A DESIGN FRAMEWORK FOR SOCIETIES OF AUTONOMOUS ROBOTS

Elpida S. Tzafestas
LAFORIA-IBP
Université Pierre et Marie Curie
4, Place Jussieu
75252 Paris Cedex 05
FRANCE
brensham@laforia.ibp.fr

Abstract

This paper describes work on design principles for autonomous robots societies. The design problem is formulated as a decision problem where an optimisation formula is applied to a set of design criteria, each one being an evaluation of input task specifications across design alternatives. A three-level-nested cellular architecture for autonomous robots that decouples action from reflex as well as from arbitration is briefly exposed and the experimentation and evaluation context necessary for design parameters identification presented. Investigations on an exploration task reveal that the design knowledge sought should be of statistical nature and two identified design/control parameters are found to be the adaptation mechanism and the time. Those observations, together with the introduced concept of operational cycles, trigger a discussion on the overall design problem, where a unified model of recursive self-regulation is proposed and the suitability of the cellular architecture for it is underlined.

1. Introduction

This paper describes work on design principles for autonomous robots societies. The long-term goal of the project is to automate the design process. To this end, various needs have been identified:

Defining the design framework. What product do we expect design to yield? What are the principles on which design is based? In the next section, the design problem is formulated as a decision problem where an optimisation formula is applied to a set of design criteria, each one being an evaluation of input task specifications across design alternatives, i.e. robot specifications. The design problem is thus split into a principle and a model or knowledge subproblem: The first is that of translating input task specifications into design criteria and combining them to a global metric that will be used for optimisation, i.e. for choice of the best design alternative. The second is that of defining each criterion as a function of the possible output robot specifications in question. A description of the structure proposed as a solution to the first subproblem and which is to be filled with experimental or other data is also given. The application domain chosen is the field coverage tasks domain which has the properties of including instances of variable task complexity, as well as of imposing explicit performance metrics.

With the above in mind, one then needs to adopt *a robot control system architecture* that lends itself to specification construction as well as parameter instanciation, while demonstrating a degree of extensibility. Among the existing robotic control system architectures, there is hardly anyone that doesn't claim extensibility. The issue of architectural extensibility and its relation to arbitration schemes is treated in (Tzafestas 1994b). However, there has been so far little concern on specification construction and parameter instanciation. So, in section 3, a cellular control architecture is presented which is claimed to fulfill all these requirements and hence enable massive experimentation for different parameter settings, which are in turn the design alternatives. Its main tenet is that of decoupling action from reflex as well as from arbitration. The relation of the proposed architecture to neighbouring approaches, mainly subsumption and connectionism, is also briefly sketched.

Next, one needs to define the *experimentation and evaluation context*. In section 4, the context corresponding to the field coverage problems is exposed and a simulation environment that was built as a vehicle for those experimentations described. Emphasis is again put on the definition and use of specifications rather than complete descriptions of the robots' control system, as well as on the necessity of an adequate observation mechanism.

In section 5, some results on a set of examples drawn from the well-known space exploration paradigm are presented. Those results suggest that the kind of knowledge we are seeking is of statistical nature and concerns the individual or combined effects of certain parameters. Two such parameters identified are the adaptation mechanism and the time. The explicit use of time as a design/control parameter also suggests that to control a society, we need to identify its proper operational cycles.

The above trigger a discussion on the overall design problem: The idea to optimise is found to be spurious and the concept of bounded optimality arises which in turn leads to asking for a unified design principle and model. Self-regulation is then proposed as the phenomenal law at every level of a robotic society and it is argued that a closed motivational space for individual robots is necessary to support such a law in the micro level. At last, in section 7, conclusions are drawn and future perspectives mentioned.

2. Toward autonomous robots societies design automation

Research in autonomous robots societies is in an intermediate stage between initial enthusiasm and systematisation. It has been shown that putting multiple autonomous, i.e. unsupervised, robots together can lead to astounding global results and performance (see for instance, Mataric 1992a) and the power of the approach has been attributed not to the individual robots possibilities, but to the versatility of robot-robot and robot-world interactions (Brooks 1991a).

The work underway precisely attempts to identify a set of principles for systematising and, in the future, automating the design of societies of autonomous robots and focuses mainly on individuals' local structures and capabilities (which, of course, account for any phenomena observed). In an attempt to proceed in both directions, top-down and bottom-up, from population tasks to individual robots and vice versa, a top-down decomposition strategy drawn from industrial contexts was adopted (Tzafestas 1991)[1]. Design process is the process that stands between input of generalised performance specifications and output of actual product specifications. For our autonomous robots society design purposes, product specifications are the number, types and individual specifications of robots (sets of parameter-value pairs), while generalised performance requirements are all requirements on the society's functionality and structure, including environmental hypotheses, cost limits etc.

The structure of the input specifications is given in fig. 1. Functions and constraints are requirements that decompose into a set of quantitative criteria. Functions may generally decompose further into subfunctions and are different from constraints in that a designer is committed to the latter ones, while it retains relative freedom for the former ones. The criteria correspond to measures of input specifications satisfaction and are

---

[1] This work built on top of an implicit notion of a design problem space (Goel & Pirolli 1989) by adding structural parameters to the usual functional specification model and defining the latter one as a special-purpose hierarchy.

expressed as functions of the individual robots and the task's parameters. To design then means to choose such a parameter setting among all possible ones, or from another point of view to tune the robots parameters to such a set of values that best satisfies those criteria. Hence, the design problem splits to two subproblems: one of translating input specifications to design criteria and combining them to a global metric which will be used for optimisation, and one of defining each criterion as a function of the possible output specifications in question. We call the first subproblem *the design principle problem* and the second one *the design model or design knowledge problem.*

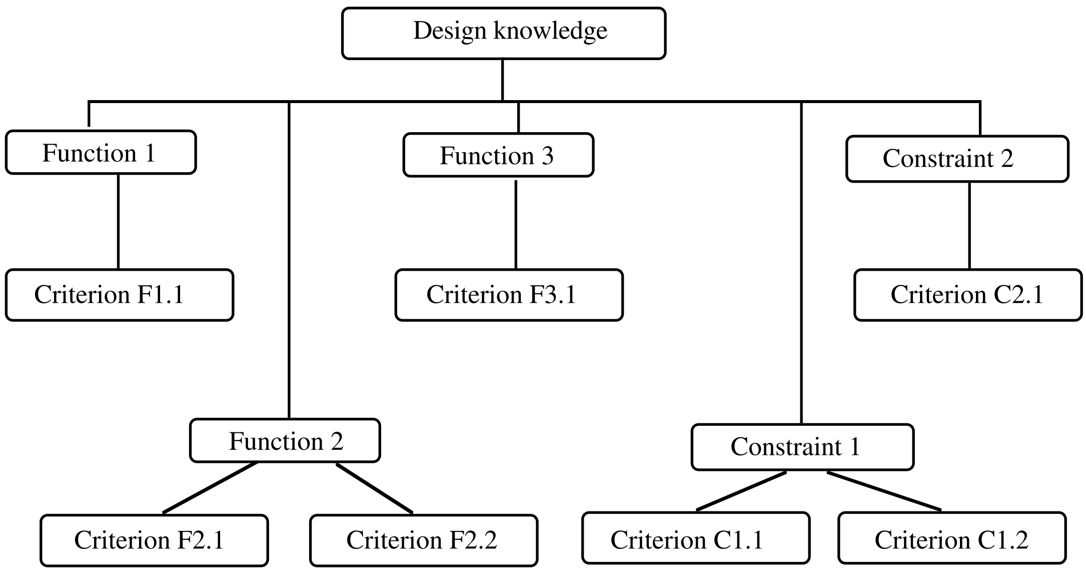

Fig. 1 : The design model structure

The optimisation formula used is simply a discrete functional over the set of criteria:

$$U = \Sigma\, w_i\, u_i \qquad (1)$$

where the weight w_i is the degree of importance a designer assigns to the criterion-i and u_i its utility, i.e. a measure of profit or proximity to its optimal value. For the formula to work, we need also assume that

$$\Sigma\, w_i = 1 \qquad (2) \qquad \text{and}$$

$$0 \leq u_i \leq 1 \qquad (3)$$

which gives $0 \leq U \leq 1$. The designer's goal is to maximise U. Criteria that belong to constraints are generally interval expressions and do not participate in (1)[2].

[2] In (Tzafestas 1991), to each criterion I had associated a measure of importance and a cost and I had pointed out that "speaking about importance measures makes no sense if one does not consider the corresponding costs: The higher the importance of a function, the higher the price a designer is ready to pay to implement it, while high cost for a low importance function might under certain circumstances lead to its suppression" (adapted from pages 30-31). However, since I didn't do any optimisation then, I didn't realise the necessity of reasoning in terms of profit (utility) rather than cost. If in (1) and (3) the

Contrary to what is advocated in (Carle et al. 1994, p. 5), the present method does not rely on a top-down distribution of a global evaluation function over the entities involved, but rather on a bottom-up propagation and synthesis of it out of local criteria. The justification for this method appears in (Tzafestas 1991, pages 34-36). Top-down distribution can only have the sense of assigning local bounds to the designer's freedom and these will be overlapping, otherwise there wouldn't be any need to distribute, in the first place! The analysis can therefore solely be based on bottom-up propagation and this propagation captures the nature of cooperative -or, most usually, conflicting- requirements.

The parameters that participate in the criteria are all those quantities that affect performance, or, better, all those *observable* quantities that affect performance. Only a few of them will be liable to tuning and hence controllable by the designer.

To identify those parameters that are crucial for design, we decided to proceed by performing massive computer simulations for different parameter settings. The kind of knowledge we expect to acquire is the *effect of individual parameters* on global performance, as well as the identification of *parameter interactions* (an example of interactive parameters is given in section 6). For this goal of identification, simulation appears to be the only way to quickly and consistently evaluate global performance for a variety of parameter settings. However, attention must be paid as to the portability of the architecture on real hardware as well as to the operational assumptions, such as representation of the world by the simulator. We hope that the strong structural character of the control architecture adopted (see section 3) keeps us away of illusions as far as the basic capabilities of a robot are concerned, and careful design of the simulation environment (see section 4) prevents any operational simplification to mislead us in erroneous evaluation. For instance, identification is a too high-level goal for accurate sensor/actuator operation and hence introduction of noise to become imperative, as in (Harvey et al. 1992).

An essential hypothesis behind this line of research is that sole structural parameters can account for any global behaviour and performance, so that we need to adopt a robot control architecture that allows systematic exploration of *structural variants*. More specifically, while residing in the paradigm of reactive behaviour-based robots (Brooks 1986, Arkin 1987), the stance taken is that of a need for classification of structural components into various types as well as classification of component interactions (see also section 3), so that the focus of study is the relation between those types and the performance observed. This also implies we don't believe in general purpose architectures, but in architectures amenable to parameterisation, which are suitable for classes of applications, i.e. for which classes of applications they yield acceptable, or else adequate, results. We call this the *structural stance* and in section 5 it is precisely shown how tuning of certain local parameters in the robots may have drastic impact on global performance. Of course, finally, global behaviour is a result of interaction between robots and the world, so that the researcher's role is to identify those structures that better fit the world, i.e. those structures that represent naturally the parameters that are crucial to global performance, and this can only be done by identifying the *laws* of this structural coupling (Maturana & Varela 1987, chapter 4) between the robot and the world.

The application domain chosen for this work is the *field coverage tasks* domain, i.e. the domain of robotic tasks that presume maximal spatial coverage as the goal of the robotic society. The justification of this choice is twofold: First, maximal spatial coverage is a desirable property of a large variety of tasks, such as exploration and its variants (sweeping, sampling etc.). This view of tasks is also expected to allow us to naturally

---

u_i's represent costs (c_i), we arrive at optimising by minimising U -which now expresses total cost- while using *exactly* the same knowledge. However, using the costs, the above relation between cost and degree of importance is nowhere represented. On the contrary, defining u_i as profits and using (1)-(3) biases selection of alternatives with high w_i *and* u_i, favouring therefore functions of high importance.

proceed to more knowledgeable tasks in the industrial and services domain, like supervision and inspection, which on top of exploration add some extra requirements on prescribed activities and unexpected events servicing. And second, there is a way to evaluate the spatial coverage by observing the actual portion of the environment covered during a simulation, and this is a clear numeric performance metric, which in combination with other metrics may be used to compare different parameter settings for the robotic society.

Finally, formulating the design problem as above, implies that the designer's role -that of specifying a society of robots for a task at hand- is in fact to predict to a degree the robots' interactions with each other and with the world so as to assure the successful completion of the task. His problem then may be seen as *a control problem*: How to control the robotic society, so that its operation converges to the desired result. Clearly, since the exact structure of the world is not known in advance, this is an ill-defined problem, so that the approach taken by the researchers is to equip the robots with an adaptation mechanism that will make them more responsive and more fit to the environment they find themselves in. Hence the design problem translates to: How to equip *properly* the robots so that they interact with the world in a way which will make them converge to the desired result, or what adaptation mechanisms should be provided so that the desired result is assured for the class of worlds under consideration. The design problem then becomes that of controlling adaptation of robots, i.e. *adaptation is the main design parameter* -or better, control parameter- at hand, where, given the structural stance taken above, adaptation stands for the self-tuning of oneself's parameters according to the feedback from the world. Some representative examples may be found in section 5.

3. A cellular robot control architecture

With the above in mind, one then has to decide on a robot control system architecture that will enable extensive experimentation and exploration of the parameter space. An initial choice was to remain in the reactive, behaviour-based paradigm which was shown (Brooks 1991a) to have many desirable properties: no central world representation bottlenecks, minimisation of time lags between perception and actuator level command output, fault-tolerance and low implementation cost. In one word, the power of the behaviour-based paradigm lies in its simplicity. It is left to the interaction with the world to give rise to complex phenomena, and it is this potential emergent complexity that researchers in this area try to tame and exploit.

To allow massive experimentation, as was set forth before, one has to find a means of systematisation of study. Given the structural stance taken as well as the design framework adopted, the architecture then needs to fulfill the following requirements:

1) Extensibility. One needs to be able to easily add new structural possibilities, either components or connections, and study their individual impact as well as their interactions.

2) Specification construction possibility. One needs to be able to define a one-to-one relation between an instance of the architecture and a specification. During design it is the specification that will be manipulated, so that it must be ensured that this specification induces no ambiguity on the actual structure of the robot's control system.

3) Parameter instanciation possibility. Finally, one needs to explicitly define the parameters that constitute this specification and whose impact one intends to study. Admittedly, not all of these parameters will be of interest in different contexts. The goal of the comparative study is then to discover those that actually play a role and translate this role into what was previously defined as the design model.

The architecture that was developed and which tries to overcome certain limitations of previous approaches, while fulfilling all of the above requirements is a cellular control architecture. Its main points are summarised here, while a more complete description can be found in (Tzafestas 1994a):

1) *The cell is the elementary unit of storage and processing.* A cell represents and corresponds to a piece of hardware capable of some minimal storage and processing. It has a number of input connection lines, a number of output connection lines, a static or dynamic memory vector and a processing unit. Cells are therefore structural primitives differentiated by design or by evolution in a specific context and represent a uniform substrate for building complex aggregates. Inter-cell connections have a control and a data part. The cell's complexity lies between that of a simulated neuron and an augmented finite state machine. Cells also possess an internal execution time which may or may not be controllable by the designer. The use of this internal time as a control parameter is shown in section 5.

2) *Cells are categorised into functionally different types*. There are three types of cells, *sensor cells*, *actuator cells* and *processing cells*. Sensor and actuator cells are responsible for primitive sensing or effector operations, respectively. Processing cells categories include: min, max and find-a-stimulus cells (that output respectively the lowest, highest or just the first non-zero input), time-out cells and priority cells (whose inputs are considered as statically or dynamically ordered and that output the highest priority input). There are also three special processing cell types: decoders, arousal cells and observer cells, whose roles are explained below. This conception of the cell differs radically from the ones found in cellular robotic systems (Wang 1990), where robots situated in space are regarded as cells and there is no adaptation mechanism, and in dynamic reconfigurable robotic systems (Fukuda & Kawauchi 1991), where cells have a complex internal structure and some communication capability, but there is no notion of organisation of intelligence, i.e. of the control system.

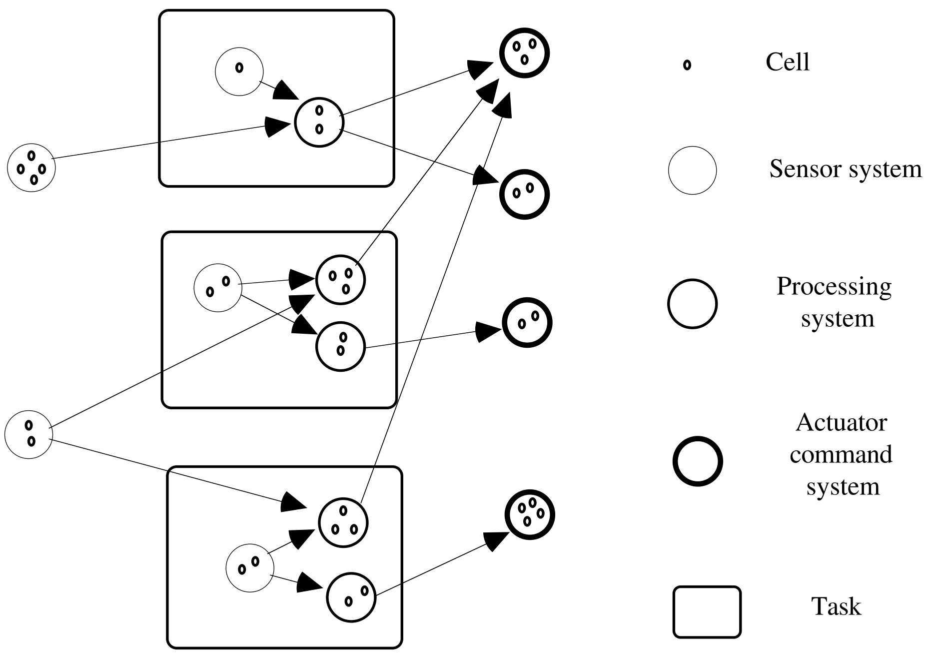

Fig. 2 : The cell network

3) *To manage complexity, the cell network has a three level nested structure* (see fig. 2). Cells are clustered to cell aggregates and aggregates are clustered to tasks (which correspond to what is generally called behaviours). Cell aggregates belong to one of four types: *sensor systems*, *actuator command systems*, *processing systems* and *arousal systems*. Sensor systems consist of a number of logically correlated not interconnected sensor cells and may be dedicated to specific tasks or shared. Actuator command systems are shared and their general structure is shown in fig. 3. The decoder cell combines input from tasks with input from reflex sensors and outputs the executable command to the corresponding actuator. In general, this command may take one of several values, so that the decoder need only make sure it isn't in conflict with any of the reflex inputs, which represent forbidden command values. Although the above is more a representation than a precise description of how reflexes work, it offers the advantage of **decoupling action from reflex**, i.e. of decoupling commands issued by the high-level tasks from low-level reflex commands. This is contrary to the other behavioural approaches found in the literature (see, for example, Brooks 1986 and Arkin 1987) and has as a direct consequence the dismissal of the need to subsume reflex level commands and hence to complexify the interfaces between components (here tasks). What is actually done in case of conflict between commanded action and reflex is open to the designer. For the case of navigation, we assumed that another valid command is chosen -if there exists one- which corresponds to robots being biased toward navigation. An alternative would be to stand still (do nothing) in case of conflict.

Finally, to every task is associated an arousal system, whose role is to activate it when fired (see point 4 below).

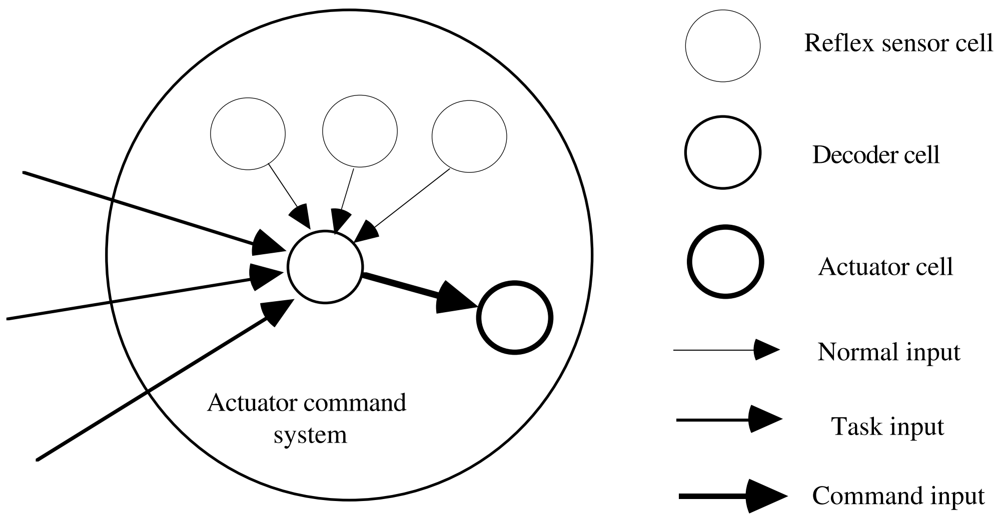

Fig. 3 : An actuator command system

A task thus consists of an arousal system, processing systems and dedicated sensor systems (if any). Processing systems are connected to sensor systems or other processing systems as input and actuator command systems as output. Tasks at this level should not be confused with society tasks (such as exploration). The same term was precisely used to underline that what is sought is the relation between individual tasks and society tasks. Finally, it should be stressed that *the cell level is the one and only execution level, while the aggregate and task levels are used solely for observation/evaluation and design purposes.* For an application domain considered, the types of usually encountered tasks must be identified: in section 5, and for the case of exploration, dispersion, foraging and homing are used as the main building blocks out of which several structural variants are defined.

4) *Arbitration is based on the principle of activation/inhibition*. Each time one particular task arises, it temporarily inhibits all the tasks connected to same actuator command systems as itself, so that an actuator command system receives only one non-zero task command at a time. As a result of the above, there is no explicit arbitration, but rather the overall behaviour of the robot is one of switching between tasks (activities) in response to the world it finds itself in. The arousal system structure is shown in fig. 4 and comprises an arousal cell and several observer cells. The arousal cell is a thresholded cell that fires whenever its activation level rises above an activation threshold. Whenever this happens the arousal cell sends inhibition signals to the mutually exclusive tasks. Total activation in the robot is constant and it flows from task to task in a manner indicated by observer cells whose role is solely to detect input presence. Presently used observer cells are either time-out cells or counter cells. Some provision has been taken for future integration of arousal systems whose activation is locally computed and independent of activation flow. The above scheme implies no interface between tasks, other than the flow of activation, which is nothing else, but *a measure of the robot's motivation to perform a given task*. Inter-task information fusion is thus minimal and occurs in the actuator command level. As a result and contrary to all previous approaches, arousal (selection equivalent) of a task is represented separately and does not interfere with the actual sensing and processing local to the task.

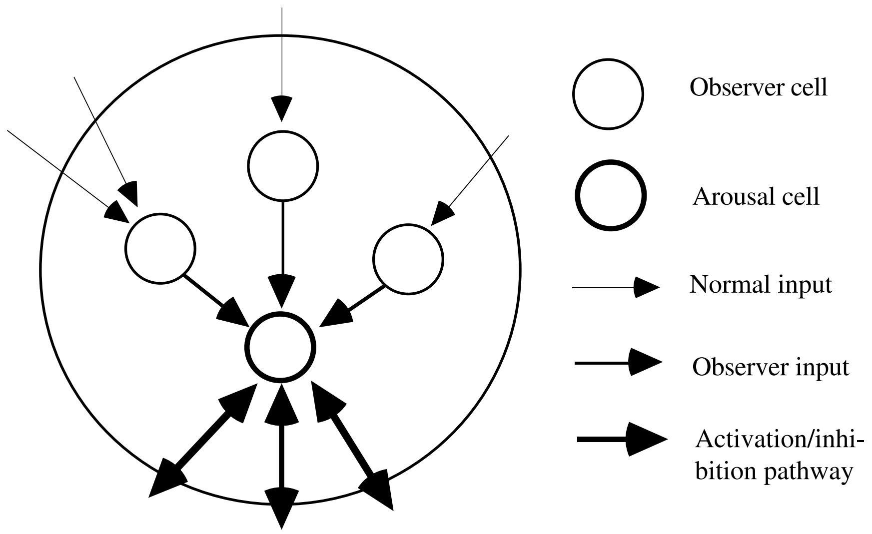

Fig. 4 : An arousal system

5) *Adaptation occurs by local reinforcement or replacement at the cell level*. The way the network is structured, adaptation may only occur at the cell level, though its effects will be indirectly observable at the task and at the robot's observer level. In principle, the whole internal memory vector of a cell is adaptable, either as a whole or in parts, through blind or informed reinforcement (term used to denote both increase or decrease of a value) or replacement (with a computed value, usually the cell's new output). Adaptation within the processing systems of a given task usually results in "smoother" individual task performance, while adaptation within a task's arousal system usually leads to more complex observable phenomena as far as the overall performance of

the robot is concerned. We call these two kinds of adaptation, *local adaptation* and *connection adaptation*, respectively, and they correspond to adaptive action and adaptive selection of action. An illustrative example is presented in section 5.

Discussion

The cellular approach to autonomous robot control systems was adopted as a more naturally structured descendant of existing behavioural control architectures (subsumption and variants) that suffered from unprincipled design of both individual layers and inter-layer interfaces. The idea is to dispose of a repertoire of primitive components (cells) and combine them in a proper way to build higher-level structures. This splits the problem into asking what is a basic set of primitive components and how are higher-level structures built out of them and is obviously in line with the non-conventional connectionism research directions of *modularisation* and *compositionality* (Chandrasekaran et al. 1988).

On the connectionist side now, it should be first understood that each cell is a point of uniform processing of inputs (every time some inputs need to be given special care, a new cell is added to the network). The decoders and the arousal cells are the exceptions to this rule: they correspond to the points where cyclicity appears in the cell network, the activities (tasks) structure being otherwise acyclic. Given this principle, it becomes obvious that *a cellular structure is just a typed connectionist one*, where the use of higher-level primitives (min, max etc.) than the ones presumed in traditional connectionist research, permits to preserve clarity of specification and analysis for given tasks, while considerably pruning structural complexity. The examples presented in section 5 show that the structure of each task reproduces a layered reactive algorithm, so that the organisation of the control system may thus be seen as a set of layered activities combined with each other in a connectionist way through activation/inhibition pathways. As explained before, decoupling of the reflexes from these activities, together with the feed-forward activity structure, dismiss the need to subsume layers inside a task; otherwise, the function of a task is equivalent to that of a subsumption chain.

All this being said, it appears that the interest of this architecture resides both in its strong structural character which allows the former design framework to apply, as well as in the fact of being inherently connectionist, which permits questions on adequation of activation and adaptation mechanisms to be raised. The extensibility issue is treated more in depth in (Tzafestas 1994b) where it is argued that *compositional extensibility*, as opposed to incremental, is what is necessary and that this fits better the designer's viewpoint of *modelling the multiple laws of interaction* between components instead of relying on potentially useful synergies.

Finally, and as far as specification and parameterisation are concerned, it is obvious that the control system of a robot can be completely specified by giving the types and specifications of all its components. In turn, each component is recursively specified in the same way, until the cell level which is by definition the lowest one is reached. An example task and its specification are shown in figure 6. The specifications have then the same structure as the components they correspond to, what forces the designer to only consider real structural parameters without falling into the trap of functional parameters not really present in the control system. Of course, since not all parameters are interesting for experimentation or design, the specifications usually considered are a reduced form of the complete specifications, as shown in figure 6 where actually only a few parameters for the foraging task are considered.

4. The experimentation and evaluation context

To conduct those massive experiments prescribed, a simulation environment[3] was built that takes into account the concern for *simulating structures' operation rather than observable functionality*.

The main hypothesis of the experimentation context was that the *world is a repository* of agents, objects and signals, whose evolution in time may be observed.

- *Agents* are all those entities that are engaged in some sort of activity and may respond to changes in their immediate environment[4]. Agents are thus different from objects in that they are dynamic or mobile or both. Simulated robots are represented as agents. Other agents present may include uncontrollable agents whose behaviour must be taken into account, but on which the designer/observer has no influence. This could be for instance the case of circulating vehicles among which a garage supervisor robot has to navigate. Agents may be simple or composite. Robots are in fact composite agents. The cellular architecture's three-level structure is reflected in the corresponding description language: robots are represented as agents that consist of sensor systems, tasks and actuator command systems, each task is represented as an agent that consists of an arousal system, dedicated sensor systems and processing systems and so on. Finally, the environment allows for cohabitation of different types of agents.

- *Objects* are static and passive and may be directly accessed by agents. Obstacles and sources of mineral are typical examples of objects in the exploration example.

- Finally, *signals* are "immaterial" objects. They may be perceived just like objects but they are not situated, that is they don't occupy any physical space.

This hypothesis implies there is no special provision for communication. In reality, communication is achieved by emission and perception of signals in the world. This constitutes *a unified framework of perception and communication*, as is also the usual approach in literature. An example from exploration again is the orientation signal a space base may emit for the explorer robots to perceive and home back to it.

The world is represented in the simulation environment as a closed, two-dimensional discrete grid[5], which is the most commonly encountered world representation (see, for example, Tyrrell 1993).

Time management, which is an aspect that never appears on simulation environments description, is based on a time-slice technique with some additional memory: each agent has an internal clock which represents the duration of its current

---

[3] The simulation environment was developed in Smalltalk-80, version 4.1, on a SUN workstation. In retrospect, however, it seems to me that Smalltalk-80, despite being a powerful representation language with a rich library of general-purpose classes and graphical interfaces development facilities, lacks all the processing power necessary for principled experimentation. Indeed, I haven't been able to quantify some of the experiments described in section 5 due to this limitation.

[4] An agent is an agent because an observer may attribute some intention to it.

[5] Using a discrete simulated world, however, gives rise to some artificial phenomena, in the sense that they wouldn't appear in their real world counterparts. For instance, I sometimes observe my simulated robots loop within a few places in the grid as a result of artificial synchronisation. In the real world, nothing is exactly accurate, so that I wouldn't care about loops. On the other hand, continuous space would only be represented by the simulator as a variable scale grid, and, as was demonstrated by a sample experiment I conducted, this would again lead to potential for artificial synchronisation, but this time in a much larger scale. Finally, I decided not to add explicit noise to get around this problem, because for the kind of knowledge I am seeking (see sections 5 and 6) this is not important and because whenever I did it the variance of some results could rise to an order of magnitude above the variance without noise, which means that those results couldn't be reliably evaluated.

"action". Every time this duration expires the agent executes. This technique is called the time-triggered execution technique, since it assumes that each agent is triggered by the elapse of time. Actually, each agent is triggered by what are defined as its inputs (for instance, a processing cell is triggered by the presence of valid signals in its input lines), so that this mechanism is an artificial representation of the time lag between data input and action output at the single agent level. The advantage of using this technique rather than a usual synchronous or pseudo-synchronous round-robin is then that it allows asynchronicity between the variable time-length actions of different agents while representing naturally the flow of signals from sensors to actuators.

Evaluation of a design alternative involves inspection of several aspects of the simulated world's state. To be able to evaluate once those aspects are identified, *an observation mechanism* was introduced. A number of special objects, called observers, are attached to all those entities (for instance agents) that are to be observed and operate as parasites during simulation. Once the simulation is over, the states of all observers are interpreted and collected to a statistics record of the simulation. Clearly, this constitutes an advantage of the simulation environment over real hardware robots: we can evaluate and we can do it without modifying neither the robots program nor the world, what would risk to bias the results obtained[6]. Furthermore, introducing an explicit observation mechanism forces a user to formalise the conditions of interest[7]. This would also constitute an adequacy criterion for an observation mechanism: whether it allows to formalise the conditions of interest and represent the results in a straightforward, easy to interpret way.

Finally, it is important to stress again the *use of specifications*, this time at all levels of the definition of what is a simulation to be performed. Namely, a simulation is run after being initialised on a set of specifications, for the world, the agents and the observers, respectively. The results obtained are then directly associated with the specifications to which they correspond. Specification and observation thus allow for *automation of the experimentation process*. For the environment user, specifications also imply a constrained freedom of intervention: in fact, a user wishing to intervene on an agent's operation, can only modify it through its specifications, which is equivalent to modifying its control system, or its program, while there is no possibility to modify the actual data it uses. As was pointed out in section 3, it is the role of reflexes to abort from an action whenever that data changes abnormally (for instance, when the user moves a robot to another place in the simulated world), and it is the role of the designer to equip an agent/robot with the appropriate reflexes, in terms of functionality and real-time performance, that will allow it to survive to such abnormalities.

5. Investigations on an example task : Space exploration

One application falling into the category of field-coverage tasks is space exploration. A number of robots are assumed to land on a region and be left to explore it. The robots operation consists of three phases: *dispersion phase*, *foraging phase* and *homing phase*. During the first phase robots disperse away from the space base and from each other, while during the second phase, which was named after its ethological counterpart, they actually explore by searching for what they are supposed to identify, sources of various minerals in our case. Finally, during homing they return to the space base, following an orientation signal emitted by it. It is supposed that during dispersion the robots navigate at relatively high speed with their mineral sensors off to save energy

---

[6] For example, Connell, in order to evaluate without biasing, had to video-tape the arm in action and then manually plot trajectories during slow-speed playback (Connell 1990, p. 73).

[7] For instance, (McLennan & Burghardt 1994) measured cooperative communication in their simulated world as rates of successful actions that exceeded those expected by chance.

and avoid distraction, while during foraging they navigate at slower speed and are responsive to those sensors' data. This is why, in this example, the dispersion phase has been defined as distinct from foraging phase. Homeostatic tasks have also been introduced and are described in (Tzafestas 1994b).

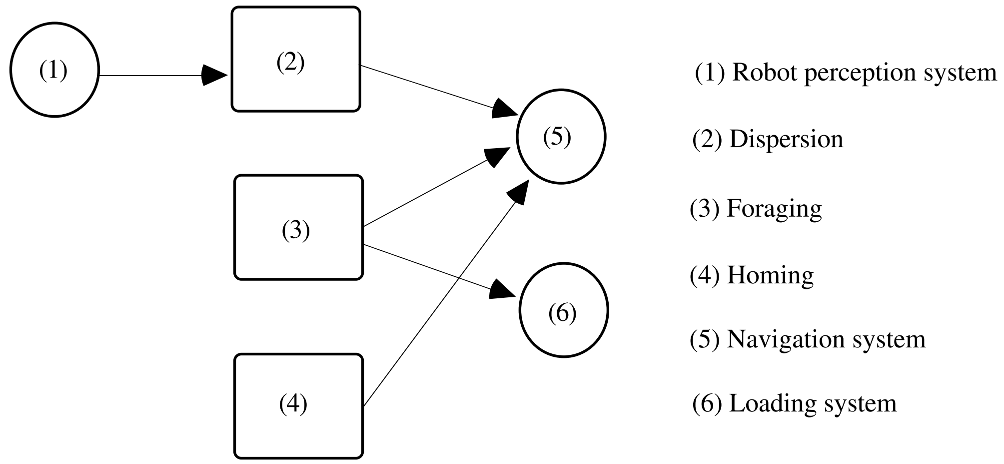

Fig. 5 : The structure of the network for the basic explorer robot

Network complexity. The cell network for an explorer robot without any homeostatic tasks has the structure shown in fig. 5 and comprises 37 cells, out of which 11 are sensor cells and another 5 are reflex sensor cells (they are distinct in that, in principle, normal sensor cells are allowed to be turned on/off). The actual processing part of the tasks is structurally fairly simple (see figures 6 and 7). This is due to the fact that the cellular architecture as defined has the potential to represent such parallel reactive algorithms in a fairly natural way. Take, for instance, the foraging task. As can be deduced from its structure (fig. 6), the local presence of a source of interest takes precedence over the distant perception of it (in the figure, four directions of limited perception are assumed), so that the usual problem of arbitration between "consummatory" and "appetitive" actions (which are the ethological counterparts of local and distant-perception-oriented actions) disappears! Indeed, the relation between such actions is *naturally represented* in the network itself, which also suggests that this distribution of the overall exploration task into individual activities uses a "good" behavioural *grain of activity* -good in the sense that it minimises complexity of the network structure as well as of the arbitration mechanism. This is also a consequence of the type and quantity of the sensors used. It was early identified in behaviour-based robotics research (Flynn & Brooks 1989) that the hard thing to build is sensors. Whether on real robots or on simulations, we are all debugging sensors all the time[8].

What design knowledge? As was decomposed the exploration problem, dispersion should take into account two parameters: the robot sensors capacity (i.e. their range of detection) and the size of the region each robot is supposed to cover, given that the society's role is defined as exploring a region within a certain radius from the base. In

[8] Oddly enough, though, (Kube & Zhang 1994) employ too few sensors and too many behaviours of different grains of complexity, what leads them to tackle the control problem as a pattern classification one and adopt an adaptive logic network (ALN) as the behaviour arbitration component.

fact, for our design purpose, what interests us is to find a general dispersion algorithm, or, at least, a parameterisable dispersion algorithm, that would give a good repartition of the robots in space as the radius of interest varies. Then, letting each robot forage locally, without any interference with other robots would accomplish the task. An assumption behind this line of work, is that exploration is in fact a domain where robots are supposed to be very sparse in space, and where their designer is interested in exploring bigger and bigger regions with the same population of robots, what would eventually lead him to turn to considering low spatial densities of robots. By inspecting the simulation of this algorithm, it was precisely found that the dispersion algorithm adopted (shown in fig. 7 and drawn from Mataric 1992b) worked well in the case of a small world, where the range of robot sensors was greater than their predicted radius of work region, which translates to density of robots being relatively high. For lower densities, there was no assured convergence on a good spatial repartition. This constitutes a first indication toward searching for a *statistical design model*, where parameters are not absolute values, but relative measures, such as densities.

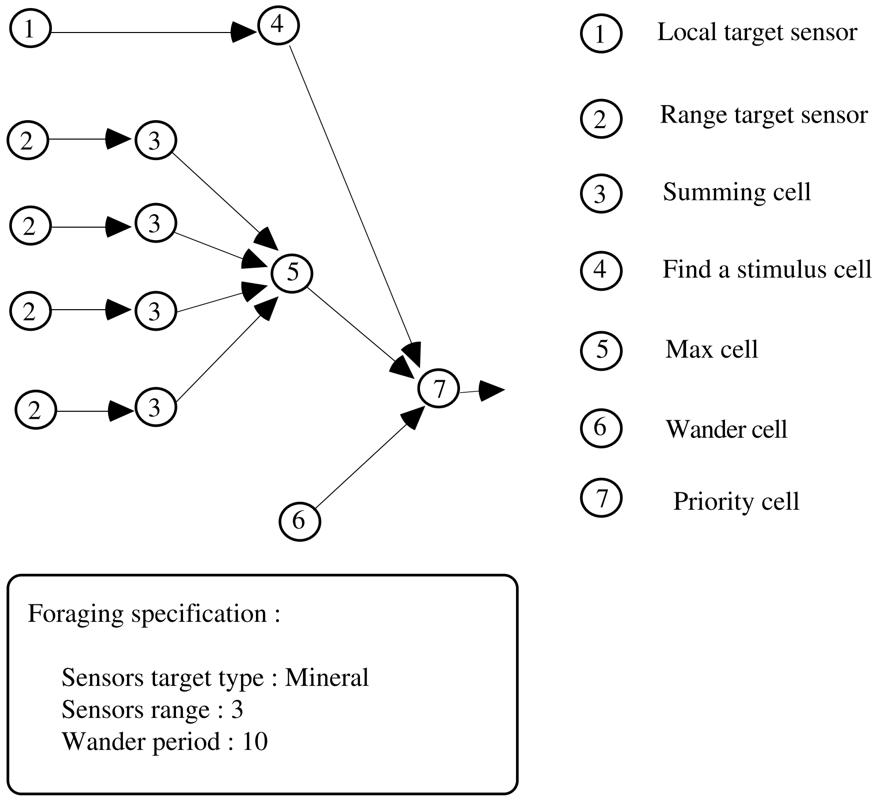

Fig. 6 : The foraging task's structure and specification

A second feature of this dispersion algorithm is that it suffices to replace a single processing cell to transform it to a flocking algorithm: replacing the min_cell by a max_cell actually makes the robots get closer to each other instead of trying to escape from each other. Similar observations were presented in (Braitenberg 1984).

On the globality of sensors. It is generally known that successful completion of a task necessitates a proper balance of global and local sensors (for some directions toward formalisation of this issue see Parker 1993). Now, let us look in the homing task. Initial setup for homing assumed a single local signal sensor mounted on each robot which would detect the orientation signal emitted by the space base whenever it needed to home back to it. Short after, it was seen that, to save energy, one could define this emission to be periodic. Inspection of the simulation in this case revealed that while for distant robots this sensor sufficed, whenever the robot approached the base it ran the risk of oscillating due to the short period of emission. Several trials of various periods of emission showed this period needed to be relatively large compared to interval between successive emissions, and yet this didn't assure convergence in the case of robots whose time constants where of the same order as the emission interval. This all seemed too complicated a way to tune the homing task, so we decided to add a more global sensor that would detect the base whenever close enough and whose output would take precedence over the signal sensor's one. As was expected, introduction of this sensor solved the problem. Closer examination, however, reveals that this second, range sensor, is by no means more global than the initial signal sensor, albeit of a broader perception range. Indeed, it is the signal sensor that provides absolute orientation information for the base, while the range sensor provides only local relative information. The misleading point was that a global sensor is a sensor of broad range. This is not true. A global sensor is a sensor that provides global information. Doing this by sensing only in a short range constitutes additional profit (*"What robotics needs is simple sensing that conveys complex ideas"*, (Miller 1994), p. 53). This kind of secondary sensing appears to simplify the design of task networks -in the absence of a signal sensor in the above case, the processing part of the homing task would have had the form of a random walk augmented with various map-building and map-updating capabilities, clearly far more complex than the present 5-cell implementation (two cells for the sensors, two for receiving the sensors outputs and one for the priority cell). The drawback of secondary sensing remains, however, the need for the designer to set up the robots' world in a way that makes (secondary) sensing possible -in the above example, find a way to make the base transmit a simple orientation signal all over the desirable exploration area.

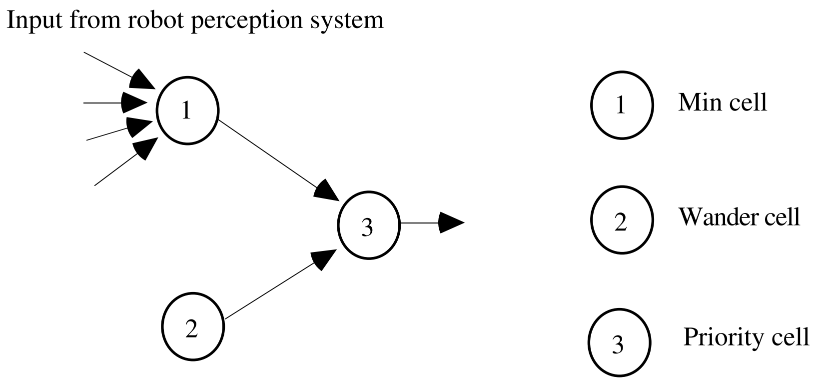

Fig. 7 : The dispersion task

Adaptation as a control parameter. What would it *mean* for a dispersion task to be adaptive, or, how could "moving away from objects" be adaptive? The only visible answer is to move away more selectively, that is, only when there are really too many objects round. Implementing this on the dispersion network involves changing the sensitivity of the min_cell to its inputs. In fact, it suffices to add an internal variable which serves as a threshold below which the min_cell is not responsive to its inputs. The default min_cell then corresponds to this threshold being implicitly set to 0. Adapting this sensitivity factor through blind reinforcement, i.e. become less sensitive to the presence of other objects, leads the robot to gradually slow down escape, so that what appears eventually as a good repartition in space (i.e. a steady state) will be more robust to perturbations, such as repeated unexpected encounters with other objects. Decreasing sensitivity, hence, corresponds to increasing *inertia* of the dispersion task, or else *stabilising the dispersion's performance*. Again, these are only true in the case of a highly populated space (perhaps because, after all, in a sparse world, dispersion is not a good idea). The preceding analysis also constitutes an argument to the proposition that, for adaptation to be feasible, we need some kind of explicit representation of what there is to be adapted, although this shouldn't be confused with the traditional AI central and symbolic representations (Brooks 1991a).

Now, consider adaptation of the arousal system of the dispersion task. At first, having an observer cell that triggers timely flow of activation to another task will cause dispersion to gradually lose activation up to a point where this other task will arouse. Say that this other task is foraging and that it, too, has in its arousal system an observer cell that triggers timely flow of activation back to dispersion. As a result, the robot will be switching between dispersion and foraging. Adapting one or both of the tasks' arousal systems by reinforcement of the flow quanta will result in variation of switches between dispersion and foraging. Using *informed reinforcement* (e.g. by connecting a counter observer cell to the dispersion's min_cell) will in turn force the robot to switch between the two tasks in a certain way (preset by the designer, e.g. in the previous case to minimise encounters with other robots and hence maximise spatial coverage): the robot will then appear as *tailoring and regulating its own activation state* (which, remember, is a picture of its motivations) according to the external conditions. Connection adaptation is therefore the means to *stabilise jointly* the two tasks' performance.

These observations lend support to the hypothesis that local and connection adaptation may be used, respectively, as control directions toward *individual and cooperative stability* of activities.

Controlling time. Another set of simulations concerned the wander period of the foraging task (similar observations apply to the dispersion task, as well). Examination of various wander time-outs revealed that this parameter is of importance only when the spatial density of sources of interest is relatively low, in which case a low time-out does not allow the explorer robot to travel long distances in a certain direction and hence discover remote sources. In high-density environments, however, the value of the wander time-out does not play a significant role, since encounters of sources are much more frequent than in the previous case. To reduce these observations down to a more statistical criterion as before, whenever the average distance between two sources of interest is substantially greater than the range of perception (which translates to low density of sources of interest), the average distance of continuous travel in a direction should be long enough for exploration to be efficient, otherwise perception alone suffices. More important yet, this nearly optimal value for the time-out does not need any tuning; it is a *task invariant* ! And a task invariant that does not define an isolated performance peak, but *a wide nearly-optimal performance plateau* : performance degrades visibly only when the value of the wander time-out becomes substantially greater or smaller than its nearly-optimal value, where "substantially" denotes change of order of magnitude. This analysis suggests that, once the time-out parameter's laws of interaction with the other parameters are understood, it may be used for controlling the whole population's performance, or, equivalently, it may be used as a true control parameter.

The above experiments demonstrated that reactive algorithms can be easily implemented on the cellular control architecture and this in an incremental and parameterisable way. Time and adaptation may be used as active, indirect control parameters for robotic tasks, that act as *entry points* to the statistical model. Indirection of control means that the controlled parameters have no straightforward relationship with the final result (for instance, in the previous case of foraging, the wander time-out does not correlate directly with the spatial coverage value), so that this may be viewed as an *emergent control* strategy. While adaptation was expected to be a control parameter (see section 2), it is not at first glance clear why it should be so for the time, too. The answer appears to be that time is *"the last resort to survival"* (Tokoro 1993, p. 6). Indeed, some algorithms are so primitive that this is the only intervention possibility (as is the case with foraging). Furthermore, such a low-level control parameter has a drastic impact on performance: around its near-optimal value, nothing else counts (e.g. adapting the sensitivity of the foraging's max cell won't do much in a poor world, where only optimal time-out counts). These also suggest that we need to introduce an additional concept for control, the *operational cycle*. Intuitively, an operational cycle corresponds to a certain activity delimited in time and space. A series of such activities exhibiting the same delimitation property is an operational cycle of a higher level. The foraging task's operational cycle is the maximum value between continuous period of wander and continuous period of attraction by a source of interest. The joint foraging and dispersion operational cycle is the period between successive starts of foraging (or dispersion) separated by a period of dispersion (foraging) - this joint operational cycle only makes sense in the case of a highly populated world, as explained before. Designing a whole society for a given task can be best done if one knows or identifies the necessary operational cycles because they translate into local structural parameters of the robots. This, in turn, reduces to knowing in advance the important features of the task (for example, having an estimate of the sources density) or, instead, learning them (we could imagine another set of observer cells counting and averaging intervals between successively encountered sources and continuously updating the wander's time-out). These by no means imply that a robot or a society can only operate within those cycles: they imply instead that they then operate almost optimally, i.e. within controllable limits. By bounding performance within limits, the concept of operational cycle also constitutes a basis for indirect inversion of the design problem to an exploration one: Given the observed performance of the society, what was the actual shape of the world? A formalisation of the operational cycle might permit to tackle questions like this one.

6. General discussion : Design as a recursive self-regulation problem

For the case of exploration, the instanciation of the model of fig. 1 may be as follows:

Function 1: Task completion

Criterion 1: Coverage

$u_{coverage}$ = coverage % / 100.

Criterion 2: Overlapping

$u_{overlapping}$ = 1 / overlapping (average of visits-per-place)

Function 2: Task-independent operation

Criterion 1: Time

$u_{time} = t_{min} / t$

Criterion 2: Energy

$u_{energy} = (e_{max} - e) / (e_{max} - e_{min})$

Criterion 3: Implementation cost

$u_{implementation\_cost} = (c_{max} - c) / (c_{max} - c_{min})$

Constraint: Cost limit

Criterion: $c \leq c_{limit}$

The min and max indices imply min and max values over the whole range of design alternatives[9].

Up to this point, the design problem has been formulated as a decision one[10]. A number of interesting observations may be drawn: First, the design process, i.e. the process of deciding on an alternative, relies on the criteria rather than on the alternatives themselves. Analysis of the individual criteria might not convey any useful information on the relative value of alternatives and the weight assignment procedure will tend to become intuitive in order to implicitly direct the design/decision process toward a solution which is "felt" right[11]. Second, this hierarchical decomposition occults some valuable information: the inter-relation between criteria and the continuity of alternatives. Indeed, in the above structure, it is clear for instance that coverage and overlapping are conflicting criteria (an alternative which results in greater coverage is expected to raise the overlapping penalty). On the other hand, and as far as the alternatives are concerned, it was shown above in the case of time control that one does not need to consider *every single value* for the timeout parameter, but rather reason on ranges or classes of such values. Last, but not least, this hierarchical structure does not represent explicitly the parameters that alternatives consist of, so that interactions between such parameters (such as the connection adaptation parameters of the dispersion and foraging tasks -see section 5 above) disappear from the reasoning (design) process[12]. These all lend support to the hypothesis that either the hierarchical structure of section 2 is all too rigid or the optimisation formula artificial. We believe it is the second one (and this also makes the structure to appear rigid).

What is design for optimality? The concepts of task invariants and operational cycles bring us to reconsider the issue of optimality: **Actually, we don't know what optimality is!** We don't know what is the best course of action to follow in a particular situation, because we have incomplete knowledge about situations that might arise and about the exact shape of the world we plunge our robots in. What we *do* know is that, as was shown, by controlling certain parameters we can bound population performance within certain limits (see the performance plateau reached through time control in the previous section). And this is optimal because this is the best we can do -at least, the best we can do safely and intentionally. The idea of bounded rather than single-peak optimality is also consistent with the presence of interactive parameters. For example, it has been shown that to stabilise jointly the performance of two tasks, one needs to control the connection adaptation parameters. But then those parameters can only be controlled as a whole! A consequence of considering bounded optimality is that now the designer's freedom is enhanced. No longer is the designer constrained to use traditional hill-climbing and search techniques. More loose, selective parameter exploration might do the job just as well, or even better. (Drogoul & Dubreuil 1993) solved the N-puzzle for N up to 899 at reasonable time using the general-purpose

---

[9] Other variants are possible, for example $u_{implementation\_cost} = 1 - (c/c_{limit})$, if one replaces c_{max} and c_{min} by c_{limit} and 0, respectively -this formula restricts $u_{implementation\_cost}$ to a sub-interval of [0,1].

[10] I should have expected that. After all, *"decisions underlie any action that a problem solver might take in structuring problems, reasoning, allocating resources, displaying information or controlling some physical activity"* (Henrion et al. 1991, p. 65). (Lévine & Pomerol 1993) also argue on the genericity of decision support systems and conclude that some class of DSSs *are* problem solvers. Early AI recognised this and departed on the direction of search methods -which was finally wrong, but for other reasons.

[11] (Pomerol 1993) discusses precisely this need to focus more on alternatives rather than on criteria.

[12] In (Tzafestas 1991) a DAI-inspired concept of *criteria groups* was proposed as a basis for solving the problems of inter-related criteria and interactive parameters through analysis of structural and functional dependencies. Two criteria are structurally dependent when they refer to the same parameter(s) and are functionally dependent when knowledge about the first ones restricts possibilities for the second one and vice versa. Functional dependencies are nothing else but *indirect* structural dependencies and for this reason more difficult to analyse.

distributed search method of eco-problem-solving (Ferber 1989), while classical centralised search methods have never solved a 25-puzzle -they have been too busy searching for the optimum[13].

In turn, bounded optimality, or else adequacy[14], has a surprising consequence: Actually, optimisation and alternatives (design model) are not discernible! Not only is the design problem ill-structured[15], but the boundaries between design method and design data are fluid: The data "include" in a sense all the information about performance, so that no artificial, top-down optimisation technique (through weights or whatever) needs to be applied. Instead, *a unified model of design process and design data is necessary* [16].

What would such a unified model be like? The key appears to be again the concept of the operational cycle: since operational cycles may appear at the individual robot level as well as at the society level, we need only match the two. Operating within such a cycle means self-regulating within certain limits. The designer's role is then to identify the global regulation limits and translate them into individuals' ones: understanding the exploration task means understanding the necessary cycle between foraging and dispersion in the above example -once this is done, design of the local regulatory mechanism, i.e. of the connection adaptation scheme, is straightforward. *Self-regulation should be the phenomenal law of every level.* For more complex tasks, that involve more sophisticated regulation, indirect control through connection adaptation might not be sufficient. In this case, one will need to add an explicit control variable whose value will represent the corresponding position within the cycle and will change either as a side-effect of the robot's actions (such as an energy level -you don't need to update any representation of it, just measure it!) or explicitly through activity (such as an internal abstract motivation). This means, all one needs is homeostatic tasks at a high-level, each one regulating its own variable[17]. Those variables will then appear to an external observer as nothing else but the robot's motivations to perform the corresponding tasks. Since the tasks are homeostatic, they will respond to external perturbations to the states of the control variables (from now on called motivations) by tending to bring themselves back into the proper operation limits. Several points have to be made on this self-regulation framework:

---

[13] As expected, though, the N-puzzle eco-problem-solving implementation is outperformed by special-purpose algorithms for the N-puzzle. The point remains that another general-purpose technique (the eco-problem-solving) outperforms classical search for some classes of problems.

[14] (Bourgine & Le Moigne 1991) introduce the concept of adequate decision-making building on the work of Herbert Simon.

[15] (Simon 1973) argued that every problem may be regarded as ill-structured as long as there is extensive knowledge about it and that this is a reason to believe generic problem solving methods exist. Is this why he wanted satisficing instead of optimal solutions?

[16] Although artificial intelligence has traditionally separated reasoning from knowledge, serious doubts have been independently expressed by various researchers in various contexts. Some examples: (Minsky 1965) argues that no separation between coding and knowledge structure should be expected and that the central argument behind such an erroneous separation would be belief in dualism. (Pitrat 1990) states that *"il n'y a pas de différence essentielle entre fait et connaissance; simplement dans chaque situation un sous-ensemble des faits a une certaine permanence qui lui vaut un traitement spécial et de s'appeler 'connaissance' "* (p. 26). Brooks in two complementary papers (Brooks 1991a, Brooks 1991b) argues that neither reason nor representation, in the ordinary sense, are necessary for a machine to demonstrate a degree of intelligence. In (AFCET 1994) Roy says that a good decision does not preexist for somebody to discover but is the product of a constructive process and this implies that decision model and decision data are not always indistinguishable (p. 4).

[17] Note how elegant a solution the definition of high-level homeostatic tasks is to the interactive parameters problem: in fact, what we are interested in and what is modelled is the interaction and not the parameters themselves.

i) Motivations need not be static but may flow from task to task -this seems to fit well the activation scheme of the cellular control architecture of section 3 and the kind of society tasks we are interested in. This scheme also differs from Maes' spreading activation scheme (Maes 1989) in two ways: First, total motivation has to be constant for the distributed activation/inhibition to make sense and, second, the behavioural grain adopted here is coarser than that used by Maes (see, for example, the foraging task)[18]. (Parker 1992) also uses special motivational behaviours that control the activation of behaviour sets. In an unpublished report of mine (July 1992) as well as in (Tzafestas 1993) I had advocated too the necessity for special arbitration behaviours. I abandoned that idea since, for two reasons: First and as explained in (Tzafestas 1994a), the behavioural grain used in the cell network is coarser than that used by Parker, and largely based on (Brooks 1990), so that Parker's behaviours set would be one task in the cell network -in this case, it wouldn't make any sense to add an extra arbitration task for each activity task. Instead, and since the size of the arbitration task is much smaller than that of the activity task, it looks natural to include arbitration in the task as a separate aggregate. For the same granularity reason, there is no need for a sensory filter. Second, and more important, the behavioural grain is chosen to be the one that minimises inter-task connections: We want tasks to be parts of the cell network relatively densely and carefully connected, but connections among tasks have to be quite sparse (Minsky 1991 also advocates such a sparse connection scheme, p. 46). Of course, this scheme would introduce a degree of opacity, but this isn't necessarily a problem, because Minsky suspects *"our societies and hierarchies of subsystems have evolved ways to evade the problem by arranging for some of our systems to learn to model what some of our other systems do"* (Minsky 1991, p. 48).

ii) Since total motivation is constant, the designer is now armed with a powerful and safe means to compare individual motivations at design time. This would correspond to a *"common currency"* (a measure of pleasure, according to Cabanac 1991) between what would appear as the robot's "emotions". Werbos, with which we share the view of design as an operations research problem, states that emotions may be simply interpreted as short-term evaluation functions, in our case the maximisation of pleasure according to the motivations present. These short-term evaluation functions are such that they maximise the long-term utility and are subject to change through learning (Werbos 1987, p. 15). It is important to note that these motivations should *not* be otherwise controllable by the robot itself, lest we run the risk of it learning to cheat and be useless to us: *"If we could deliberately seize control of our pleasure systems, we could reproduce the pleasure of success without the need for any actual accomplishment. And that would be the end of everything"* (Minsky 1986, section 6.13). Those motivations will make the robot appear intentional, although it is completely -but indirectly- controlled by its designer[19].

iii) Then the state of the robot's motivation (i.e. the set of states of individual motivations) will be a point in *a motivational space* and this space will have the property of being closed, since any perturbation in one state will bring the robot to another point within the same space -as long as the total motivation is constant. Actually, a robot can't be efficiently controlled if we don't adopt the above *closed world hypothesis*. A consequence of this hypothesis is that the designer should then shift its focus of work in the modelling of the interactions between tasks and between the robot and the world. This does not imply that *every* single action, outcome of action or perception has to be known in advance, but instead that every perturbation to the motivations has to be modelled -reasoning in such a qualitative and more abstract composition level appears easier. Designing the robot is then reduced to designing its motivational space and initialising the

---

[18] (Tyrrell 1993) also remarked that the grain of activity used by Maes leads to unbalanced competition and that the non-constant activation forces the use of problematic division rules.

[19] Dawkins drew the analogy between a being completely -although indirectly- controlled by its genes and an intentional -albeit unconscious- machine, such as the steam engine (Dawkins 1976, chapter 4).

state of motivations to a desirable starting point in it - an autonomous robot won't go hunting by itself, unless it is born hungry. The necessity of such a dynamics approach to autonomous robot control through use of motivational variables has been raised by (Steels 1994) and justified as a remedy against inflexibility and rigidity of algorithmic approaches and as a means to import results from physics, biology and control theory (pages 88-89). We believe that adopting the present hybrid approach of reactive algorithms (tasks) combined through a motivational space medium will preserve speed and efficiency of behaviour while enforcing smoothness and extensibility. Some initial investigations on the subject of compositionality and motivational space may be found in (Tzafestas 1994b). The notion of robots situated and navigating in a motivational space -in a way whose by-product is the completion of the designer's goal- is also close to what Tokoro speculated as volitional agents (*"Volition might be the true meaning of autonomy"*, Tokoro 1993, p. 10).

As unlikely as it may seem, the closed world assumption is not contradictory with the open systems principles of concurrency, asynchrony, decentralised control, inconsistent individual information, arm's-length relationships and continuous operation (Hewitt 1986). Hewitt himself pointed out that *"the indeterminacy of concurrent systems does not stem from invoking a random element such as flipping a coin. Instead, it results from the indeterminate arrival order of inputs to system components"* (Hewitt 1986, p. 322). In a recent review, Tokoro redefined open systems as concurrent distributed systems, extended with the additional possibility of constant change (Tokoro 1993). Again, this doesn't contradict the closed world hypothesis. There *is* concurrency, there *is* incomplete individual knowledge and there *is* change. The point is, the results of possibly inconsistent or displaced actions and the results of changes are propagated, so that the whole system (or society for that purpose) may be seen to obey to a set of laws in *a physical meta-level*. Zeghal also remarked that all ethologically-inspired approaches to autonomous agents lacked such concern for action-reaction coupling (Zeghal 1993, p. 2).

From the above discussion follows that the design problem should be rather viewed as a recursive self-regulation one driven by a set of criteria and a number of operational cycles. Global adequacy of operation (as opposed to optimality) will be ensured through local self-regulation at the individual robot level. Finally, the concept of a closed motivational space appears to be a valid substrate for design of such regulatory mechanisms.

7. Conclusions and perspectives

To summarise the work presented:

The design problem was formulated as a decision problem where an optimisation formula is applied to a set of design criteria, each one being an evaluation of input task specifications across design alternatives, i.e. robot specifications. For the identification of design and/or experimentation parameters the structural stance is taken, i.e. it is assumed that purely structural parameters account for all observed behaviour. Design was also given a second formulation as a control problem, whose main parameter is the adaptation mechanism.

A cellular control architecture was then briefly presented that combines features from subsumption architecture and connectionism and exhibits extensibility, specification construction possibility as well as parameterisation possibility. The most prominent characteristics of the architecture are: Three level nestedness (cells, aggregates, tasks), categorisation of cells, aggregates and tasks, definition of reflexes independent of the commands fed to the actuators, arbitration through activation/inhibition pathways, adaptation mainly by reinforcement at the cell level, finally definition of observer cells that serve as meta-cells.

The experimentation context and the simulation environment were next described, with emphasis on the world's role of repository, the observation mechanism and the use of specifications to automate the experimentation process.

Investigations on a space exploration task revealed that the design knowledge sought should be of statistical nature and two identified design/control parameters were found to be the adaptation mechanism and the time. Some secondary issues have also been raised, namely that of the globality of sensors and of the appropriateness of a behavioural grain of activity. Finally, the concept of an operational cycle has been introduced and its potential use for inversion of the design problem was speculated.

Following these observations, the overall design problem has been revisited and refined. The notion of optimality has been examined more closely and the concept of bounded optimality introduced which led to a quest for a unified design principle and model. The concept of operational cycle allowed some light to be shed on this issue: It was then identified that the principle that appears to be reproduced recursively at each level (i.e. the micro as well as the macro level) is self-regulation and the concept of a closed motivational space of individual robots was introduced as a substrate for design of micro-regulatory mechanisms. The study of possible implementations for this motivational space on more complex tasks, such as supervision and inspection, will constitute the next part of this work.

Post Scriptum : Why autonomy is controllable or Beethoven revisited

Science, and especially artificial intelligence, is about sharing ideas and approaches. And about invading frontiers with what was philosophy before[20]. In an attempt to excite people, I gave a lyric title to my previous report (Tzafestas 1993). I think I have been mistaken. Beauty is seldom taken seriously: *"Science is unlike art and literature in that if one scientist does not find a particular idea, then another one will, and soon"* (James Anderson, General introduction to Anderson & Rosenfeld 1988, p. xxi). Artificial intelligence is supposed to have come to sow doubt on this.

What I wanted to say in this previous report was that we shouldn't be too much obsessed by intelligence. And that, in contrast with what is generally said in robotics and which I described as an unfortunate by-product of scientific evolution, autonomy is a prerequisite to intelligence and not vice versa. The example that served as basis to my reasoning was that of the initially ignorant child that becomes a composer, and I argued that to become autonomous one doesn't need to have an explicit representation of itself nor to fully understand any artificial rules (or models) about it - Beethoven is a well-known example of this. (Covrigaru & Lindsay 1991) also argued that autonomy is the key property to intelligence. I went a step beyond this in saying that the other half of intelligence is the possibility to modify the outside world intentionally, although this intention isn't necessarily explicitly represented in the intelligent agent, and actually I believe it isn't represented at all. Hence, autonomy is viewed as the possibility of the agent to survive in a world by continuously adapting to it, while intentional modification should result in a world that better fits the agent. Covrigaru and Lindsay spoke about goals and intentions, but adopting the symbol-system-hypothesis led them naturally to assume explicit symbolic representations. My choice of a distributed control architecture with the possibility to ascend many representational meta-levels (Tzafestas 1994a) makes me reject this explicit representation assumption, whether of symbolic or other nature. In such a context, the actual intentions of a hypothetical intelligent machine (i.e. the machine's own

[20] *"... what we call intelligence remains an extreme challenge ... the frontiers are expanding at an ever increasing rate ..."* (Winston & Shellard 1990, Preface cum Introduction, p. xvii)

model of itself) may be very different from the intentions an observer perceives (*"When intelligent machines are constructed, we should not be surprised to find them as confused and as stubborn as men in their convictions about mind-matter, consciousness, free will and the like"*, Minsky 1965, p. 431). Covrigaru and Lindsay also mentioned built-in and acquired goals and insisted on the equally deterministic nature of the latter ones, too. In the end, they arrived at the same conclusion as I did, that what an autonomous machine needs is homeostatic built-in goals. Again, working on a distributed architecture made me do a further step: Controlling the machine means defining its motivational space as closed, so that it tries to regulate its own state. The designer's job is then to find such a motivational space that matches the global task at hand. The machine will still be autonomous because it will try to adapt to its environment, but it can only do so *within* its motivational space[21]. This space appears therefore as an intermediate representation level (Minsky 1991, p. 49). I also said in my previous report that I was taking inspiration from Varela's theory of autopoiesis and from Saridis' and Meystel's intelligent control theory and I justified this by saying that the property those two share and which I thought necessary for design was their applicability at all levels of a nested system, in my case at the individual robot level as well as the society level[22]. What I failed to observe by that time is that they both share also the closed world assumption[23] - perhaps this is because, without self-regulation in a closed world, you can't travel safely across levels. A final thorny point in that report was that the view of intelligence as autonomy plus the possibility of intentional modification of the world could be misinterpreted by some as an argument for dualism. I tried to circumvent this problem by saying this was a pseudo-random process. I can now be a little more specific: In fact I believe it isn't random. It is a part of the network (the part B that activates this other "autonomous" part A, like in (Minsky 1986), section 6.4) that is responsible for what looks random. The part B, like the part A, is otherwise predetermined by the initial structure of the network, even if it might have changed since birth (Covrigaru & Lindsay 1991). One just has to realise that *"as in any society, there must be watchers to watch each watcher, lest any one or a few of them get too much control of the rest."* (Minsky 1991, p. 50). My observer cells may also assume such a role, besides other things.

As was the case with my previous report, I left this one open, too. What looked self-contained then, is just a part of the story now. I expect the present story to have the same fate. A mind never stops thinking. And science never ends.

References

AFCET (1994). J.-C. Courbon, D. Dubois, B. Roy interrogés par J.-Ch. Pomerol : Autour de l'aide à la décision et de l'intelligence artificielle, *LAFORIA Research Report 94/01*, January, 25 p.

[21] Even the most intelligent of all known beings, humans, are sometimes controlled through food to their motivations.

[22] *" ... the same organization may be realized in different systems with different kinds of components as long as these components have the properties which realize the required relations."* (Varela et al. 1974, p. 188)

"The control intelligence is hierarchically distributed according to the principle of Decreasing Precision with Increasing Intelligence, evident in all hierarchical management systems." (Saridis 1985, p. 24)

[23] *"The necessary feature is the presence of a boundary which is produced by a dynamics such that the boundary creates the conditions required for its dynamics."* (Varela et al. 1974, p. 191-192)

"Exosystem is becoming a part of the control problem." (Meystel 1988, p. 280)

Anderson, J. A., Rosenfeld, E. (1988). *Neurocomputing, Foundations of research*, MIT Press.

Arkin, R. C. (1987). Motor schema based navigation for a mobile robot : An approach to programming by behaviour, *Proceedings of the 1987 International Conference on Robotics and Automation*, Raleigh, NC, March, pp. 264-271.

Bourgine, P., LeMoigne, J. L. (1991). Les bonnes décisions sont-elles optimales ou adéquates?, in H. F. Bradley (Ed.), *Operational Research '90, Selected papers from the 12th IFORS International Conference on Operational Research*, Pergamon Press, 1991, pp. 215-225.

Braitenberg, V. (1984). *Vehicles - Experiments in synthetic psychology*, Bradford Books/MIT Press.

Brooks, R. A. (1986). A Robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, RA-2(1):14-23.

Brooks, R. A. (1990). The Behavior Language - User's Guide, *M.I.T. AI Memo No. 1227*, 35 p.

Brooks, R. A. (1991a). Intelligence without Representation, *Artificial Intelligence, Special Issue on the Foundations of Artificial Intelligence*, 47:139-159.

Brooks, R. A. (1991b). Intelligence without Reason, *M.I.T. AI Memo No. 1293*, April, 27 p., also*Computers and Thought Workshop, IJCAI-91.*

Cabanac, M. (1991). Pleasure : The Answer to Conflicting Motivations, in J.-A. Meyer, S. W. Wilson (Eds.), *From Animals to Animats, Proceedings of the First International Conference on Simulation of Adaptive Behavior*, Paris, 1990, Bradford/MIT Press, pp. 206-212.

Carle, P., Collinot, A., Jacopin, E., Zeghal, K. (1994). Comportements collectifs: des agents sous influences, *LAFORIA Research Report 94/03*, April, 11 p.

Chandrasekaran, B., Goel, A., Allemang, D. (1988). Connectionism and information-processing abstractions (The message still counts more than the medium), *AI Magazine*, 9(4):24-34, Winter 1988.

Connell, J. (1990). *Minimalist mobile robotics : a colony-style architecture for an artificial creature*, MIT Press, Cambridge, MA.

Covrigaru, A. A., Lindsay, R. K. (1991). Deterministic autonomous systems, *AI Magazine*, 12(3):110-117, Fall 1991.

Dawkins, R. (1976). *The selfish gene*, Oxford Univ. Press.

Drogoul, A., Dubreuil, C. (1993). A distributed approach to N-puzzle solving, *Proceedings of the 13th Workshop on Distributed Artificial Intelligence*, 13 p.

Flynn, A. M., Brooks, R. A. (1989). Building Robots : Expectations and Experiences, *IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS '89)*, September 1989, Tsukuba, Japan, pp. 236-243.

Ferber, J. (1989). Eco-Problem-Solving : How to solve problems by interactions, *Proceedings of the 9th Workshop on Distributed Artificial Intelligence*, pp. 113-128.

Fukuda, T., Kawauchi, Y. (1991). The cellular robotic system (CEBOT), a self-organizing system, in S. G. Tzafestas (Ed.), *Intelligent Robotic Systems*, Marcel-Dekker, pp. 137-163.

Goel, V., Pirolli, P. (1989). Motivating the notion of generic design within information-processing theory: The design problem space, *AI Magazine*, 10(1):19-36, Spring 1989.

Harvey, I., Husbands, P., Cliff, D. (1992). Issues in evolutionary robotics, in J.-A. Meyer, H. L. Roitblat, S. W. Wilson (Eds.), *From Animals to Animats 2,* *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, Hawaii, Bradford/MIT Press, pp. 364-373.

Henrion, M., Breese, J., Horwitz, E. (1991). Decision analysis and expert systems, *AI Magazine*, 12(4):64-91, Winter 1991.

Hewitt, C. (1986). Offices are open systems, *ACM Transactions on Office Information Systems*, 4(3):271-287, reprinted in A. H. Bond & L. Gasser (Eds.), *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann, San Mateo, California, 1988, pp. 321-329.

Kube, C. R., Zhang, H. (1994). Collective robotics : From social insects to robots, *Adaptive Behavior*, 2(2):189-218.

Lévine, P., Pomerol, J.-Ch. (1993). Are DSSs problem solvers?, *LAFORIA Research Report 93/40*, November, 19 p.

Maes, P. (1989). The Dynamics of Action Selection, *Proceedings of the 1989 International Joint Conference on Artificial Intelligence*, Detroit, pp. 991-997.

Mataric, M. J. (1992a). Minimizing complexity in controlling a mobile robot population, *Proceedings of the 1992 IEEE International Conf. on Robotics and Automation*, Nice, France, May, pp. 830-835.

Mataric, M. J. (1992b). Designing emergent behaviors : From local interactions to collective intelligence, in J.-A. Meyer, H. L. Roitblat, S. W. Wilson (Eds.), *From Animals to Animats 2, Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, Hawaii, Bradford/MIT Press, pp. 432-441.

Maturana, H. R., Varela, F. J. (1987). *The tree of knowledge : The biological roots of human understanding*, New Science Library/Shambhala, Boston.

McLennan, B. J., Burghardt, G. M. (1994). Synthetic ethology and the evolution of cooperative communication, *Adaptive Behavior*, 2(2): 161-188.

Meystel, A. (1988). Intelligent control in robotics, *Journal of Robotic Systems*, 5:269-308.

Miller, D. P. (1994). The long-term effects of secondary sensing, *AI Magazine*, 15(1):52-56, Spring 1994.

Minsky, M. (1965). Matter, mind and models, *Proceedings of the 1965 IFIP Congress, Vol. 1*, Sparta Books, pp. 45-49, reprinted in M. Minsky (Ed.), *Semantic Information Processing*, MIT Press, Cambridge, MA, 1968, pp. 425-432.

Minsky, M. (1986). *The Society of Mind*, Simon and Schuster, New York, 1986.

Minsky, M. (1991). Logical versus analogical or symbolic versus connectionist or neat versus scruffy, *AI Magazine*, 12(2):34-51, Summer 1991.

Parker, L. E. (1992). Adaptive action selection for cooperative agent teams, in J.-A. Meyer, H. L. Roitblat, S. W. Wilson (Eds.), *From Animals to Animats 2, Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, Hawaii, Bradford/MIT Press, pp. 442-450.

Parker, L. E. (1993). Designing control laws for cooperative agent teams, *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, Atlanta, GA, May, pp. 582-587.

Pitrat, J. (1990). *Métaconnaissance - Futur de l'Intelligence Artificielle*, Editions Hermès, Paris.

Pomerol, J.-Ch. (1993). Multicriteria DSSs : State of the art and problems, *LAFORIA Research Report 93/32*, September, 15 p.

Saridis, G. N. (1985). Foundations of the theory of intelligent controls, *Proceedings of the 1st IEEE Symposium on Intelligent Control*, Troy, NY, pp. 23-28.

Simon, H. (1973). The structure of ill-structured problems, *Artificial Intelligence*, 4:181-201.

Steels, L. (1994). The artificial life roots of artificial intelligence, *Artificial Life*, 1(1-2):75-110.

Tokoro, M. (1993). The society of objects, *Addendum to the Proceedings, OOSPLA-93*, pp. 3-12, Washington, DC.

Tyrrell, T. (1993). *Computational mechanisms for action selection*, Ph.D. Thesis, Centre for Cognitive Science, University of Edinburgh.

Tzafestas, E. S. (1991). *Un système de modélisation et de saisie de cahiers des charges fonctionnels. Application aux produits électromécaniques*, Univ. Paris VI D.E.A. Thesis & Télémécanique R&D Research Report 1101, September, 56 p.

Tzafestas, E. S. (1993). De la machine à vapeur à Beethoven en passant par les robots autonomes, *LAFORIA Research Report 93/16*, May, 42 p.

Tzafestas, E. S. (1994a). A cellular control architecture for autonomous robots, *Proceedings of the 1994 International Workshop on Intelligent Robotic Systems*, Grenoble, July, to appear.

Tzafestas, E. S. (1994b). Implementing reactive algorithms on a cellular control architecture, *Proceedings of the 10th Autonomous Mobile Systems Workshop*, Stuttgart, October, to appear.

Varela, F. J., Maturana, H. R., Uribe, R. (1974). Autopoiesis : The organization of living systems, its characterization and a model, *BioSystems*, 5:187-196.

Wang, J. (1990). *The theory and engineering of cellular robotic systems*, Ph.D. Thesis, University of California, Santa Barbara, June.

Werbos, P. J. (1987). Building and understanding adaptive systems : A statistical/numerical approach to factory automation and brain research, *IEEE Transactions on Systems, Man and Cybernetics*, 17(1):7-20.

Winston, P. H., Shellard, S. A. (Eds., 1990). *Artificial Intelligence at MIT: Expanding frontiers*, MIT Press, Cambridge, MA.

Zeghal, K. (1993). Un modèle de coordination d'actions pour agents mobiles réactif et décentralisé, *LAFORIA Research Report 93/41*, December, 19 p.