COOPERATION BETWEEN DISTRIBUTED AGENTS THROUGH SELF-ORGANISATION

Luc Steels

VUB AI Lab Pleinlaan 2. 1050 Brussels Belgium

The paper uses a behavior-based approach to tackle the problem of cooperation between distributed agents. It focuses on the use of self-organisation and dissipative structures to establish emergent functionality. Other principles such as randomness, the subsumption architecture, and the gradient field are also used. Results of computational experiments are presented.

1. INTRODUCTION

The paper discusses some central issues in the cooperation between distributed agents using the following case study:

> The objective is to explore a distant planet, more concretely to collect samples of a particular type of precious rock. The location of the rock samples is unknown in advance but they are typically clustered in certain spots. There is a vehicle that can drive around on the planet and later reenter the spacecraft to go back to earth. There is no detailed map of the terrain although it is known that the terrain is full of obstacles, hills, valleys, etc.

The case study is designed to require autonomy. It is not feasible to plan and steer the whole thing out of earth because communication from and to the planet has a considerable time delay and may be cut off during certain periods. Although one solution could be that the vehicle itself wanders around and collects the rocks, it is obvious that a larger terrain could be covered much more quickly if there is a group of mobile robots that perform the task of searching and carrying the rocks to the vehicle. This would make the solution also less fault tolerant because loss of one robot is not fatal. Because the desired samples are clustered in certain spots, the robots better cooperate to accomplish the task. This gives us the problem addressed in this paper: how can these distributed robots cooperate to find samples and to carry them to the collecting vehicle. If the present case study seems somewhat far fetched, the cleanup of toxic waste or household garbage collection can be viewed as comparable tasks.

Evaluation criteria

The following criteria will be used to evaluate different solutions. These are criteria relevant for AI systems in general but they are particularly appropriate for mobile robots operating on a distant planet.

1. Robustness: The system should be able to recover when a certain action is not correctly executed. For example, when a sample is not picked up although the instruction was given, this should not lead to further malfunction.

2. Graceful performance degradation: Loss of one robot should not be fatal although it could give decreased performance.

3. Flexibility: When conditions in the environment change, this should not require major changes or incapability to function. For example, the vehicle could move while the robots are searching around, the rock samples could be exhausted in one location requiring a rerouting of resources to explore another location, there could suddenly be new obstacles on a path between the samples and the vehicle.

4. Hardware economy: This refers to the complexity of the proposed hardware and the resources (e.g. in energy) that is needed to keep the hardware operational.

5. Cognitive economy: This refers to the amount of internal representations needed. The more complex this representation the less economical the system will be.

6. Communicative economy: This refers to the amount of information that needs to be exchanged between the subsystems involved to get the task done (e.g. between the robots or between the robots and the vehicle). Less communication is more desirable if only because all communication requires substantial amounts of equipment and processing.

7. Predictability. This refers to the amount of regularity that has to be present in the environment for the total system to keep functioning. Ideally the system should be able to cope with unpredictable situations.

8. Prior knowledge. This refers to what has to be known in advance to have a successful system. For example, does a map of the terrain have to be available? Less prior knowledge is more desirable.

There are also some quantitative criteria concerning the optimality of the solution. For each sample i there will be the time needed by a robot to find the sample $t_{s(i)}$, and the time needed by the same robot to carry the sample back to the vehicle $t_{c(i)}$. The *total time t(n) needed to carry a set of n samples is therefore equal to:*

$$t(n) = \sum_{i=1,n} t_s(i) + \sum_{i=1,n} t_c(i) \tag{1}$$

Let us assume that d is equal to the distance between the location of the sample and the location of the vehicle. We also assume that the time needed by a robot to cover one unit of distance is equal to one unit of time. Then the optimal performance with only one robot is given by

$$t(n) = 2nd \tag{2}$$

When *m robots are available, the optimality is given by*

$$t(N) = 2\frac{n}{m}d \tag{3}$$

Therefore if we have as many robots as samples, we can optimally carry all samples in as much time as it takes a single robot to carry one sample. This proves that parallelism helps (if we can keep the other design objectives such as fault tolerance, communicative economy, etc. optimal).

2. TWO PARADIGMS

Work on distributed cooperating agents started in the late seventies (Lenat (1975), Steels (1979), Lesser (1979), Smith (1980), Kornfeld and Hewitt (1981)). The basic assumption was that the agents were complex entities of the sort then proposed for a single artificial intelligence, i.e. each agent had substantial internal reasoning powers so that they could build up models of the world, relate and update these models with sensors that interpret the world, and communicate through complex messages for example to negotiate a solution or to exchange partial views of a plan. All of this was firmly within the symbolic AI tradition. More recently, there has been a revival of research on distributed agents, particularly in logic-based AI (see e.g. Konolidge (1982), Rosenschein and Genesereth, 1987). The reasoning powers of the agents are assumed to be even more pronounced than proposed in earlier work. Each agent is now a strictly rational agent using a logical description of the world and performing logical inference to plan its action and cooperate with other agents.

The logical approach

Applying the logic-based approach to our case study we would end up with the following proposal:

1. Each robot is equiped with a logical inference machine of the sort described by Genesereth and Nilson (1987). It has a representation of the world in the form of expressions in an extended form of predicate calculus (e.g. this representation records

the position with respect to other agents and the vehicle). The robot builds up this representation progressively by sensing and reasoning. It can plan its actions by reasoning logically about the world and about possible actions it could take. It performs temporal and spatial reasoning and has axiomatic representations of the preconditions and effects of an action. Thus the robot can form plans to decide in which direction it should move next, whether it should pick up some object, whether it should go back to the site on which it earlier found a rock, and so on. The robot is capable to translate these plans into concrete action by activating its effectors and adapt the plan when changes in the world have made it obsolete or when an action did not get executed as planned.

2. To handle the cooperative part, each robot has additional functionalities: It is able to engage in a dialog with other agents or the vehicle. This dialog involves exchanging something close to logical formula. It could for example ask another agent where it is located or whether it found something and where. Each agent develops not only a model of itself and its own actions but also of what other agents know and belief. It performs meta-level reasoning to think about how it should relate its own actions to other agents and when it should engage in communication or in cooperation.

The position taken in this paper is that the above scheme is entirely unrealistic for the following reasons:

1. The technological complexity required in each agent is too high. If the agent has to be equipped with a logical inference machine, it would need at least several megabytes of memory and a processor fast enough to execute the tens of thousands of logical inferences required per second. This implies that the agent has to carry a computer around of the size of a powerful workstation. This computer has to remain operational in difficult circumstances (assuming it survives the trip to the planet). To establish communication between agents, we would need moreover radio equipment, components that formulate requests to other agents, and components that decode requests and formulate answers. The more complex the nature of the messages between agents, the more complicated these components will be.

2. So far we have no working programs that demonstrate how an agent can extract a logical description of the world from currently available sensors, and worse no such programs seem to be forthcoming. Existing vision systems require vast amounts of computation which would have to be added to the logical inference machinery. Known algoritms do not get much further than (unreliably) extracting elementary information from the image, making some leading vision researchers question the idea whether a general purpose vision system that can extract a complete symbolic description from the image in real time is at all feasible (Ullman, 1987). Even such seemingly simple problems such as knowing where each agent is located are much more difficult than is commonly assumed.

3. Logic-based AI faces a set of fundamental problems which have not found adequate solutions and which would be needed in this application. These problems include the frame problem, effective reasoning about time and space (particularly the combination of the two), coping with the dynamics (and therefore non-monotonicity) of the world, and so on. These are the research problems that logic-based AI has been studying for the past few decades, but the fact that no solution seems within reach for non-toy problems indicates serious fundamental difficulties.

The behavior-based approach

An alternative to knowledge is behavior. Behavior is unconscious and without much deliberation or volition. It is typical for tasks involving perception, locomotion, speech and other skills. Nobody can explain for example how he is able to parse a speech signal, or which muscles he is moving to pick up a suitcase. Professional typists when asked cannot immediately give declarative descriptions of the keyboard although they exhibit behavior that implies 'knowing' something about the arrangement of the keys.[1] Skilled typing is therefore a clear case where behavior rather than knowledge seems to be at the heart of the task. AI research and engineering so far has stressed knowledge-based approaches. This is entirely appropriate for tasks such as computer configuration, theorem proving, or legal reasoning, where the problem solver is capable to explain how he is solving the problem and what knowledge he brings to bear. But it is not necessarily the case that all tasks related to intelligence should be solved using knowledge-based techniques. In any case this paper explores the use of behavior instead of knowledge to handle the task proposed in the case study.

[1] You can try this yourself (if you can type 'blindly'): What letter is under your right hand index finger on the middle bar?

3. CHARACTERISTICS OF BEHAVIORS

There is no consensus yet on what the distinctive features are of a behavior based approach to AI. In our own research we have been exploring the following characteristics (Steels, 1988)

1. ANALOGICAL REPRESENTATIONS

Logical descriptions of the world are avoided almost entirely. There is no such thing as a central fact base that contains a description of the relevant state of the world in terms of a set of facts (or possibly another equally symbolic representation, such as a semantic network or frames). Instead representations are exploited that are a lot closer to the world itself. These representations will be called analogical because they keep some part of what they represent *implicit* in the representation. For example, suppose we have to handle relations in space. Two representations could be used (see Fig 1):

- The symbolic representation uses facts based on predicates like *left-of*, or *position (x,y)*.
- The analogical representation uses a grid where the objects occupy positions mirroring the positions they take in the world.

left-of (object-1, object-2)
square (object-1)
round (object-2)

Fig 1. a. Symbolic representation. b. Analogical representation.

The main reason for using analogical representations is that they are closer to the sensors and require no (or a much simpler) categorisation of the world before they can be used. For example a sonar sensor or a camera, can give us a bitmap in which objects occur as blobs occupying a position analogous to the position they occupy in reality. A heat sensor gives us immediately an analogical and continuous representation of how warm it is.

2. ANALOGOUS DYNAMICS

Using mainly analogical representations makes it no longer possible to use logical inference as primary mechanism for decision making. The first thing we will do is directly link analogical representations with perception and action. A very simple example of this is a system with a thermometer that is linked to a valve. The higher the temperature the more the valve is closed. The thermometer is an analogical representation of temperature: the higher the temperature the higher the position of the thermometer. There is a direct connection between the world and this representation. There is also a direct connection between the representation and the action of the valve.

The second thing we could do is execute various operations over analogical representations to transform it and give us more information. One example of this is a gradient field which is established on an analogical representation of a terrain. The process of creating the gradient can be seen as a diffusion operation starting from the target and going around obstacles until it reaches the vehicle. The route followed by the vehicle is given by following the highest gradient towards the target. (see Payton (1989) for a similar example).

3. EMERGENT FUNCTIONALITY

It is typical for behaviors that the required functionality does not get established by explicit design but that it emerges as a side effect of (i) the internal dynamics and/or (ii) the (dynamical) interaction with the environment. A typical example of the first phenomenon is a tensile structure, such as the geodesic domes built by Buckminster Fuller, in which different stresses interact to reach equilibrium and thus a stable structure capable to carry weight. It is only when the last component is put in that the whole structure suddenly obtains its functionality (fig 2 from Kenner, 1976). This contrasts with a building constructed with bricks or concrete in which the capacity to carry load comes from the rigidity of the component materials.

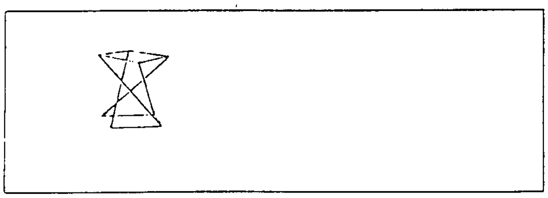

Fig 2. Tensile structure illustrating emergent functionality through internal dynamics.

An example of the second phenomenon is the forward movement of certain kinds of worms which are not capable to generate movement on their own, but when they are put in water with a certain density, the skin spontaneously starts to contract and stretch locally leading to forward movement (Oster, et.al. 1983). The movement is therefore a consequence of the interaction between the behavior of the medium in which the worm is located and the behavior of the skin.

In this paper, we will explore specifically how functionality can emerge based on self-organisation. Self-organisation is a technical term from the theory of complex systems (Prigogine (1976), Nicolis and Prigogine (1978)). It refers to a phenomenon in which a so-called dissipative structure spontaneously emerges out of the interaction of the behavior of many elements. A typical example is the formation of vortices in fluid flow. Self-organisation is now understood to be a fundamental process in biology (particularly in evolution, pattern formation, or the immune system), and in physics and chemistry (Babloyantz, 1987). There is no reason why it should not be of equal relevance to AI.

To get dissipative structures which are the product of self-organisation the following properties must be fulfilled:

1. There must be a basic underlying dynamical system which shows an equilibrium behavior, i.e. the system keeps evolving until it reaches a state of rest. The state of rest is still dynamic from the viewpoint of the underlying objects but there are no major macroscopic properties of the system changing. For example, when two closed containers with different gas pressures are connected, the joint system will evolve until there is equal pressure in both containers, although the movement of the molecules responsible for the pressure is still going on.

2. The dynamical system must be exposed to an outside disturbance, i.e. it must be open. In the case of the water flow, the disturbance is the force causing the fluid to flow. In the case of a selectionist system, the disturbance is environmental pressure.

3. The dissipative structure forms itself in response to this external force. It feeds on itself, which implies that the dynamics is going to be non-linear, and it is present as long as the force is present. This implies that there is not only a mechanism that establishes structure but also one that breaks it down again. Thus the vortices in the water will be present as long as the water keeps flowing at a rapid rate. When the flow is no longer there, the water will move back to an equilibrium state.

All of this is mapped onto the case study as follows. We will design a system of interacting robots whose equilibrium behavior consists in exploring the terrain around the vehicle. The presence of rock samples constitutes a disturbance. The desired dissipative structure consists of a spatial structure (i.e. a path) formed by the robots between the samples and the vehicle. This structure should spontaneously emerge when rock samples are present, it should enforce itself to maximise performance and should disappear when all samples have been collected.

4. STEP 1: THE IMPORTANCE OF RANDOMNESS

Before we can build this system we have to get the basic behaviors of the agents implemented. Some solutions, which may seem obvious at first, have to be rejected if we want to maximise the design criteria stated in the beginning. For example, a complete map of the terrain and knowledge where a robot or the vehicle are located on this map, or a complete vision system that is telling us what kind of object a robot has in front of it and what the exact boundaries are of this object, are luxuries that will not be assumed to be present.

Instead we start from the following simple behaviors:

Movement behavior

1. Choose randomly a direction to move.

2. Move in that direction

Handling behavior

1. If I sense a sample and am not carrying one, I pick it up.

2. If I sense the vehicle-platform and am carrying a sample, I drop it.

The handling behavior establishes a direct connection between sensors and effectors and runs parallel with the movement behavior. There is no sophisticated control strategy or internal reasoning of any sort. To handle obstacles, an additional behavior is added:

Obstacle avoidance

If I sense an obstacle in front, I make a random turn.

The moving behavior and the obstacle avoidance behavior are coupled using a subsumption architecture (Brooks, 1987), i.e. when there is an obstacle, obstacle avoidance behavior overrides random movement (fig 3).

Fig 3. Subsumption architecture. Bottom-level layers take precedence over top-layers.

Will this solution solve the problem? Yes and no. The robots clearly perform a random walk or Brownian movement. Hence they will accidentally stumble into the samples and later on find back the vehicle. The following is a classical theorem from probability theory:

The random walk theorem: *Starting from any point in a random walk restricted to a finite space, we can reach any other point any number of times.* (Chung, 1974).

It follows from this theorem that the robots will find the rock and they will find back the vehicle. The only problem is that it may take a very large amount of time. Nevertheless this solution has some of the features that we want:

1. There is robustness. If a sample sensor malfunctions temporarily no harm is done. If the pick up action did not get executed properly, the sample sensor will still be on and the robot will pick it up again. If the vehicle-platform sensor malfunctions, the robot will wander around until it stumbles again on the platform. If the sample is accidentally dropped it will be picked up again or possibly found by another robot which will then carry it to the vehicle.

2. There is flexibility because changes in the vehicle, the rock samples or sudden obstacles can all be handled.

3. There is extreme cognitive and communicative economy, and no prior world knowledge is required.

4. Most importantly, parallelism can be exploited without additional overhead.

To understand how parallelism helps, it is useful to go back to equation (1):

$$t(n) = \sum_{i=1,n} t_s(i) + \sum_{i=1,n} t_c(i) \quad (4)$$

Clearly when the number of robots increases, the probability that one of them will reach a particular point in the space increases as well. So, if the number of robots increases to a (possibly very large) number M, we may evolve towards the theoretical optimum for the needed exploration time:

$$\lim_{1 \to M} \sum_{i=1,n} t_s(i) = nd \quad (5)$$

However no improvement with parallelism can be expected for the corresponding carrying time t_c because each robot carrying a sample must find its own way back. This puts an important limit on this solution. It shows that to optimise the total time we necessarily need more mechanism. It is not enough to just multiply the number of robots.

To test these theoretical insights, a number of experiments were conducted to study the relation between the number of robots and the overall time needed to solve the task. These experiments were implemented in an actor-based complex dynamics language RDL (Steels, 1989)[2]. The interface to the system is shown in fig 4. There are different windows each displaying one aspect of the simulation. One window called ROBOT shows the positions of the different robots. The size of the square indicates how many robots there are in that location. A second window called CARRYING-SAMPLE shows the positions of the robots carrying samples. Other windows show the remaining rock samples, and the collected samples already at the vehicle.

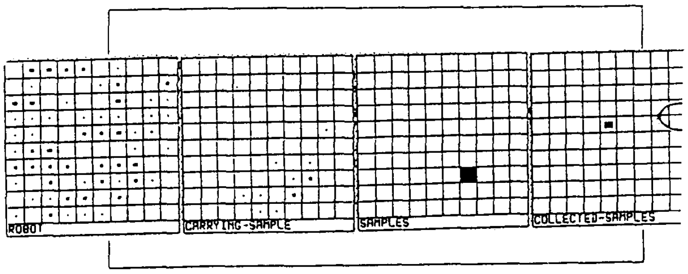

Fig 4. Snapshot of simulation environment.

The number of samples n is equal to 100. Starting with 8 agents, parallelism was raised by a factor of 2 up to 256 agents. The results of the experiments are displayed in fig 5. The *x-axis represents* 4×2^x. *The y-axis* gives the average time for 50 experiments. 8 agents require close to 30.000 time steps which decreases to about 8000 time steps for 256 agents. As expected we see a steady but limiting increase in the overall efficiency as parallelism increases.

[2] Development of this language is partly sponsored by ESPRIT project P440.

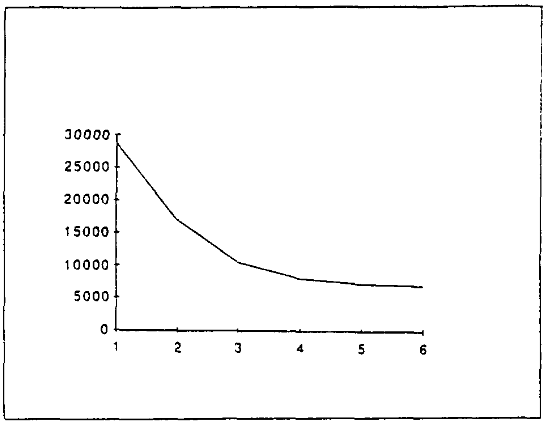

Fig 5. Relation between parallelism and efficiency.

The results are better understood when the average *total time needed t(n)* is decomposed into t_r and t_e, as in fig 6. As predicted, t_e decreases but t_r remains constant at about 7800 which is the average time for a robot to find back the way to the vehicle.

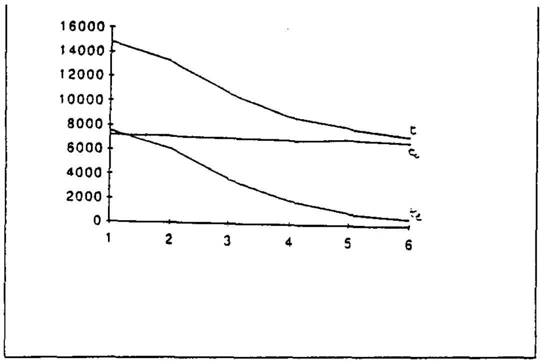

Fig 6. Decomposed parallelism relation.

5. STEP 2: THE GRADIENT FIELD

A gradient field is a field emanating from a certain point and diminishing in strength as the distance to the point increases. *Gradient fields are extremely common in* nature. They play a key role for example in the formation of structures such as shells, plants, and neural networks (Meinhardt and Gierer, 1982). Physical phenomena such as magnetism, radiation, or diffusion are all instances of mechanisms that *establish* a gradient field.

A gradient field will be used here to solve the following problems:

1. We have seen that t_r is not optimised through parallelism. The gradient field must solve this problem. It must be a guide to the robots how to get back to the vehicle.

2. We also want the robots to keep moving within a bounded area around the vehicle. *This will increase the probability that they will find* the samples in that area. When the vehicle moves forward, the robots should move along with it.

A *gradient field is established through a diffusion* process. Let p =<i,j> be a position in a discretised spatial representation, and n(p) the von Neumann neighbourhood of p = <i,j>:

$$n(p)=\{p' \mid p' = \langle i+k, j+q\rangle, -1 \le k \le 1, -1 \le q \le 1, p' \ne p\} \qquad (7)$$

Let g (t,p), a non-negative integer, represent the gradient at a time t in a position p. The diffusion processes used in the experiments is defined using the following difference equation:

$$g(t_k+1, \langle i,j\rangle) = \sum_{m \in n(i)} \frac{g(t_k, m)}{8} \qquad (8)$$

Boundaries are assumed to be dissipative. An additional rule helps further in the decay process:

$$g(t_i+1, p) = 0 \text{ if } g(t_i, p) \le 8 \qquad (9)$$

Note some important consequences of this diffusion process:

1. There is a decay, consequently continuous supply of g is needed.

2. *The extent of the field depends on the amount of g* originally available. This implies that we can engineer the system to specify how far robots may wander away from the vehicle by increasing g.

A physical way to implement g is by the emission of a sound wave from the vehicle. Increasing g then means increasing the sound level.[3] Each robot must have a sensor that can detect the sound and determine the direction *where it came from. If* the vehicle, which is responsible for emitting g, moves, the field will move along.

Let us now turn to the behaviors *that use the principle of the gradient field.*

The robot can be in two modes:

- Exploration: In this mode the robot moves away from the sound source, i.e. it follows the lower gradient.
- Return: The robot moves back to the sound source, i.e. it follows the higher gradient.

[3] An alternative, which was also verified in computational experiments, is that the robots put themselves markers on the ground as they are exploring the area around the vehicle. There also then needs to be a process of collecting the markers similar to the mechanism that we will discuss in section 6.

These modes determine two additional moving behaviors:

Return movement

If I am in return mode I chose the direction of highest gradient.

Explore movement

If I am in exploration mode I chose the direction with the lowest gradient.

The following behavior regulates which mode the robot will be in:

Mode determination

1. If I am in exploration mode and I sense no lower concentration than the concentration in the cell on which I am located, I put myself in return mode.

2. If I am in return mode and I am at the vehicle-platform I put myself in exploration mode.

3. If I am holding a sample, I am in return mode.

These behaviors are coupled using a subsumption architecture to the other moving behaviors as in fig 7:

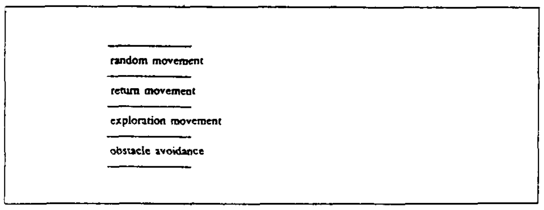

Fig 7. Subsumption relation between moving behaviors.

Fig 7 illustrates what happens when there are no rock samples. On the left we show an additional window called SOUND that implements g. Next to it there are a series of successive snapshots showing the distribution of the robots.

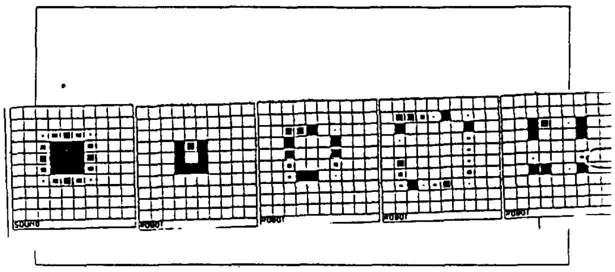

Fig 8. Exploration around the vehicle in waves.

There is clearly a pulsating wave of robots coming out of the vehicle until they reach the boundaries of the SOUND diffused from the vehicle located in the middle of the space and returning back to the vehicle. The robots clearly explore systematically the area around the vehicle. When the vehicle moves the SOUND area will also move and the robots will be attracted to the new location of the vehicle. Given that the movement is not too fast none of them will be lost. If one of them is lost it will fall back on random behavior.

When a rock sample is discovered, it will efficiently be brought back to the vehicle guided by the gradient field.

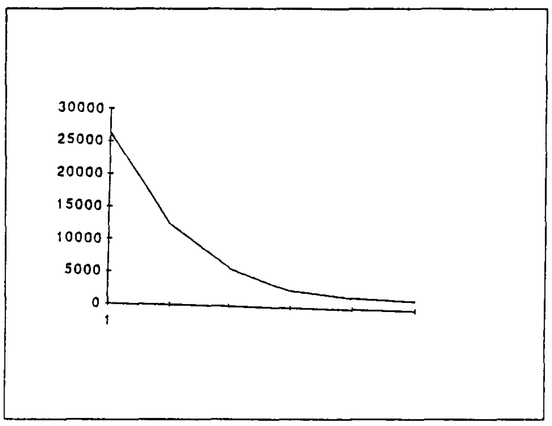

Fig 9. Absolute times with gradient fields.

The following experiments confirm this. Fig 9 represents again the evolution in (average) absolute time with parallelism increasing from 8 to 256 by a factor of 2. The time needed t(n) drops from about 25000 to about 500. Fig 10 gives the decomposition in t_s and t_c. We see that the second term in the computation t_c has dropped dramatically to be constant and equal to the distance between the sample location and the vehicle.

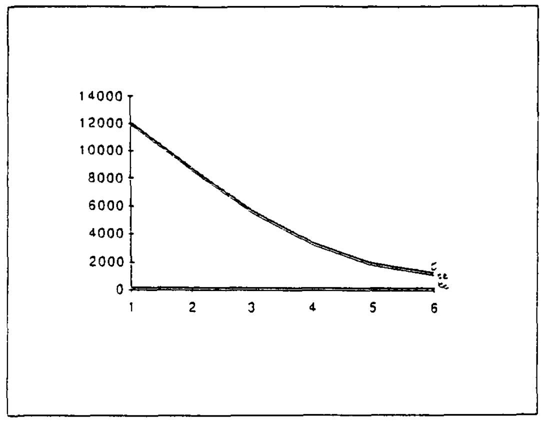

Fig 10. Decomposition of average time with gradient fields.

6. STEP 3: SELF-ORGANISATION

We are now ready to focus on how the principle of self-organisation could be applied. Specifically we want to optimise the first term t_s by establishing cooperative behavior between the distributed agents. When one robot has discovered the samples, it should communicate its finding to others and efficiently establish a path between the sample location and the vehicle. The means of communication between the robots will be a trail composed of 'crumbs'. The crumbs should be easily detectable by a robot sensor. Each robot has a certain supply of these crumbs and is capable to put them down and pick them up.

The following behaviors are added to the repertoire of the mobile robots to make this all happen:

Crumb handling

1. If I carry a sample, I drop 2 crumbs.

2. If I carry no sample and crumbs are detected, I pick up one crumb.

(1) will establish the path. (2) will break it down again. Note that the speed of break-down is less than the speed of buildup, although it does not really matter how much. We also need a new movement behavior:

Path attraction

If I am not carrying a sample and I sense crumbs, I move towards the highest concentration of crumbs.

This behavior is integrated with the other movement behaviors using the subsumption architecture (fig 11).

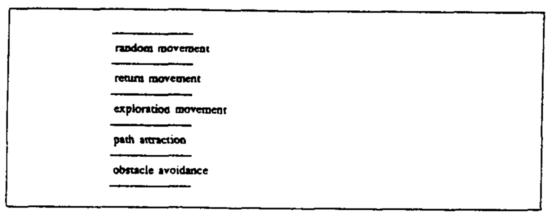

Fig 11. Subsumption relation between moving behaviors.

Fig 12 contains a few snapshots of a simulation in progress. Fig 12 a. is taken when one robot has discovered the samples which are located in the bottom right. Consequently there is a small square in the CARRYING-SAMPLE window at that position. The robots are scattered around randomly in the area to be explored around the vehicle.

Fig 12 b. illustrates the formation of a path. A second robot has now discovered the sample and the first one has arrived back at the vehicle (given a first small square in the COLLECTED-SAMPLES window). Note that crumbs have been put down by the first robot and that there is already a grouping of the robots because of attraction to the crumbs path.

In fig 12 c. we see several of these robots coming back with samples to the vehicle and the set of collected samples has already increased. Notice however that the crumbs oath has been broken down considerably because more robots without robots.

samples are walking along the path to the samples (and thus breaking down the samples) than there are robots returning with samples and thus reestablishing the path.

Fig 12 d. shows that the crumbs path has been reestablished again by returning robots. We also see that robots now maximally cooperate in the task. There is only one robot that is not on the path.

In fig 12 e. the samples are exhausted and the path is being broken down again. Robots are already scattered around in search of new samples. In fig 12 f. the system has returned to the original equilibrium behavior.

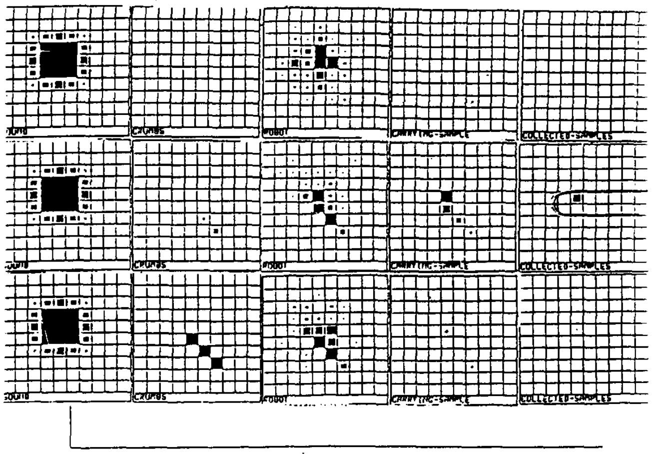

Fig 12a. Different snapshots of simulation in progress.

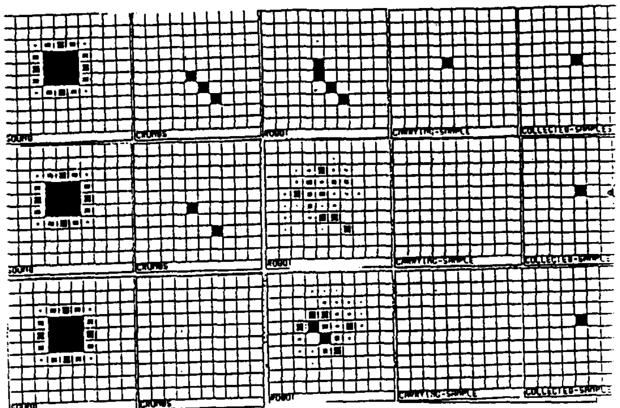

Fig 12b. Different snapshots of simulation in progress.

The experimental data confirm that almost optimal performance is reached after a few time steps. Fig 13 contains again the average times t(n) needed by an increasing number of robots (going from 8 to 256). We go from about 2500 timesteps which is a drastic reduction from 30000 timesteps to slightly under 1000 time steps with 256

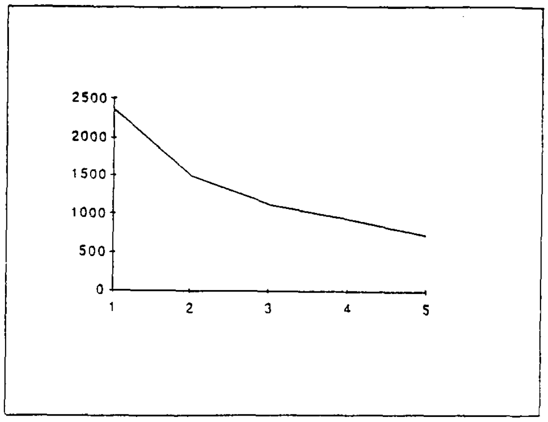

Fig 13. Total time with dissipative structure in place.

Further inspection reveals that most of the remaining non-productive time is spent in the initial phase when the first sample needs to be found. Once the sample is found, an efficient path is established fairly quickly even with a small group of robots. To illustrate this better fig 14 shows the exploration time t_e needed for each sample in one experiment with 16 robots. The x-axis represents the discovery order of a sample. The y-axis represents the time needed to discover a particular sample. Starting from roughly the 17th sample, the time is equal to the distance between the vehicle and the sample (except for sample 26 and 27 which were discovered by robots with a longer discovery path). The initial search phase can only be made smaller when more

robots are available to scan the area around the vehicle.

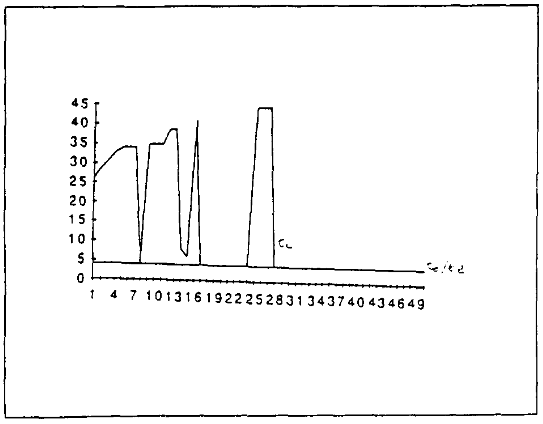

Fig 14. Time needed to discover each sample.

This experiment illustrates particularly how functionality can be made to emerge:

1. The *strengthening* of a path is not programmed in but is due to the non-linear interaction of the behaviors: Robots not carrying samples are attracted by the path which increases the chances that they will arrive at a sample and on return contribute to the establishment of the path.

2. The *breakdown of the path* also follows from the interaction between the environment and the robots. When there are no more samples, this will lead to less crumbs being put down, and because there is a depletion process the path will eventually disappear.

3. Also the fact that robots follow the path towards the vehicle and not away from it is, surprisingly, not programmed in but follows from the fact that on average more crumbs will be located nearer to the rock samples. Most robots by the way start on the path from the vehicle to which they return due to the pulsating in/out movement.

7. CONCLUSIONS

The solution we arrived at scores high on the various evaluation criteria proposed the beginning of this paper:

1. The solution is robust because the improper execution of an action is not fatal to the whole system.

2. The solution is fault tolerant, because loss of a robot is not fatal either although loss of the vehicle would of course be fatal. If there are fewer robots this will lead to a graceful performance degradation, but even with two robots (or one) the total system still remains operational.

3. The solution is flexible. Environmental changes have little impact. Any kind of object can be on the way between the robot and the vehicle.

4. There is extreme cognitive economy. The internal representation consists only of the state of the robot (exploration mode or return mode). There are no complex representations of time or space, no explicit representations of action, no representation of where the other robots are or where the vehicle is.

5. There is extreme communicative economy. The robots use the world to communicate among themselves: They leave markers behind in the world. Never is there any point to point communication. There are no complex messages to formulate or decode.

6. The environment does not have to be very predictable. For example, things would work equally well when there are two or more sources of rock samples.

7. Finally no prior knowledge in the form of terrain maps or other information is needed.

The robots themselves are definitely constructable using current technology. They do not require very sophisticated sensors or effectors. The assumed subsumption architecture has been well tested now in a number of applications. The solution satisfies also the quantitative criteria. We have moved from a very inefficient solution involving random search to an efficient cooperation between the robots as soon as a sample has been found.

The real importance of this paper is however not in the adequacy of the found solution but in the underlying principles. These principles can be summed up as follows: We have a number of dynamical mechanisms at our disposal: (partially) random movement, a gradient field, a dissipative structure. Each of these mechanisms is dynamical in the sense that it depends on the non-linear interactions with the environment. Structures grow dynamically and decay again. Although gradient fields have been used in other work, this is the first paper experimentally illustrating the use of dissipative structures. The idea was already proposed in Steels (1987) which contains suggestions on other applications.

The relation to ant societies is obvious but not as straightforward as one may think. Many ant societies use individual recruiting (one ant fetches another one), and if they do mass recruiting they are typically much less effective (30 % or even 5 %) than the robot ecology proposed in this paper (Deneubourg, et.al. 1985).

8. ACKNOWLEDGEMENT

I am indebted to several members of the VUB AI lab dynamics group (particularly Bernard Manderick and Jo Decuyper) for help in understanding complex dynamics in its many manifestations, and to Pattie Maes and Walter Van de Velde for their invitation to the Lagos Workshop on Autonomous Agents which prompted the writing of this paper.

9. REFERENCES

Babloyantz, A. (1986) Molecules, dynamics, and life. An Introduction to Self-organization of matter. Wiley, New York.

Brooks, R. (1986) A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation, 2:14-23, 1986.

Chung, K. L. (1974) Elementary Probability Theory with Stochastic Processes. Springer-Verlag, New York.

Deneubourg, J., J. Pasteels and J. Verhaeghe (1983) Probabilistic Behaviour in Ants: A Strategy of Errors? In J. Theor. Biol. (1983) 105, 259-271.

Genesereth, M. and N. Nilsson (1987) Logical foundations of artificial intelligence. Kaufmann, Los Altos.

Kenner, H. (1976) Geodesic math and how to use it. University of California Press. Berkeley.

Konolidge, K. (1982) A first-order formalization of knowledge and action for a multi-agent planning system. Machine intelligence 10. Ellis Horwood, Chichester.

Kornfeld, W. and C. Hewitt (1981) The Scientific Community Metaphor. IEEE Transactions on Systems, Man, and Cybernetics. SMCS-11(1):24-33, January 1981.

Lenat, D. (1975) Beings: Knowledge as interacting experts. Proceeding IJCAI-4, pp. 126-133. Kauffman, Los Angeles.

Lesser, V. et.al. (1979) Working papers in distributed computation: Cooperative distributed problem solving. Dept Comp. Science Amherst.

Nicolis, G. and I. Prigogine (1977) Self-Organization in Nonequilibrium Systems. Wiley-Interscience, New York.

Oster, G.F., J.D. Murray, and A.K.Harris (1983) Mechanical Aspects of Mesenchymal morphogenesis. Journal Empryol. exp. Morph. 78, 83-125.

Payton, D. (1989) Internalized Plans: a representation for action resources. Hughes AI Center, Calabasas, Ca.

Prigogine, I. (1980) From Being to Becoming. Freeman, San Francisco.

Rosenschein, S. and M. Genesereth (1987) Communication and Cooperation Among Logic-Based Agents. In Proceedings of the 6th Conference on Computers and Communications, p. 594-600. Phoenix.

Smith, R. (1980) The contract net protocol: High-level commjnication and control in a distributed problem solver. IEEE Trans. Comput. vol. C-29, pp. 1104-1113.

Steels, L. (1979) Reasoning modeled as a society of communicating experts. MIT AI Lab. Technical report. Cambridge, Ma.

Steels, L. (1987) AI and complex dynamics. Proceedings of the IFIP workshop on concepts and techniques for knowledge-based systems. Mount Fuji, Japan.

Steels, L. (1988) Steps towards common sense. In ECAI-88, Munchen. Pitman Pub. London.

Steels, L. (1989) CDL: A complex dynamics language based on localised actors. VUB AI Lab Memo 89-5.