desired article are mapped to a probabilistic map of the store [5]. During operation, the *User Log In* subsystem inhibits the User Localization subsystem to keep the robot from paying attention to visual or auditory stimuli. If no login occurs for a certain amount of time, the user model is inactivated by the *Erase User Model* module, putting the subsystem back to an inactive state.

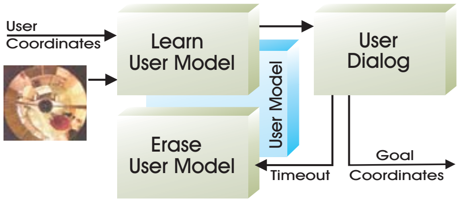

Figure 3: Meta-behavior *User Log In*. After a user model is learnt, the robot engages in a dialog, resulting in the coordinates of an article. If no login occurs, the model and the whole subsystem is set inactive.

3.4 Subsystem Interactive Tour

This meta-behavior (figure 4), is initiated when the *User Log In* provides a desired goal position in the market. The *User Guidance* plans a route with an occupancy map. The goal of the *User Tracking* module is to keep the user in the omnidirectional view, while updating the model regularly. It is not intervening as long as its desires are satisfied, but takes over execution when the user falls behind. The *Homing behavior* monitors the energy level and, when a critical threshold is reached, is allocated exclusive access to the drive. All three modules calculate a v-ϕ-map and pass it to the *Obstacle Avoidance*, which suppresses motor commands conflicting with the actual obstacle configuration. To this end, it pessimisticly incorporates ultrasonic sensors, optical flow and inverse perspective maps. It is possible however, that the intention of the planner can not be realized, because of dynamic obstacles. In this case, the local map is passed to the planner, which generates a new plan incorporating the local and global map.

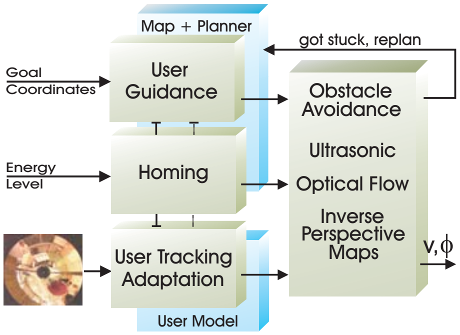

Figure 4: Structure of the submodule *Interactive Tour*. Three modules compete for the robot drive (angle and velocity). The User Guidance reaches for the target position, while the User Tracking strives to keep the robot near the user, the later having preference over the first. If the energy level gets low, the Homing behavior inhibits all other modules.

4 Outlook and future work

The arbitration system can be shown to work in simulations and some meta-behaviors, but is not jet introduced to all the described submodules, because single behavioral modules, such as the user model, are not yet available. Thus major effort focuses on the development of those modules and the assemblage of the system. Another part that is still missing is a more intuitive and flexible interaction with a human user. In our practical scenario, it is crucial for the robot to offer its services in a way, comprehensible for unintroduced users.

References

[1] Arkin R.C. (1998) Behavior-Based Robotics. MIT Press.

[2] Bergener T., et al. (1999) Complex behavior by means of dynamical systems for an anthropomorphic robot. Neural Networks 12, pp 1087-1099.

[3] Böhme H.-J., et al. (1999) Person Localization & Posture Recognition for Human-Robot Interaction. In GW'99 - The 3rd Gesture Workshop, Gif-sur-Yvette, pp. 105-116. Springer, Lecture Notes in Artificial Intelligence 1739.

[4] Paschke P., Schauer C. (1999) A spike-based model of binaural sound localization. International Journal of Neural Systems 9(5), pp 447-452.

[5] Gross H.-M., Böhme H.-J. (2000) Perses - a Vision-based Interactive Mobile Shopping Assistant. Proc. IEEE-SMC Conference, Nashville 2000.