THE DYNAMICS OF ACTION SELECTION

Pattie Maes

AI-LAB
V.U.B.
Pleinlaan 2
B-1050 Brussels
pattie@arti.vub.ac.be (UUCP)

Abstract

The paper addresses the problem of action selection for an autonomous agent. An autonomous agent is viewed as a collection of "competence modules". The hypothesis that is put forward is that rational action selection can be modeled as an emergent property of an activation/inhibition dynamics among these modules. Results of a computer simulation are discussed and a first step towards a qualitative theory is presented. The advantages and disadvantages of this non-hierarchical, distributed form of control of action over the classical, programmed, centralized method are studied.

1 Introduction

This paper addresses the following problem: Given an autonomous agent which has a number of general goals and which is faced with a particular situation at a specific moment in time. How can this agent select an action such that global rational behavior results? Characteristics of rational behavior are: it is goal-oriented, opportunities are exploited, some looking ahead is done, it is highly adaptive to unpredictable and changing situations, it is able to realize interacting and conflicting goals, and there is a graceful degradation of performance when certain components fail, all of this with limited resources and incomplete information.

The paper studies this problem in the context of Minsky's Society of the Mind theory [10] (to which Brooks' Subsumption Architectures are very much related [3]). This theory suggests to build an intelligent system as a society of interacting, mindless agents, each having their own specific competence. For example, a society of agents that is able to build a tower would incorporate agents for finding a block, for grasping a block, for moving a block, etc. The idea is that agents cooperate (locally) in such a way that the society as a whole functions properly. Such an architecture would be attractive because of its modularity, distributedness, flexibility and robustness: agents can be added, changed, or modified without caring about ihe other agents.

One of the open problems is however how action could be controlled in such a distributed system. More specifically: (i) how is it determined whether some agent should become active (taking some real world actions by steering the effectors) at a specific moment or not, and (ii) what are the factors that determine a cooperation among certain agents. Several solutions can be adopted. One approach is to hand-code (and by that hard-wire) the control flow among the agents [3]. Another approach is to use a meta-level which tells the agents whether they are allowed to perform an action or not. This paper investigates yet another, entirely different type of solution.

The hypotheses that are tested are:

(i) rational action of the global system can emerge by letting the agents activate and inhibit each other in the right way,

(ii) no "bureaucratic" agents are necessary (i.e. agents whose only competence is determining which other agents should be activated or inhibited) nor do we need global forms of control.

The research questions that are studied are how adequate this solution is and which activation/inhibition dynamics is appropriate. To this end two tools have been developed: a simulation environment and a qualitative theory. Using the simulation environment, societies of agents can be defined and their behavior simulated. Several input parameters can be varied, such as the threshold for becoming active, the strengths of activation and inhibition, the influence from the global goals and the state of the environment, etc. Experiments have been performed for several applications. The resulting systems exhibit a selection of action which can be made more/less data-oriented (and thereby more/less opportunistic), more/less goal-oriented, more/less deliberated, and more/less fast by choosing certain relations in the input parameters. They also exhibit the expected properties of flexibility (adapting to new or unforeseen situations), robustness (graceful degradation of performance) and modularity (agents are black boxes, only their expected behavior has to made explicit, therefore it is easy to introduce new agents or modify agents).

The second tool used to test the hypotheses is a qualitative theory of the behavior of these dynamical systems. A first set of laws can be stated about the relation between the input parameters and the observables and between the structure of the network and the observables.

2 Theory. Part I: Abstract

The agents in a society resemble the operators of a classical planning system. An agent i can be described by a tuple (l_p, l_a, l_d, α). l_p is a list of preconditions which have to be fulfilled before the agent can become active. l_a and l_d represent the expected effects of the agent's action in terms of an add list and a delete list. In addition each agent has a level of activation a. An agent is executable at time t when all of its preconditions are observed to be true at time t. An executable agent may become active, which means that it will - perform some real world actions. The operation of an agent (what computation it performs and how it takes its action) is not made explicit. I.e. agents could be hard-wired inside, they could perform logical inference, whatever.

Agents are linked in a network through predecessor and successor links. The description of the agents of the society in terms of a precondition list, add list and delete list completely defines this network. There is a successor link from agent 1 to agent 2 ("1 has 2 as successor") for every predicate P which is member of the add list of 1 and also member of the precondition list of 2 (so more than one successor link between two agents may exist). Formally, given agent $1=(l_{p_1}, l_{a_1}, l_{d_1}, \alpha_1)$ and agent $2=(l_{p_2}, l_{a_2}, l_{d_2}, \alpha_2)$, there will be a successor link from 1 to 2, for every predicate $P \in l_{a_1} \cap l_{p_2}$. Further, a predecessor link from 1 to 2 ("1 has 2 as predecessor") exists for every successor link from 2 to 1.

The links of the network are used to spread activation among agents. There is an external input of activation coming from the state of the environment and the global goals of the society. This input is continuous: there is a permanent flow of activation towards the agents which partially match the current state or promise to realise one of the global goals. The state of the environment and the global goals may change unpredictably at any moment in time. If this happens, the external input of activation will automatically flow to other agents. An agent is said to partially match the current state if one of its preconditions is observed to be true. An agent is said to promise to realise one of the global goals if one of the goals is member of the add list of the agent.

Agents spread activation along their links as follows. An executable agent spreads activation forward. It gives away part of its own activation to some of its successors. Intuitively, we want these successor agents to become more activated because they are "almost executable", since more of their preconditions will be fulfilled after the agent has becomes active. Formally, given that agent $1=(l_{p_1}, l_{a_1}, l_{d_1}, \alpha_1)$ is executable, it spreads forward through those successor links for which the predicate that defined them $P \in l_{a_1}$ is false. An agent that is not executable spreads activation backward. It gives away a part of its own activation to some of its predecessors. Intuitively, such an agent spreads to the agents that "promise" to fulfill its preconditions that are at the moment false, so that the agent may become executable itself afterwards. Formally, given that agent $2=(l_{p_2}, l_{a_2}, l_{d_2}, \alpha_2)$ is not executable, it spreads backward through those predecessor links for which the predicate that defined them $P \in l_{p_2}$ is false.

At every timestep the following computation takes place for all of the agents, (a) The input from the state and goals to an agent is computed, (b) The spreading of activation of an agent is computed, (c) A locally computed "forgetting" factor ensures that the overall activation level remains constant, (d) The agent which fulfills the following three conditions becomes active (in case there is one): (i) It has to be executable, (ii) Its level of activation has to surpass a certain threshold and (iii) It must have a higher activation level than all other agents which fulfill conditions (i) and (ii). When two agents fulfill these conditions, one of them is chosen randomly. Steps (a) through (d) are repeated infinitely.

Four global input parameters can be used to "tune" the input and the spreading of activation, and thereby the behavior of the whole society, (i) The threshold for becoming active (and related to it the quantity of activation put into the society at every timestep). (ii) The percentage of their activation executable agents spread forward, (iii) The percentage of their activation non executable agents spread backward. And (iv) The relative amount of external input that comes from the goals as opposed to from the state of the environment. Interesting global properties are: the sequence of agents that have become active, the optimality of this sequence (which is computed by a domain-dependent function) and the speed in choosing an action (the number of timesteps an agent has become active relative to the total number of timesteps the system has been running).

3 Theory. Part II: Illustration

This section illustrates the theory with a concrete, simple example. Later in the paper more interesting examples are discussed. Our toy example is that of a robot with two hands that has to sand a board. The definition of the agents in terms of their preconditions, add and delete lists is presented in figure 1. On the basis of these definitions the spreading activation network in figure 2 is constructed.

This network performs action selection in the following

PICK-UP-SANDER
preconditions: (hand-empty, sander-somewhere)
add-list: (sander-in-hand)
delete-list: (hand-empty, sander-somewhere)

PUT-DOWN-SANDER
preconditions: (sander-in-hand)
add-list: (hand-empty, sander-somewhere)
delete-list: (sander-in-hand)

PICK-UP-BOARD
preconditions: (hand-empty, board-somewhere)
add-list: (board-in-hand)
delete-list: (hand-empty, board-somewhere)

PUT-DOWN-BOARD
preconditions: (board-in-hand)
add-list: (hand-empty, board-somewhere)
delete-list: (board-in-hand)

SAND-BOARD
preconditions: (sander-in-hand, board-in-hand)
add-list: (board-sanded)
delete-list: ()

Fig. 1. Definition of the agents involved in the toy example.

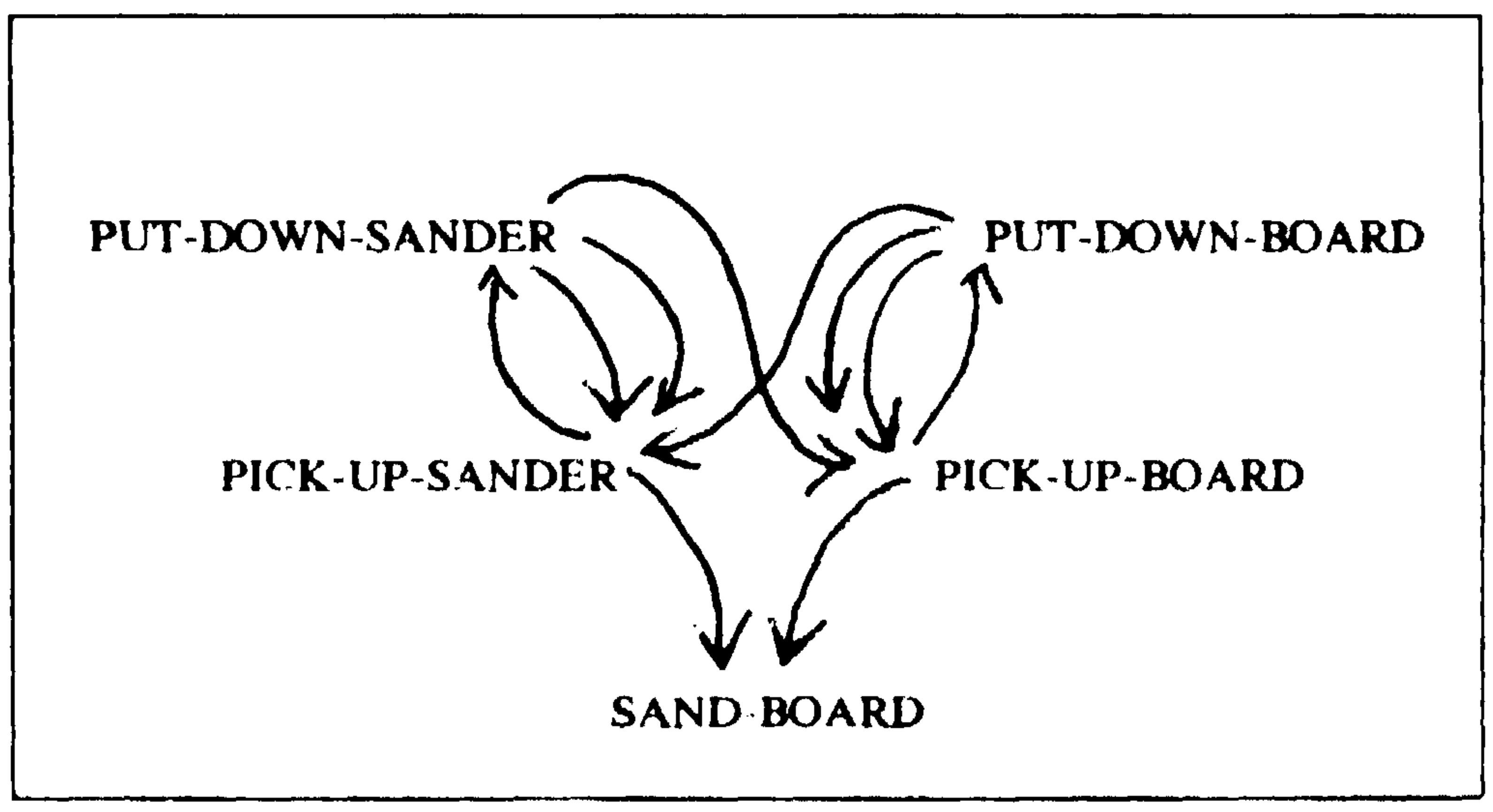

Fig. 2. A spreading activation network. Only the forward links (from an agent to its successors) are shown. The actual network has twice as many links (one backward link in the reverse direction for every forward link shown here).

way. The activation levels of the agents are initialized to zero. Suppose that at *time~0* the state of the environment is *S(0)=(hand-empty\hand-empty$_f$ sander-somewhere$_t$ board-somewhere)* and the goals of the society *G(0)=(board-sanded)* (hand-empty is represented twice since the robot has two free hands). This means that *pick-up-sander* and *pick-up-board* receive external input of activation (because they partially match the state), while *sand-board* receives input because it promises to realize the goal of the society. During the internal spreading of activation phase, *sand-board* spreads backwards to *pick-up-sander* and *pick-up-board,* while these spread forward to *put-down-sander$_t$ put-down-board* and *sand-board.* The remaining two agents (i.e. *put-down-sander* and *put-down-board)* will not have anything to spread yet. The external input and internal spreading of activation will continue and makes activation increase in certain agents. More specifically after some time either *pick-up-sander* or *pick-up-board* will have reached an activation level that surpasses the threshold. At that moment one agent - suppose it is *pick-up-sander* - becomes active and performs an action in the world.

Suppose that the action succeeded, such that now *S(t)=(hand-is-empty,sander-in-hand, board-somewhere).* The patterns of spreading of activation have changed now. The state influences different agents (more specifically, *pick-up-sander* receives less activation because one of its preconditions is no longer satisfied, and *put-down-sander* also receives some activation since one of its preconditions has become fulfilled). The input from the goals remains the same. Agents that now spread backwards are: *pick-up-sander*, *sand-board, put-down-board.* The agents that spread forward are: *put-down-sander, pick-up-board. Pick-up-board* will very soon have accumulated enough activation to become active (since *sand-board* spreads only to this one). Finally, after *pick-up-board* has become active, *sand-board* accumulates enough activation and is executable so that it also becomes active. As a result the goal is achieved.

4 Theory. Part ID: Mathematics

This section of the paper describes the theory in a rigorous, mathematical way so as to make the results reproducable. Given:

- a society of agents 1..n,
- a set of propositions P,
- a function S(t) returning the propositions that are observed to be true at time t (the state of the environment as perceived by the society); S being implemented by an independent simulator,
- a function G(t) returning the propositions that are a goal of the society at time t; G being implemented by an independent simulator,
- a function executable(i), which returns 1 if agent i is executable at trime t (i.e. all of the precondition list propositions of agent i are member of S(t)), and 0 otherwise.
- π: the total level of activation, $\pi \in R$,
- θ: the threshold of activation, $0 \le \theta \le \Pi$,
- φ: the percentage for forward spreading, $0 \le \varphi \le 1$,
- β: the percentage for backward spreading, $0 \le \beta \le 1$.
- γ: the percentage determining the relative input from the goals, versus the state, $0 \le \gamma \le 1$.

The external input of activation to agent i at time t is defined as follows:

$$\text{external-input}(i,t) = \sum_{j_i} S + \sum_{k_i} G$$

where j_i ranges over the predicates in the precondition list of agent i that are member of S(t) and k_i ranges over the predicates in the add list of agent i that are member

of G(t). Further S and G are chosen such that:

$$\sum_i \sum_{j_i} S = \Pi * (1 - \gamma)$$

$$\sum_i \sum_{k_i} G = \Pi * \gamma$$

where i ranges over the agents of the society.

The following equations specify what an agent s spreads backward (b) to an agent i and what an agent p spreads forward (f) to an agent i, given that s is successor of i and p is predecessor of i.

$$b(s,i,t) = \begin{cases} \sum_{s_i} B & \text{if executable}(s,t) = 0 \\ 0 & \text{if executable}(s,t) = 1 \end{cases}$$

$$f(p,i,t) = \begin{cases} \sum_{p_i} F & \text{if executable}(p,t) = 1 \\ 0 & \text{if executable}(p,t) = 0 \end{cases}$$

where s_i ranges over the preconditions of agent s which are not true at time t and which are in the add list of agent i and p_i ranges over the add list predicates of agent p which are not true at time t and which are in the precondition list of agent i. Further, B and F are chosen such that:

$$\sum_m \sum_{s_m} B = \alpha(s,t) * \beta$$

$$\sum_q \sum_{p_q} F = \alpha(p,t) * \varphi$$

where m ranges over the predecessors of agent s and q ranges over the successors of agent p and a(i,t) stands for the activation level of agent i at time t.

The following equations specify what an agent i retains for itself and what it receives from the other agents during spreading activation:

$$\text{retains}(i,a,t) = \begin{cases} a * (1 - \varphi) & \text{if executable}(i,t) = 1 \\ a * (1 - \beta) & \text{if executable}(i,t) = 0 \end{cases}$$

$$\text{receives}(i,t) = \sum_s b(s,i,t) + \sum_p f(p,i,t)$$

where s ranges over the successor of agent i and p ranges over the predecessors of agent i.

Finally, the activation level of an agent i at time t is defined as:

$$\alpha(i,0) = 0$$

$$\alpha(i,1) = \text{retains}(i,\text{external-input}(i,1),1) + \text{receives}(i,1)$$

$$\alpha(i,t) = \frac{\text{retains}(i,\text{external-input}(i,t) + \alpha(i,t-1),t) + \text{receives}(i,t)}{2}$$

The division by 2 takes care that the global activation level remains constant, namely equal to II The agent that becomes active at time t is agent i such that:

$$\alpha(i,t) \geq \theta \qquad 0)$$

$$\text{executable}(i,t) = 1 \qquad \text{(ii)}$$

$$\forall j \mid j \text{ fulfills (i)} \wedge \text{(ii)}: \alpha(i,t) \geq \alpha(j,t) \qquad \text{(iii)}$$

5 Results. Part I: Empirical

We are using two tools to test the theory: a computer simulation and a quantitative theory. This section discusses the results of the former one. A (computer-) environment has been built in which societies of agents can be denned and their behavior simulated. The program is written in Common-LISP on a SYMBOLICS machine. Figure 3 shows the graphical interface of the system. It is also possible to obtain a trace showing in detail how the spreading activation has evolved.

Several example applications have been experimented with:

• The example discussed in the prologue of Minsky's Society of the Mind book [10]. The global goal of this society is to build a tower of blocks as high as possible. It consists of 7 agents (find-place, lay-first-block, move-block, grasp-block, see-block, release-block, destroy-tower) and 26 links. The network has interesting features, such as loops, local high concentrations of links, and destructive agents (the destroy-tower agent).

• The example discussed in the planning chapter of Charniak and Mc Dermott [4]. A two-handed robot has to perform 2 tasks: paint itself with a sprayer and sand a board with a sander. This example involves 10 agents and 68 links. This task is already more complex: the robot has to coordinate the use of its hands or otherwise be clever enough to use a vise to hold the board and perform the jobs in parallel. An experiment with this example is shown in the bitmap.

• The well-known conflicting goals examples from the blocks world [11]. This is a real big network involving 18 agents and 594 links. An example of the kind of tasks the society has to perform is the achievement of the conjunctive goal *(and on-a-b on-b-c)* given that *on-a-c, b-on-table* and *c-on-table* are true.

All of the problems were solved for specific (large) ranges of parameters. The results of these empirical studies are discussed in the following subsections.

Planning Capabilities

The simulated societies exhibit certain planning capabilities. The notion of a plan is however very different from the classical one. A society does not build an explicit representation of one specific plan, but instead expresses its " intention" or "urge" to take certain actions by high activation levels of the corresponding agents. Societies are able to consider (to some extend) the effect of sequences of actions: if a sequence of agents exists which transforms the current state in the goal state, then this sequence becomes highly activated through the forward spreading (starting from the current state) and the backward spreading (starting from the goals). Goal-oriented planning behavior is obtained through the spreading of activation backward, while data-oriented (opportunistic) behavior through the spreading of activation forward. Local maxima in the selection of action can be avoided, provided that the spreading of activation can go on long enough (the threshold is high enough), so that the society can evolve towards the optimal activity pattern.

Conflicting or interacting goals can be dealt with. For example, the two goals of painting itself and sanding a board interact, since achieving them in parallel would require the agent to have three hands (one to hold the board, one to hold the sprayer and one to hold the sander). We obtained the three possible solutions to this problem (performing one task after the other or using the vise and doing both in parallel) for different selections of parameters. For some selections of parameters, the spreading activation is such that *put-board-in-vise* becomes active (it receives activation from *sand-board-in-vise* and is also urged to become active by *pick-up-sprayer* in order to make a hand free). For different selections of parameters the urge to fulfill either one of the goals is so strong that the society first performs one task and afterwards puts some tool down to start the other task.

The approach belongs to the class of reactive planners [6], [5], [8], [1]. There is no separate execution module: whenever an agent has accumulated enough activation it becomes active (and takes some real world actions). The system is completely 'open'. The environment may change during the process of action selection. Even the goals may change at run time. As a result the external input as well as the internal spreading activation patterns will change to reflect the modified situation. Even more, the external influence during the "planning** or spreading activation phase is so important that plans are only formed as long as the influence (or "perturbance") from the environment and goals is present.

An important difference with classical action systems as well as with most reactive planners is that there is no centralized preprogrammed search process. Instead, the operators (agents) themselves select the sequence of operators that are activated, and this in a non-hierarchical, highly distributed way. The difference between this work and the bulk of work in distributed planning and action [7] [2] is that in the latter the planning agents communicate on a much higher level.

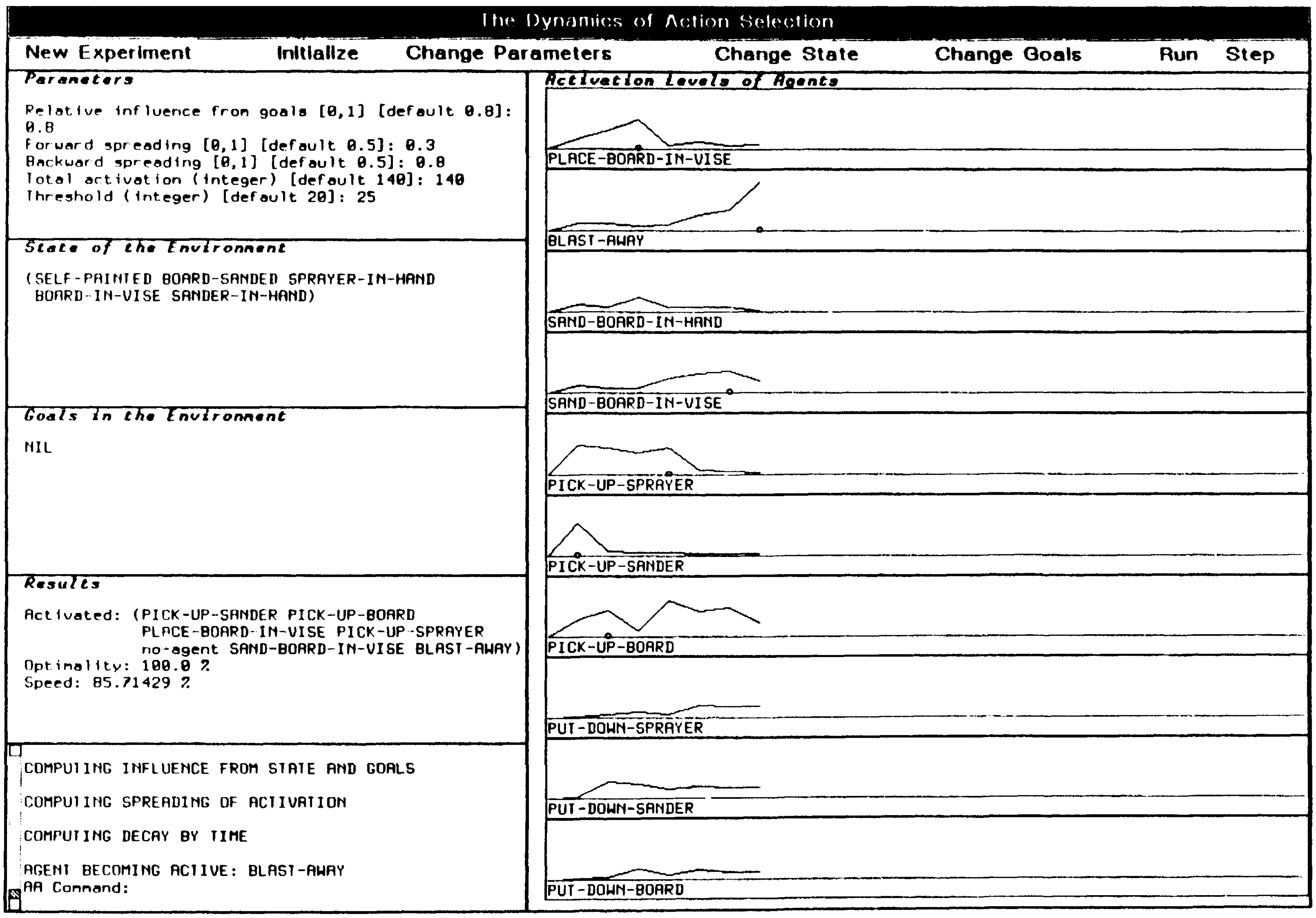

Fig. 3. The upper pane is a menu of commands. It makes it possible to define a new society, to initialize the current society, to change the global parameters, to change (manually) the current state of the environment, to change the goals of the society and to run or step through the behavior of a society. The left-hand panes display the parameters, the current state of the environment, the current goals of the society and the results of the simulation. The right hand-panes display the activation levels of agents over time. The little circles tell when an agent has become active.

They reason about each other, they debate and negotiate among one another. Another important difference is that there is no search tree constructed, i.e. there is no explicit representation built of the state changes after taking certain actions. Consequently, the system does not suffer from the disadvantages of search trees such as: that information is duplicated in several parts of a tree; trees grow exponentially with the size of the problem; trees only make a strict representation of plans possible (impossible to work with uncertainties); etc.

However, there are limitations to this form of planning:

- A society can only look ahead in a local neighborhood (in time) which is determined by the level of threshold. The higher the threshold the more a society can look ahead.
- There is only a very limited model of the environment. It is not possible to model how the environment most probably will change without the agent's interference, e.g. by other agents. Only the effects of own actions are represented. A society is however able to cope with the problems that this causes, because it performs a partial replanning at every timestep.
- There is no description of the past "search", i.e. no "memory" of past states neither locally, nor globally. As a consequence the same planning mistake can be made over and over again.
- The representation of the environment is oversimplified at this moment. Neither abstractions, nor variables are being used.
- Although the behavior is deterministic, it is impossible to exactly predict what will happen on a global level. Section 6 shows however that it is possible to construct a qualitative theory.

Varying the Global Parameters

By varying the global parameters the behavior can be tuned. Figure 4 shows the results of four experiments with different choices of parameters.

	1.	2.	3.	4.
percentage forward spreading φ	0.3	0.3	0.3	0.3
percentage backward spreading β	0.8	0.8	0.8	0.8
threshold θ	25	25	35	25
relative influence from goals γ	0.15	1.0	0.8	0.8
-------------------				
optimality	0.69	1.0	1.0	0.92
speed	0.65	0.65	0.59	0.7

Fig. 4. This table illustrates some of the effects of the input parameters on the global results showing the data for 4 experiments.

The behavior can be made more or less *data-oriented* in its selection of action, for example by varying the parameter γ (cf. column 1 and 2 in the above figure). The data-orientedness is proportional to the level of opportunism, or the degree to which a society exploits opportunities. The behavior can be made more or less *goal-oriented* by varying γ as above. For example, for $\gamma=1$, backward search is performed.

The behavior can be made more or less *deliberated* by increasing the threshold θ which makes the spreading activation process go on for a longer time before a specific action is selected. This allows the society to look ahead further (cf. column 3 and 4 in the above figure). The behavior can be made more or less *fast,* by varying the threshold θ as above. The resulting selection is however less optimal. It is still an open question how the values for these parameters should be selected. At the moment we set them ourselves such that the resulting behavior has the characteristics we want it to have. We envision to use a second society of agents which would tune the parameters of the first one so as to obtain the required level of data-orientedness, goal-orientedness, deliberation and speed. These agents would reflect the qualitative laws discussed in section 6.

Flexibility and Adaptivity

Because of the continuous "replanning" the behavior is flexible. Unforeseen situations (due to incomplete knowledge about the environment or the effects of actions) can be dealt with. For example, if after the activation of *pick-up-board,* the board is not in the robot's hand (e.g. because it slipped away), the same agent becomes active once more, because it still receives a lot of activation from the agents that want the board to be in the robots hand. Another example: if some other agent picks up the board and puts it in the vise, *pick-up-board* is no longer urged to become active. All of these experiments have been simulated with success.

The societies are also modular. It is possible to define new agents or to delete agents. The dynamics adapts to the new situation and the society still does whatever is in its possibilities. For example, when the agent *put-board-in-vise* is deleted, the society still comes up with a solution. The behavior also adapts to changes made to the agents, the environment or the goals. For example, an experiment was performed in which a third hand was given to the robot at run time. The robot immediately made use of it to realize its goals in a more efficient way. In another experiment *spray-paint-self* was given as extra precondition that the board had to be sanded. As a result, the society produced action sequences which fulfilled this new constraint. It even came to a solution faster because this constraint reduced its search space.

Impasses

One drawback of the theory is that for a certain selection of the parameters impasses emerge. They are not too frequent (around 8%), and come in two forms:

- loops, i.e. the same sequence of agents is activated over and over again.

- deadlocks, i.e. there is a converges to a situation in which none of the agents is strong enough to become active and all activation levels remain constant over time.

It is questionable whether a solution to impasses should be built in. The hypothesis could be adopted that in a real environment the state and goals will change anyhow after some time δt which is very small. This changes the spreading activation patterns and therefore would get the society out of its impasse. If we do want to avoid (even temporal) impasses, this cannot be guaranteed by a careful selection of the parameters. One very simple solution however could be to introduce some randomness in the system. Another solution might be to use the second network mentioned above to monitor possible loops or deadlocks in the first network.

6 Results. Part II: Qualitative Theory

A second tool that we are developing to study the proposed theory of action is a qualitative theory. The dynamics of the resulting systems are too complicated to be studied with a system of differential equations. So, the only possible formal theory is a qualitative one. The complete qualitative theory is discribed in [9]. We give in this section some idea of how it allows to (partially) predict, understand and tune the behavior of a society. A first type of laws is about the relationship between the parameters and the results. Examples of such laws are:

$$\theta \propto \frac{1}{\text{speed}} \qquad \text{(i)}$$

$$\theta \propto \text{optimality} \qquad \text{(ii)}$$

$$\beta \propto \text{goal-orientedness} \qquad \text{(iii)}$$

Law (ii) indicates that the optimaiity of the selection increases when the threshold increases , while (i) says that the selection becomes slower at the same time, (iii) says that the backward-spreading parameter may be used to control the goal-orientedness. A second type of laws is about the relationship between the structure of the network and the results. Some examples are (where '#' stands for 'the number of):

$$\#\ \text{agents-realizing-goal} \propto \text{time-necessary-for-goal} \qquad \text{(iv)}$$

$$\#\ \text{goals} \propto \frac{\quad}{\text{goal-orientedness}} \qquad \text{(v)}$$

$$ft\ \text{propositions-in-state} \propto \frac{\quad}{\text{data-one ntedness}}$$

$$\propto \frac{1}{\text{opportunism}} \qquad \text{(vi)}$$

Law (iv) states that the more agents there are that can realize a specific goal, the longer it takes for the society to realize that goal. So a goal which can only be achieved by one agent is realized very fast (cf. routines). When the society has many ways for realising a goal, the consideration of them makes it slower, (v) indicates that the more goals a society has the less goal-oriented it becomes (and therefore the more data-oriented and opportunistic), (vi) states that the behavior becomes less data-oriented (and thus also less opportunistic) when more things are observed in the environment.

7 Conclusions

The results reported upon in the paper demonstrate the feasability of using an activation/inhibition dynamics among agents proposing actions, to solve the problem of rational action selection. The algorithm discussed here is not meant to be the ultimate one. We are currently experimenting with variations of it. In particular we are trying to exploit the information in the delete lists of agents to better model negative interactions among agents. Another variation that is being worked upon is to allow parallellism in the activation of agents. Apart from this, future work will be concerned with: a systematic study of the limitations of the theory, the application of the theory in real mobile robots, and the introduction of learning.

Acknowledgements

Publications by, and/or discussions with M. Minsky, L. Steels, R. Brooks, J. Feldman and J.R. Anderson inspired and influenced this research. Financial support was received from the EEC (COST 13) and the Belgian Ministry of Science (Incentive Action in AI). The author is a Belgian National Science Foundation fellow.

References

[1] Agre P. and Chapman D. (1987) Pengi: An Implementation of a Theory of Activity. Proceedings of the Sixth National Conference on Artificial Intelligence, AAAI-87. Morgan Kaufmann, Los Altos, California.

[2] Bond A. and Gasser L. (1988) Readings in Distributed Artificial Intelligence. Morgan Kaufmann, San Mateo, California.

[3] Brooks R. (1985) A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation. Volume RA-2, Number 1.

[4] Charniak E. and Mc Dermott D. (1985). Introduction to Artificial Intelligence. Addison-Wesley.

[5] Firby R. (1987) An Investigation into Reactive Planning in Complex Domains. Proceedings of the Sixth National Conference on Artificial Intelligence, AAAI-87. Morgan Kaufmann, Los Altos, California.

[6] GeorgefT M. and Lansky A. (1987) Reactive Reasoning and Planning. Proceedings of the Sixth National Conference on Artificial Intelligence, AAAI-87. Morgan Kaufmann, Los Altos, California.

[7] Huhns M. (1987) Distributed Artificial Intelligence. Pitman, London.

[8] Kaelbling L. (1987) An Architecture for Intelligent Reactive Systems. In: Reasoning about Actions and Plans: Proceedings of the 1986 Workshop. Morgan Kaufmann, Los Altos, California.

[9] Maes P. (1989) A Qualitative Theory of the Dynamics of Action Selection. AI-memo 88-28, AI-LAB,