A case study in the behavior-oriented design of autonomous agents.

Luc Steels
Artificial Intelligence Laboratory
Vrije Universiteit Brussel
Pleinlaan 2, B-1050 Brussels, Belgium
E-mail: `steels@arti.vub.ac.be`

Abstract

The paper documents a case study in the design and implementation of a robotic multi-agent system. It illustrates known design guidelines, namely that the physics of the environment must be exploited, that behavior is the result from the interaction dynamics between the agent and the environment, and that emergent behavior can and must be utilised whenever possible. But the case study also challenges certain views, such as the subsumption architecture, the need for an action selection mechanism, the goal-oriented design methodology dominating the literature on planning, and the algorithmic style of writing control programs. Alternatives are explored in the form of a cooperative, parallel, behavior-oriented design.

1 Introduction

An autonomous agent is a physical system that has its own resources to operate independently in a dynamically changing real world environment. The resources include energy, computational power, sensors, actuators, and body parts. A multi-agent system is an ecosystem in which two or more autonomous agents cooperate. The paper describes an experiment exploring cooperation in multi-agent systems. The experiment focuses on individual cooperation, such as found between two birds that are rearing young, rather than societal cooperation as observed in insect societies. Instead of taking a knowledge-oriented approach, in which the agents make models of each other and negotiate through explicit natural language-like communication, we explore a behavior-oriented approach [11] in which cooperation is forced upon the agents by the environment and emerges from the activities of individual agents. McFarland [7] has defined the biological background and motivation for the experiment.

Designing autonomous agents and multi-agent systems is notoriously difficult. The paper intends to illustrate a set of design guidelines about which there seems to be a consensus in the field [2]: (a) exploit the physics, (b) exploit the interaction dynamics between the agent and the environment, and (c) use emergent behavior when possible. It also illustrates some novel principles (see [10] for a more extensive discussion): (a) use a cooperative as opposed to a subsumption architecture, (b) use parallelism as opposed to action selection, (c) perform a behavior-oriented as opposed to goal-oriented design, and (d) view control programs as dynamical systems.

The first part of the paper describes the main characteristics of the experiment. The second part describes the realisation in terms of physical robots. The final part discusses the design guidelines explored in the experiment.

2 The experiment

The agents used in the experiment are relatively small robotic agents (30 x 10 cm and 20 cm high). They are equipped with sensors, motors, rechargeable batteries, a central processor in the form of a small PC-like computer, and a sensory-motor board to offload most of the sensor and actuator processing from the main processor. Prototypes of the agents have been built using Lego^{TM}-technology (figure 1) (as in [4] or [3]). A more robust version of the agents, which can operate for days in a row, is currently under construction.

The first two characteristics of the experiment center around a single agent operating in an environment in which it can gather energy but also benefits from weakening competition for the same energy. The remaining characteristics have been chosen to bring in a multi-agent perspective.

[1] An agent can recharge itself.

The primary goal of an autonomous agent is to sustain itself. This implies at least that the agent has at all times enough energy to keep on functioning. Each agent has therefore a set of batteries and the capability to recharge itself. The environment contains a charging station with two disks mounted on poles (figure 2). The agent has two charging rods sticking out, one at the top and one at the bottom. When these charging rods make contact with the disks, current is drawn and the battery starts charging. There is a continuous supply of energy to the

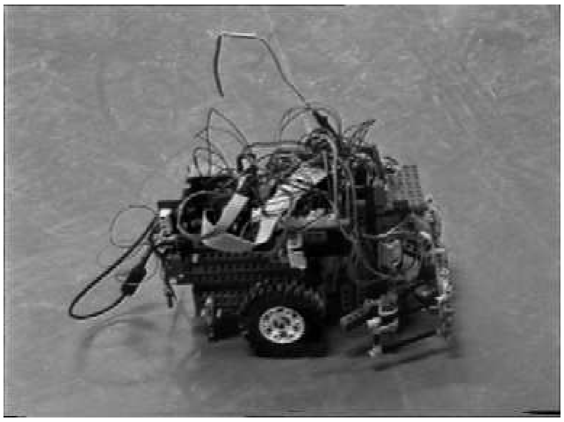

Figure 1: Prototype of robotic agents used in the experiment. The body has been built with LegoTechnics^{TM}. The main processor is a pocket PC computer inserted in the robot body.

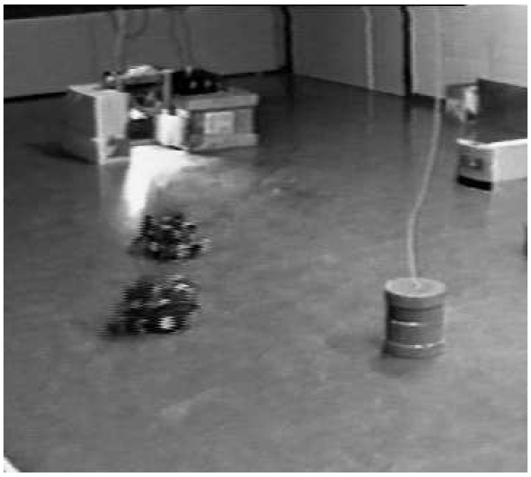

Figure 2: The complete ecosystem consists of two or more agents, obstacles, a charging station and boxes with lamps mounted in them.

whole system.

[2] There is competition for energy in the form of additional lights.

There are lamps which are in competition for the overall energy available in the ecosystem. The lamps are mounted in boxes installed in the environment (figure 2). They draw current from the same total energy source as the charging station. Lights can be diminished by an agent as it knocks repeatedly against the box in which the lamp is mounted. Lights regenerate slowly after they have been dimmed. There are additional obstacles in the environment which do not act as competitors for energy but need to be avoided by the robot as it is seeking out box-lights. Energy stored in the charging station is drained, if all lamps are fully on. Hence it is in the interest of the agent to move away from the charging station to dampen the box-lights.

The boxes as well as the charging station must be detectable by the agent. The charging station emits blue light and the boxes emit yellow light. The agent performs phototaxis using photosensors mounted on the left and right sides of the body. They are covered with filters to be only sensitive to yellow or blue light.

[3] There is an opportunity for cooperation between agents.

Obviously it would be beneficial for the agent which is recharging, if there were another agent weakening the competition for the available energy. So there is an opportunity for cooperation. While one agent recharges, the other one seeks out boxes and pushes against them to dim the lights. In a first series of experiments we use two agents, but a larger group is planned.

[4] One agent cannot survive.

The agents can cooperate but they are also in competition and can potentially exploit each other. One agent may stay on the charging station while the other agent dims the lights, eventually running out of energy. To avoid this situation, the experiment is set up in such a way that one agent cannot survive on its own. Consequently it is in its own interest to let the other agent occasionally get on the charging station. Each agent has the capability to emit sound and to perceive sound through a microphone. Using this capability an agent can indicate that it needs to be on the charging station.

[5] A balance must be sought between egoism and altruism.

An agent can help another agent get to the charging station by turning additional light on, so that the other agent has better chances to reach the charging station through phototaxis. On the other hand, when an agent does not want another agent to come nearer (and thus risk that it will be pushed out) it can turn off the light on the charging station, thus hiding the charging station for the approaching agent. This way an agent can help another agent find the charging station, and behave "altruistically", but it should at the same time worry about its own self-preservation and so occasionally behave "egoistically".

These five characteristics in many ways parallel similar situations in nature. Seeking out the charging station can be compared to foraging, charging to feeding, and damping the box-lights to anti-parasitic behavior. There is in addition the kind of cooperation we find between individual animals, such as between two birds that are keeping eggs warm on a nest and occasionally have to go out for food [6]. These situations have been well studied in ethology and part of the case study is to make explicit

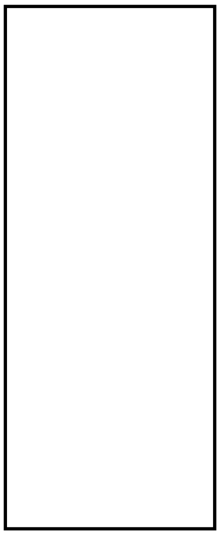

Figure 3: Hardware layout of the robot used in the experiment. The sensors are two pairs of photosensitive sensors, 6 infrared sensors, 4 bumper sensors, a loudspeaker, and a sensor measuring the energy level in the battery. The actuators are a left and right motor and a loudspeaker.

comparisons with known properties of animal behavior [7].

3 The behavior systems

The robot used for the experiment has the hardware characteristics outlined in figure 3. We use a dynamical systems approach to the programming of the behavior systems using a programming language PDL designed for this purpose [9]. The C-based implementation of PDL is used. PDL supports a set of quantities (such as RightFrontIR for the measured reflection of infrared reflection or WeakenCompetitor for the 'drive' to seek out competitors). The quantities are frozen at the beginning of a time cycle. The value of a quantity q can then be obtained by the form `value(q)`. Various processes can add to the value of the quantity at the next time cycle. This is done using the procedure call `AddValue(q, v)` which adds the value v to the quantity q. At the end of the cycle, each quantity takes on the sum of all the added values. The new values are sent off to the actuators and new sensory quantities are read in. The PC-compatible processor used in the experiment is able to execute 20 cycles per second for the complexity of the programs discussed in this paper. Integer arithmetic is used for processor efficiency reasons. The range of values for sensors and actuators is between 0 and 255.

3.1 Overview of Behaviors

A behavior-oriented design starts by identifying desirable behaviors and then seeking the subset for which behavior systems need to be developed. In the present case the following behavior systems are needed:

- Forward movement: The robot moves forward at a default speed.
- Touch-based obstacle avoidance: The robot touches an obstacle, retracts and turns away.
- Smooth obstacle avoidance: The forward path deviates as the robot comes in the neighborhood of an obstacle.
- Blue phototaxis: The robot is attracted to blue light.
- Halt while recharging: The robot stands still as it is drawing current from the charging station.
- Yellow phototaxis: The robot is attracted to yellow light.

The behavior systems realizing these behaviors in interaction with the environment are all active in parallel. Several of these behaviors will not be observed together because they contradict each other. For example, retraction (in touch-based obstacle avoidance) contradicts moving forward. The regulation which behavior is visible depends mostly on the environment as perceived by the sensors. An exception is the choice between blue and yellow phototaxis for which a motivational system is used, as explained later.

The different behaviors implement the various functionalities (translated to goals in classical AI) needed in the experiment. For example, blue phototaxis implements 'go to charging station', assuming that blue light is mounted on the charging station. Sometimes a functionality emerges from the interaction of different behaviors. For example, yellow phototaxis together with touch-based obstacle avoidance implements 'dim out competing light' in the presence of a box with a light mounted in it. Blue phototaxis together with touch-based obstacle avoidance implements 'drive into the charging station' in the presence of the charging station. A major point of this paper is that the design of the mechanisms should not focus on functionalities (and definitely not on goals) but on behaviors.

The rest of this section documents the different behavior systems. For some behavior systems, minor details have been left out due to space limitations.

3.2 Forward movement

Forward movement is implemented through a stabiliser [10] which adds or subtracts part of the difference between the current speed and the default speed. As a result speed always moves progressively back to the default. The default speed may be different for the left and right motors because motors are not necessarily equal. In the present case the default is set at 150.

$define\ DefaultRightSpeed\ 150$
$define\ DefaultLeftSpeed\ 150$
$AddValue(LeftSpeed, -(LeftSpeed - DefaultLeftSpeed)/10;$
$AddValue(RightSpeed, -(RightSpeed - DefaultRightSpeed)/10;$

3.3 Touch-based Obstacle Avoidance.

Two behavior systems (in parallel) ensure the obstacle avoidance competence. The first one is based on the bumper sensors. The second one uses the infrared sensors. Infrared-based obstacle avoidance is more efficient because it avoids bumping into obstacles, but it is less reliable because infrared reflection gives only approximate information about the presence of obstacles. Obstacles sensed in the back should also be avoided because occasionally the robot may bump into obstacles as it is moving backwards or it may have to move away when another object hits the back.

Touch-based obstacle avoidance is accomplished through a disturber [10] which increases speed in the opposite direction and causes a rotation away from the touch location (to the left when touched left and to the right when touched right). On robots with translation and rotation motors, touch-based obstacle avoidance would be implemented by influencing the speed of the translation and rotation motors. In the present case, we have a right and left motor so rotation is implemented by introducing a difference between the right and left motor speeds.

The influence from touch-based obstacle avoidance on the motor speed must be sufficiently strong to make the effect of normal forward movement ineffective. In a cooperative architecture, the forward movement behavior system remains active at all times. In a subsumption architecture, the obstacle avoidance behavior system would inhibit the forward movement behavior system.

$define\ Retract; 300;\ defineDeltaRetract; 100$
$define\ Jump\ 200;\ define\ DeltaJump; 200$
$/*\ 1.\ When\ left\ front\ bumper\ touched\ */$
$AddValue(LeftSpeed, -Retract * LeftBumper);$
$AddValue(RightSpeed, -(Retract + DeltaRetract)$
$|\quad * LeftFrontBumper);$
$/*\ 2.\ When\ right\ front\ bumper\ touched\ */$
$AddValue(LeftSpeed, -(Retract + DeltaRetract)$
$|\quad * RightFrontBumper);$
$AddValue(RightSpeed, -Retract * RightFrontBumper);$
$/*\ 3.\ When\ left\ back\ bumper\ touched\ */$
$AddValue(LeftSpeed, Jump * LeftBackBumper);$
$AddValue(RightSpeed, (Jump + DeltaJump)$
$|\quad * LeftBackBumper);$
$/*\ 4.\ When\ right\ back\ bumper\ touched\ */$
$AddValue(LeftSpeed, (Jump + DeltaJump)$
$|\quad * RightBackBumper);$
$AddValue(RightSpeed, Jump * RightBackBumper);$

Side effects:

[1] The inverse translation progressively decreases because the forward movement behavior system brings the speed back to the default.

[2] When the robot is touched in the front and the back simultaneously, it will not make any change in movement because the influences cancel each other out.

Figure 4 illustrates this behavior. We see that there is indeed backward movement and a turning away to the left. The figure is taken with a camera from the top of the robot arena. A lamp is mounted on the robot and the trace is produced by filtering the images coming from the camera in real time.

3.4 Smooth Obstacle Avoidance.

Smooth obstacle avoidance is implemented by the creation of a repelling force field, similar to a potential field [1], based on the measured infrared reflection. This influences the motor dynamics in such a way that when the infrared sensors on the left side (LeftFrontIR and LeftSideIR) are low (meaning obstacle approaching), there is a turning away to the right and when the infrared sensors on the right side (RightFrontIR and RightSideIR) are low, there is a turning away to the left. Low MiddleFrontIR causes inverse translation.

$AddValue(LeftSpeed,$
$|\quad value(RightFrontIR)/2 + value(MiddleFrontIR) +$
$|\quad value(RightSideIR)/4 - value(LeftFrontIR)/2$
$|\quad - value(LeftSideIR)/4 - IRBackground);$
$AddValue(RightSpeed,$
$|\quad value(LeftFrontIR)/2 + value(MiddleFrontIR) +$
$|\quad value(LeftSideIR)/4 - value(RightFrontIR)/2$
$|\quad - value(RightSideIR)/4 - IRBackground);$

IRInfluence is equal to the normal background infrared so that there would be no impact from IR-based obstacle avoidance unless the perceived IR is different from normal IRBackground.

Figure 4 illustrates this behavior. We see that the robot now turns away from the wall instead of bumping against it.

As with many sensors, there is a need for adaptive behavior. When there are a lot of obstacles in the environment, there is so much infrared reflection that the agent will move away from the area. This is like us being blinded by light to which our eyes adapt by letting less light onto the retina. A similar mechanism has been introduced that brings down the amount of IR that has been emitted, and brings it back up when relatively little IR is perceived.

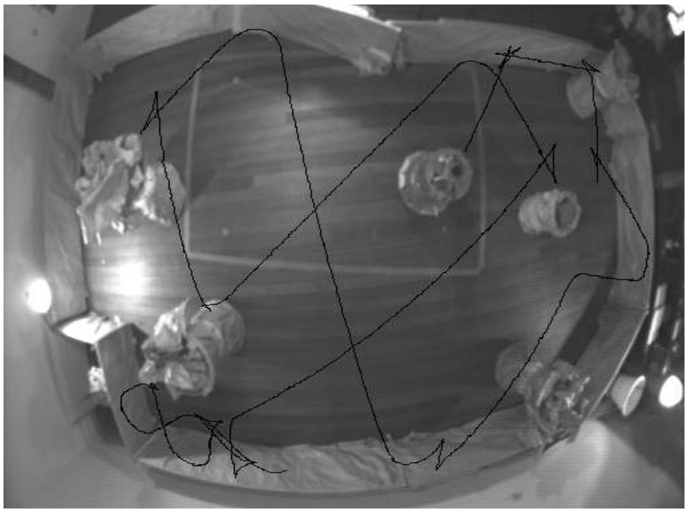

Figure 4: Touch-based obstacle avoidance and infrared-based obstacle avoidance. Both are executed in parallel with forward movement. After a period, the robot moves away from an object it senses through the infrared sensors instead of bumping into it.

3.5 Blue phototaxis (Attraction to the charging station)

Phototaxis is achieved by the creation of an attracting force field which influences the motor speed dynamics so that the robot turns right when there is less blue light on the left side and left when there is less blue light on the right side. The deviation is determined by subtracting the left and right yellow photosensor quantities and multiplying by a parameter which determines the weight of the influence.

$AddValue(RightSpeed, PhotoFactor *$
$|\quad (value(BluePhotoLeft) - value(BluePhotoRight));$
$AddValue(LeftSpeed, PhotoFactor *$
$|\quad (value(BluePhotoRight) - value(BluePhotoLeft));$ Side effects:

[1] A regular zig-zag behavior is typically observed (see figure 3.7). This zig-zag behavior is not explicitly programmed (for example by specifying that there must be for a while forward movement then left turn to a certain angle, then again forward movement, etc.) but follows from the interaction between the agent's internal dynamics and those of the environment, in particular the changing position of the agent with respect to the charging station.

[2] The robot will end up between the walls of the charging station (and thus ready for charging) due to the interaction between the obstacle avoidance and phototaxis behaviors (figure 3.7). As the robot is attracted to the light, it moves in on the charging station. If it bumps into the side of the charging station, it will move backwards and then try again. This is a very clear example of an emergent functionality from the viewpoint of the total system. The parking behavior has not been programmed explicitly (although it could be), but it nevertheless occurs reliably.

3.6 Halt while recharging.

Recharging takes place when the robot is located in between the two disks of the charging station and when current is flowing to the batteries to recharge them. Recharging starts up automatically as soon as the charging rods mounted at the top and bottom of the robot make contact. EnergyInflow is not directly sensed, but the robot can determine the EnergyAvailability. The quantity EnergyInflow is then based on comparing the change of EnergyAvailability over a time period. The EnergyInflow is a function of the energy available in the charging station and the energy already stored in the batteries. As batteries near completion, less energy is drawn.

The robot must not move forward when energy is flowing in. We therefore need an additional influence on the motor speed related to the availability of energy determined by testing whether there is a positive rate of change in the battery charge.

$AddValue(RightSpeed,$
$|\quad -(DefaultRightSpeed * EnergyAvailability);$
$AddValue(LeftSpeed,$
$|\quad -(DefaultLeftSpeed * EnergyAvailability);$

Side effects:

[1] As the robot is charging, the batteries will become fuller and less energy will be drawn from the charging station. EnergyInflow will decrease and the default forward movement influence on the motor speeds will take over. Consequently the robot will leave the charging station.

[2] If the available current is reduced as a result of competition from the box-lights, EnergyAvailability decreases. If competition for the energy from the lights increases, less current will be available from the charging station and EnergyInflow decreases. As a result, the robot will automatically leave the charging station.

3.7 Yellow phototaxis (seek out boxes) and revised blue phototaxis.

A motivation is an internal state which is a function of sensed aspects of the environment and the internal state of the robot. It influences the occurrence of certain behavior by being a parameter to the dynamics. Motivations can either be explicit quantities, or, the quantities influencing the motivation can be directly used in the dynamics of behaviors which are sensitive to the motivation. We need two motivations in the present case: (1) A motivation for energy (comparable to hunger in ani-

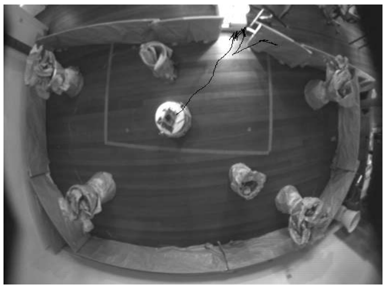

Figure 5: The figure shows phototaxis towards the light source mounted on top of the charging station. A characteristic emergent zig-zag behavior is observed. Notice also the parking behavior which emerges from phototaxis and touch-based obstacle avoidance.

mals). This motivation should increase as the amount of energy available in the robot decreases. (2) A motivation to avoid energy drain. This motivation should increase if the amount of energy available from the charging station is low and a charging robot is still in need for more energy.

The energy motivation is directly determined by the availability of energy, measured and translated to the quantity EnergyDebt. This should play a role in the strength with which the phototaxis behavior occurs. The EnergyDebt decreases due to charging. The motivation to stop the drain is more complex. It should be a function of the energy debt in the batteries and the energy inflow. It should strengthen (yellow) phototaxis behavior and decrease as the agent manages to put out the box-lights.

First the blue phototaxis behavior is extended to become sensitive to energy availability:

$AddValue(RightSpeed, PhotoFactor *$
| $(value(BluePhotoLeft) - value(BluePhotoRight))$
| $/EnergyDebt);$
$AddValue(LeftSpeed, PhotoFactor *$
| $(value(BluePhotoRight) - value(BluePhotoLeft))$
| $/EnergyDebt);$

EnergyDrain is a motivational quantity, regulated as follows:

$AddValue(EnergyDrain, EnergyDebt * EnergyAvailability);$
$AddValue(EnergyDrain,$
| $YellowPhotoLeft * YellowPhotoRight *$
| $(LeftBumper + RightBumper));$

The EnergyDrain quantity is then used in determining the tendency for going towards the boxes and dim out the lights:

$AddValue(RightSpeed, PhotoFactor *$
| $(value(YellowPhotoLeft) - value(YellowPhotoRight))$
| $/EnergyDrain);$
$AddValue(LeftSpeed, PhotoFactor *$
| $(value(YellowPhotoRight) - value(YellowPhotoLeft))$
| $/EnergyDrain);$

Side effects:

[1] As the robot is attracted to the yellow light, there is enough momentum that it will bump into the box, thus causing the light to be dimmed. The bumping causes touch-based obstacle avoidance and therefore retraction, but because the is still causing phototaxis, a second approach takes place, etc.

[2] Bumping stops when there is no longer yellow light emmitted by the lamp associated with the box. It also stops when the EnergyDrain quantity has become so low that infrared-based obstacle avoidance becomes stronger.

4 Discussion

The complete ecosystem with all the behaviors described in the paper has been implemented and has been demonstrated to operate. The system works with one agent as well as two. Observers unfamiliar with the design interpret the behavior as cooperative although no explicit cooperation has been programmed. Extensions of the behavioral repertoire have been designed and implemented so that agents can engage in communication and explicitly 'decide' to leave the charging station as a response to a request from another agent. This leads to a more efficient exploitation of the available energy. These extensions will be discussed in another paper.

The case study illustrates some of the design guidelines published earlier [2]:

- *The physics of the agent and the ecosystem is exploited.* For example, the momentum from forward movement enforced by yellow phototaxis causes the robot to bump into a box and thus dim the light. There is no 'dim the light' behavior system needed.

- *The total design exploits the interaction between internal and external dynamics.* For example, when the agent is charging there will be internally an increase in energy and when the competition from the lights increases there will be a decrease in energy at the charging station. The agent does not model these physical processes but its internal dynamics (e.g. as in the halting behavior system) is in concordance with it.

- *Emergent behavior is explored whenever possible.* For example, there is no explicit parking behavior. It emerges from the interaction between blue phototaxis and obstacle avoidance.

The case study also explores some novel design principles:

[1] *We use a cooperative as opposed to a subsumption architecture.* Our approach has been to progressively consider different behaviors, each time adding more

mechanisms to achieve more competence. This methodology has been recommended by several researchers, in particular Brooks [2]. But we see additional behavior systems cooperating rather than subsuming existing behavior systems. A particular behavior system may never inhibit the in- or outflow of information to another behavior system and the effects of different behavior systems are always summed. In this sense, the architecture is a cooperative as opposed to a subsumption architecture.

[2] *We use parallelism as opposed to action selection.* Many researchers assume that the overall activity of the agent has to be split up into different, mutually exclusive actions and that consequently there is an action selection mechanism necessary which selects what action is the most appropriate at a particular point in time [5]. Instead, we work from the hypothesis that behavior systems exert a continuous influence on the actuators by a large set of parallel processes. The influences are summed. As observers we sometimes see only one action and not others but this is the consequence of properties in the environment. For example, retraction and turning away, which is part of touch-based obstacle avoidance, will not be visible if there are no obstacles. However, the action of doing obstacle avoidance is never explicitly selected. It is always there. The same is true for all the other 'actions' observable in the experiment.

[3] *We perform a behavior-oriented as opposed to a goal-oriented design.* There is a long tradition in AI to perform design by identifying goals, identifying actions that can satisfy those goals, and by then refining the analysis in terms of preconditions and postconditions. Existing planning systems all operate with this abstraction. Also in more recent work on reactive agents, a goal-oriented analysis has been proposed [8]. Instead we use a behavior-oriented design approach. The different needed observable behaviors are identified. These behaviors have as a side effect that certain goals will be achieved (if one insists on a goal-oriented analysis). For some behaviors, a behavior system is developed which establishes the desired behavior in continuous interaction with the environment. This behavior system is integrated with already existing behavior systems (to ensure that the mutual influences are compatible) and it is always active.

[4] *Control programs are based on dynamical systems as opposed to algorithms.* There are no sequential steps, control flow in the form of goto's, conditional statements, timers, etc. Instead each behavior system establishes a continuous link between a set of quantities and a set of other quantities.

The case study demonstrates that all these design principles lead to robust working systems. Larger scale experiments are needed to see whether they scale up and whether additional design guidelines are desirable.

5 Acknowledgement

The experiment discussed in this paper is a team effort at the VUB AI Lab. David McFarland has played a crucial role to make the experiment in tune for ethology. Several engineers and computer scientists have participated in the design of the hardware of the robot and the ecosystem. The most important role in hardware construction has been played by Danny Vereertbrugghen. Filip Vertommen has been responsible for the PDL version used in the experiment. Peter Stuer has been the principal designer for the behavior systems and the global ecosystem. Geert Machtelinck has made important contributions to the specific behavior systems reported in this paper. Research has been funded by the ESPRIT basic research project SUBSYM (particularly some of the early hardware development) and by the Belgian Government IUAP Centre of Excellence grant to the VUB AI lab.

References

[1] Arkin, R. (1989) Motor Schema based mobile robot navigation. Int. Journal of Robotics Research. Vol 8, 4 p. 92-112.

[2] Brooks, R. (1991) Intelligence without reason. Proceedings of IJCAI-91. Morgan Kaufmann, San Mateo Ca. p. 569-595.

[3] Jones, J.L. and A.M. Flynn (1993) Mobile Robots. Inspiration to implementation. A.K. Peters, Wellesley Ma.

[4] Donnett, J. and T. Smithers (1990) Lego Vehicles: A Technology for studying intelligent systems. In: Meyer, J-A., H.L. Roitblatt, and S.W. Wilson (1993) From Animals to Animats2. Proceedings of the Second International Conference on Simulation of Adaptive Behavior. MIT Press/Bradford Books, Cambridge Ma. p. 540-569.

[5] Maes, P. (1989) The Dynamics of Action Selection. In: Proceedings of the 11th International Joint Conference on AI (IJCAI 89) Morgan Kaufmann, Pub. Los Altos. p. 991-997.

[6] McFarland, D. (1990) Animal behaviour. Oxford University Press, Oxford.

[7] McFarland, D. (1994) Towards Robot Cooperation. [submitted to SAB1994].

[8] Nilsson, N. (1994) Teleo-reactive programs for robot control. In: Journal of AI Research. [to appear]

[9] Steels, L. (1992b) The PDL reference manual. VUB AI Lab memo. 92-5.

[10] Steels, L. (1993) Building Agents with Autonomous Behavior Systems. In: Steels, L. and R. Brooks (eds.) (1993) The 'artificial life' route to 'artificial intelligence'. Building situated embodied agents. Lawrence Erlbaum Associates, New Haven.

[11] Steels, L. (1994) The artificial life roots of artificial intelligence. Artificial Life Journal, Vol 1,1. MIT Press, Cambridge.