F. Tièche, C. Facchinetti, and H. Hügli, "Multi-Layered Hybrid Architecture to Solve Complex Tasks of an Autonomous Mobile Robot," Proc. Third Golden West International Conference on Intelligent Systems, Las Vegas, 1994

Multi-Layered Hybrid Architecture to Solve Complex Tasks of Autonomous Mobile Robots

François Tièche, Claudio Facchinetti, Heinz Hügli
University of Neuchâtel, Institute for Microtechnology
Rue de Tivoli 28, CH-2003 Neuchâtel, Switzerland
(`tieche@imt.unine.ch`)

Abstract

We present the architecture, implementation and tasks of an autonomous mobile robot controller developed according to the principle of hybrid architecture. The paper describes the general structure and function of its mains elements and analyses the development of a given task showing the advantages of the hybrid architecture.

1 Introduction

The ability of a mobile robot to achieve complex tasks depends essentially on the architecture of its controller. We use a multi-layer hybrid architecture to take advantage of several traditional approaches such as symbolic, behavioural, control theory based architectures. We distribute hence these competence levels on several layers: the top layer is responsible for symbolic planning, the intermediate layer is behavioural-based, and finally the bottom layer controls the robot. Our architecture extends the behavioural approach for more complex tasks, offering basically the possibility to excute sequences of simple behaviours.

For the demonstration of the capabilities of the selected architecture, we have chosen a task where the robot moves and aligns chairs in a room, using a sequence of simple vision-based behaviours.

This architecture is realised in the form of a development environment called MANO. The main features are (i) an interface controlling the robot, (ii) a blackboard centralising the information exchange, and (iii) a set of concurrent processes.

2 Related work

Among the different designs for mobile robot architectures, we distinguish several main approaches. A first one is based on **control theory** (signal-processing, process control), generally implemented either on hardware or on dedicated systems in order to achieve high speed. The lack of high level control interfaces prevents an easy design of complex tasks. Systems based on **symbols** in turn can express complex tasks but cannot control real robots since symbols are not adequate to represent the physical world. A possible link between these two approaches is the **hierarchical control** architecture filling the gap between low level control and high level planning. Within this approach, a data path leads from the sensors to the highest level, and commands are fed back to the acuators. While the system is advantageous due to its modularity, often it does not fulfil the speed requirements. The **subsumption architecture** [1] — or behavioural approach — speeds up the control by decomposing a task in a set of behaviours that react directly with the environment. There are several layers of behaviours. At each level, the upper one activates or deactivates the behaviours of the underlying layer. For real applications, it is often difficult to partition a global task on a set of elementary behaviours implemented on a robot: (i) the decision organ is distributed over several behaviours, and (ii) there is no modelling of the robot's world.

Our architecture attempts to avoid these difficulties by the use of a **hybrid architecture** [2] [5]. It takes advantage of the approaches describe above: a good modularity which makes the development easy, an efficient low level control, an intermediate behavioural control, and symbolic reasoning. It is a multi-layered architecture which separates the different levels of competence. Each layer operates asynchronously with respect to the others. The main characteristics of this architecture are described below:

- The lowest level controls the robot using control theory.
- At intermediate levels, the robot is controlled by a set of behaviours reacting directly with the environment.
- The highest level deals with symbolic reasoning,

planning, and task scheduling to reach the goals.

3 Architecture

Our architecture is composed of four layers (Figure 1). The lowest one called *sensori-motor* is based on control theory. It is responsible for the elementary movements of the robot and treats the data provided by the sensors. The second layer is behavioural-based and controls the robot with respect to the environmental characteristics. The sequencing layer links the symbolic level with the behavioural level. It sequences the simple behaviours such that more complex tasks can be performed. The planning level has both global and symbolic knowledge of the world. It is used to define long term strategy to reach a given goal.

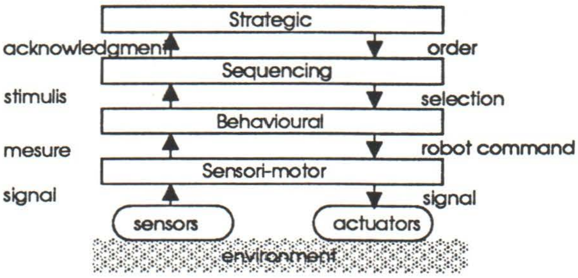

Figure 1: Architecture of the controller.

3.1 Sensori-motor layer

This layer is characterised by fast sensori-motor interactions and is usually hardwired. The movements of the robot are controlled by servo loops, both for speed and position. It is also used for the processing of sensor data which — at this level — are essentially local measures of the world.

3.2 Behavioural layer

This layer is made of a set of concurrent behaviours. We call *behaviour* the closed loop formed by a robot command, actuators, sensors, and the sensed data. The set of behaviours defines the interaction of the robot with its environment. Each behaviour extracts specific world characteristics from the measures provided by the sensori-motor layer: we call them *sign pattern*. Each time an expected sign pattern appears, the behaviour is said stimulated. It then controls the robot such that the sign pattern remains present.

At any time, one single behaviour is autorised to send commands to the robot: the behavioural layer is under the control of the sequencing layer which selects one among the stimulated behaviours. Nevertheless, each behaviour informs the sequencing layer on its internal state using so-called stimuli signals which can take the values *not stimulated, stimulated, satisfied.*

3.3 Sequencing layer

A robot task is composed of a sequence of behaviours, solving each a small part of the task. The problem is hence to describe their sequencing. According to a given strategy, this task selects and parametrises one-by-one the suitable behaviour according to the stimuli it receives. It gets a local goal to reach from the top level and returns informations on its execution status.

3.4 Planning layer

At the planning level, the long term goals are achieved by scheduldjng the individual tasks. Well known AI methods based on symbolic reasoning are used.

4 Development environment MANO

MANO is the software development environment [4], used to test mobile robot task execution. It is implemented using our hybrid architecture (Figure 2). A robot Nomad200 from Nomadics Technology provides sensors of different types: sonars, infrared range-sensors and tactile sensors. Two vision-based sensors have been added: a system based on landmarks and a second system using structured light [3]. The former uses active illumination to enhance the contrast of the landmarks made of reflective material. The latter delivers a 3D profile of the environment in front of the robot.

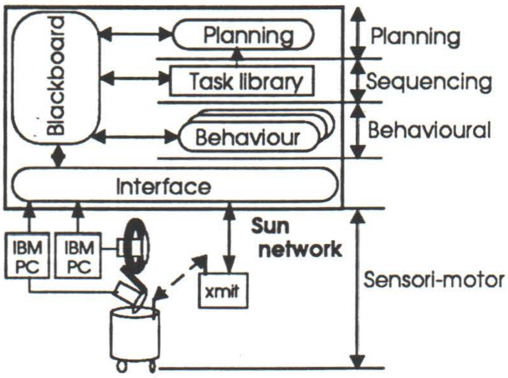

Figure 2: Development architecture MANO.

The figure shows also the four layers of the hybrid architecture. The sensori-motor layer is implemented on dedicated hardware: the servo loops controlling the

F. Tièche, C. Facchinetti, and H. Hügli, "Multi-Layered Hybrid Architecture to Solve Complex Tasks of an Autonomous Mobile Robot," Proc. Third Golden West International Conference on Intelligent Systems, Las Vegas, 1994

robot are on board and the vision data preprocessing is done on two PCs. The three upper layers are implemented on a network of SUN stations: (i) the behaviours are fully independent and run concurrently as individual Unix processes, (ii) the tasks are implemented in form of a library, and (iii) the planning is realised as a Unix process calling the tasks. The communication between planning, behaviours, and the robot is performed in the form of a blackboard.

5 Application: Tidy up chairs in a room

The complex task we implemented on MANO as an example is *tidying up chairs in a room.* To execute this task, the mobile robot needs both adequate navigation and strong interaction with its environment. On this example, we illustrate the decomposition of the task with respect to our multi-layered hybrid architecture. The task requests the robot to detect chairs which are located arbitrary in a room, and to push them up to a line defined with respect to two landmarks. In order to avoid drift, after each chair pushed, the robot returns to a homing area where it positions itself with respect to the homing landmarks. This task uses two vision sensors: the *vision by landmark* detects chairs marked with reflective material and the homing landmarks while *vision by structured light* detects obstacles in front of the robot. Figure 3 shows the tidy-up task decomposed in a sequence of simple behaviours. First the robots makes a homing (behaviour:HO), then it looks around to detect a chair (SC). If a chair is found it goes toward the selected chair (GT). The robot must be positionned on the side of the chair opposite to the push-back direction. Therefore, the robot turns around the chair (AC) and then pushes it until the line is reached (PC). Finally it goes back to the homing area (GH) and readjusts its position (HO).

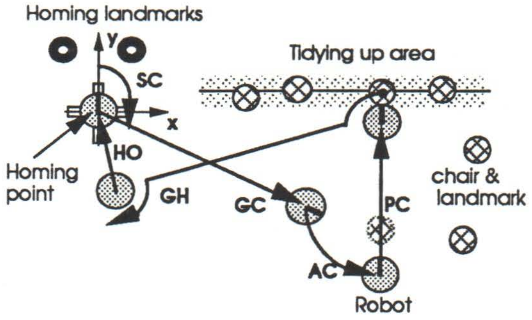

Figure 3: *TidyUpChair* decomposed by behaviours.

The *TidyUpChair* task performed as desired. Many tests have been run with various chairs and homing landmark configurations: the programmed strategy invariably leads to the kind of path shown in figure 3.

6 Conclusion

In the present paper we have shown a hybrid architecture used to control a mobile robot, its implementation in the form of a development environment and an example task: *TidyUpChair.* The hybrid architecture is four-layered: sensori-motor, behavioural, sequencing, and planning. The main advantage of the approach proposed is the extension of the behavioural approach: we showed how the sequencing layer can combine simple behaviours to achieve a more complex task. In particular, we have presented the *TidyUpChair* task as well as its decomposition with respect to the architecture proposed. It has been implemented in MANO, the test environment we developed, and the feasability has been shown.

References

[1] R.A. Brooks, "A Robust Layered Control System for a Mobile Robot System," IEEE Journal of Robotics and Automation, Vol. RA-2, No 1, pp. 14-13, March, 1986.

[2] J. H. Connell, "SSS: A Hybrid Architecture Applied to Robot Navigation," Proc. IEEE Int. Conf. on Robotics and Automation, Nice, France, May 1992.

[3] H. Hügli, G. Maître, F. Tièche, C. Facchinetti, "Vision-based behaviours for robot navigation," Proc. Fourth Annual SGAICO Meeting, Neuchâtel, Switzerland, September 1992.

[4] H. Hügli, F. Tièche, F. Chantemargue, G. Maître, "Architecture of an experimental vision-based robot navigation system," Proc. Swiss Vision, Zürich, Switzerland, September 1993.

[5] M. G. Slack, "Autonomous Navigation of Mobile Robots for Real-World Applications," in Interdisciplinarity Computer Vision, SPIE, Vol. 1838, pp. 101-107, 1992.