Cooperative behavior in simulated reactive robots

D.J.M. Willems W.F.G. Haselager

Dpt. of AI / Cognitive Science, NICI - University of Nijmegen,
P.O. Box 9104, 6500 HE Nijmegen, The Netherlands

Abstract

An application was developed with which one can create and test simulations of reactive robots in simple environments. By means of this application we investigated the development of apparently plan-like collective behavior that originated out of simple behavioral structures without involving central plans or strategies. Under the appropriate circumstances seemingly cooperative and strategic behavior emerged.

1. Introduction

One of us (Willems, [8]) developed a Java/XML based application for constructing simulated robots and environments. This application allows the user to create different kinds of robots and environments and is available for use over the internet (http://enterprise2.cogsci.kun.nl/robotics/). In this paper we will examine the interaction between two kinds of simulated robots, the Unies and the Wumpi[i], in different environments. Unies are autonomous robots that the Wumpus can eat. A Wumpus remains near the food sources which the Unies need to sustain themselves. The Wumpus is faster than the Unies, and thus it seems that one of the few ways for the Unies to get access to the food is by means of some sort of cooperation, e.g. *decoy behavior*, in which one of the Unies entices the Wumpus away from the food, so that the others can eat (see figure 1).

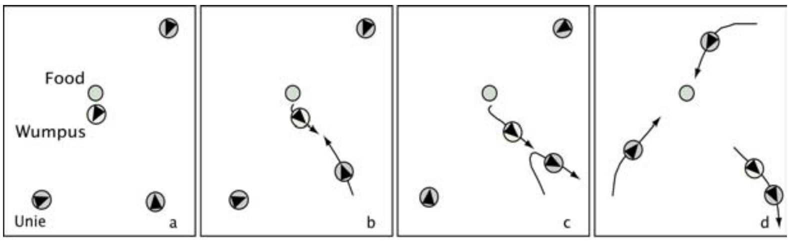

***Figure 1**: Decoy Behavior*
These diagrams show the expected decoy behavior of the Unies and a Wumpus near a Unie-food source. a) The Unies approach a food source but hold back because the Wumpus is near the food, b) The hungriest Unie comes near enough to the Wumpus for it to start the chase. c) The Unie senses that the Wumpus comes too near and reverses its direction. The Wumpus gets nearer to the Unie. d) The Wumpus chases one of the Unies. Because the Wumpus is farther away from the food, the Unies that are not being chased can approach and eat the food.

This type of behavior is interesting in that it seems to require collective plan formation and role taking. Thus, representational resources and communicative abilities seem to be needed. However, we aimed at implementing a non-representational approach to the design of the two kinds of robots using the subsumption architecture put forward by Brooks [1].

Our goal in this paper is to show that apparent cooperative behavior can be the result of the complexity of the environment. In this we attempt to follow Herbert Simon who claimed that the complexity of the behavior of the system (e.g. an ant on a beach) need not result from the complexity of the system, but rather can be a reflection of the complexity of its environment [6].

2. The simulation

In this section we will give a description of the application with which we created our simulations. A more exhaustive description is given in the user manual [7]. The application generates simulations of variable duration. It calculates the position of every object per time step. The result of this process is a simulation of the movement of objects in the environment. The simulation is rendered in two dimensions, but the simulation is actually three dimensional, objects do have height. Output of the application is in the form of a movie-like animation that can be viewed with the Play Back Panel (see figure 2).

The Play Back Panel has a complex user interface. The top part shows the environment containing obstacles, robots and food. Below the environment are the three time controlling buttons, the Start, Pause, and Stop buttons. At the top right of these buttons is a slider controlling the magnification of the environment. Below these controls is the time line that shows the camera events (zooming and moving), annotations, and movies. The application provides the possibility of annotating the simulation (providing time-dependent remarks) and of extracting movies in QuickTime format, as well as cartoon-like illustrations (such as figure 5 in this paper) that can be used in web sites, presentations and papers.

2.1. Robots and environments in XML and Java

The robots and environments are defined in XML files that can be created and modified by the user. The properties of the robots that can be defined in the definition file are for instance the size and height of the robot, their sensors and effectors, the color with which the robot is represented in the simulation, and most importantly the structure of the behavioral control system.

When a definition file is imported into the application the behavioral elements are first constructed out of the attributes that are indicated in the file, and then these elements are connected to each other as prescribed. The different behavioral elements are defined in Java. The behavior of separate behavioral elements cannot be changed within the definition file. If the user wants to add new behavioral elements such as new finite state machines, then he has to define these himself by implementing a new Java class that extends the abstract FiniteStateMachine class. For a more exhaustive explanation see the user manual of the application [7]. Environments are characterized in terms of kind and amount of obstacles, and amount of food.

Obstacles may vary in size from a bit smaller than a robot to much larger than a robot. The user can define multiple sets of obstacles with specific locations and standard deviations in the environment. The object definitions have to be written in Java and extend (subclass) the Java class Obstacle.

2.2. Simulations of reactive (subsumption architecture based) robots

For the control of the robot's behavior, we used the subsumption architecture as developed by Brooks [1, 2], the basic features of which we assume to be well known. We will limit ourselves here to a brief presentation of the control architecture of the Unies that we used in our simulations.

Four sensors were implemented for our simulations: a smell sensor to locate food and robots, a sonar sensor for avoiding obstacles, a touch sensor, and an energy sensor,

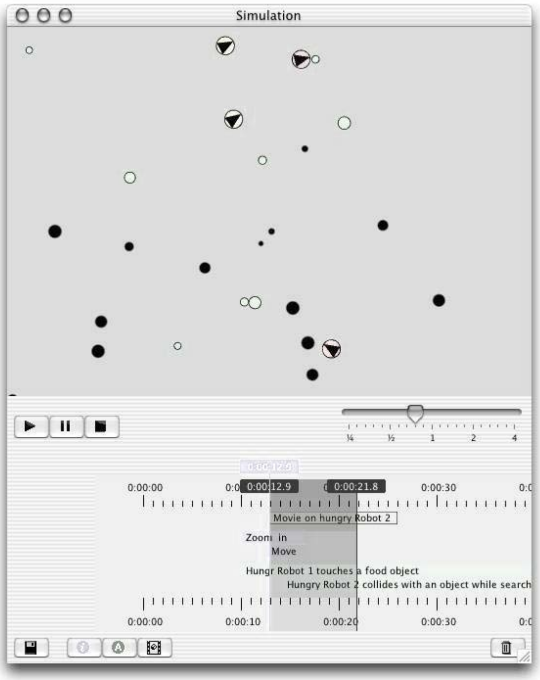

***Figure 2**: The Play Back Panel*

to measure how much energy is available to the robot. The robots have an energy depot which determines how much energy they have left. If the robot is eating food, the energy level will increase; when the energy is depleted the robot dies. The robots have three types of effectors; a speed effector for forward motion, a rotational effector for changing direction, and a feeding effector.

The Unie consists of behavioral layers that are constructed out of interconnected finite state machines. Interesting in this particular design is the finite state machine that is able to decide when the feeling of hunger takes precedence over the need to flee from the Wumpus (in the 'Wumpus' layer, see figure 3).

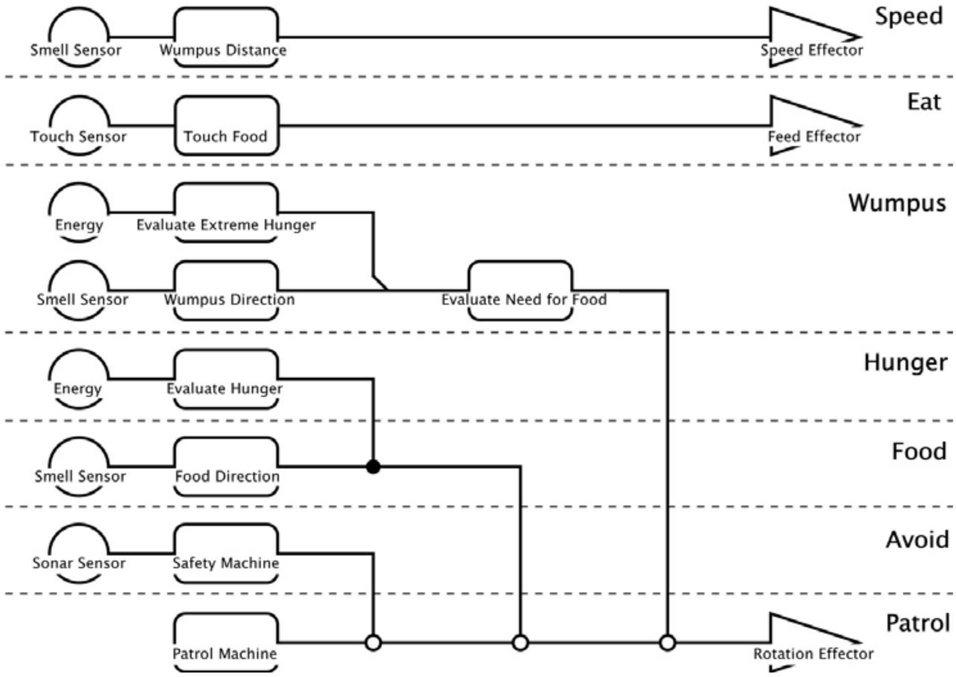

Figure 3: *The Unie architecture*

It makes this decision on the basis of the strength of the smell of the Wumpus and its relative energy level (relative to its maximum energy). The messages from the Evaluate Extreme Hunger machine are added to the messages from the Wumpus Direction machine. Both messages are then received by the Evaluate Need for Food machine that determines whether the original message from the Wumpus Direction machine gets passed on to the Patrol layer (see figure 3).

3. Types of behavior measured

To count a situation as a decoy situation, the following two conditions must be met. First, the Wumpus must be lured away from the food source. This implies that the Wumpus must be close to the food source when starting to chase a Unie. Second, at least one of the other Unies should be able to eat while the Wumpus is chasing a Unie. The decoy Unie may be caught by the Wumpus or it may escape. The first of these situations (when the Unie is caught) was described as *sacrifice* behavior and the second as *escape* behavior. Both of these situations were evaluated as an instance of *decoy* behavior, but they were counted individually. We also measured the relative occurrences of *useless death*s; the number of times a Unie lured the Wumpus away from the food source and was caught, but no Unies were able to eat. Thus, we have three situations for which we specifically searched in the test simulations (see figure 4). Moreover, we measured the times of death of each robot and the type of death of each robot (random death, hunger or as a decoy Unie). Actions such as the Wumpus chasing a Unie, or a Unie eating a food source were automatically noted by the simulation application.

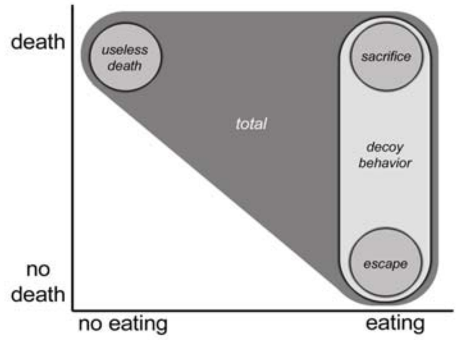

***Figure 4**: Types of behaviors*

4. The simulations

Five simulations were run per simulated environment, each with six Unies and one Wumpus. We simulated four different environments, differing in number of objects and food sources (see table 1). In the remainder of this paper we will limit our presentation to the rain-forest and desert environments.

	Number of objects		
	Food Sources	Obstacles	Surface-Type
Environment			
Desert	1	0	normal
Plains	2	0	normal
Forest	5	40	normal
Rain Forest	10	150	normal

***Table 1**: The properties of the simulated environments*

Before going on to the quantitative analyses of the results, we will present a representative case of a simulation in some detail (see figure 5).

The Unies in figure 5 show the sacrifice type of decoy behavior in a desert environment. In the first frame you can see the Wumpus circling the food source. The top left Unie is hungry and will approach the food source close enough for the Wumpus to detect the Unie (frame 2). The Wumpus will start to chase the Unie which will try to escape. It will eventually catch the Unie because the Wumpus is faster (frame 4). The Unie is caught far away from the food source, thus the other Unies can approach the food. Only one of the three Unies gets a chance to eat because it blocks the food source for the other Unies (frame 5). When the Wumpus has finished eating it will return to the food source chasing the Unies away. The Wumpus will then resume its circling behavior around the food source until one of the Unies is again hungry enough to approach the food (frame 9).

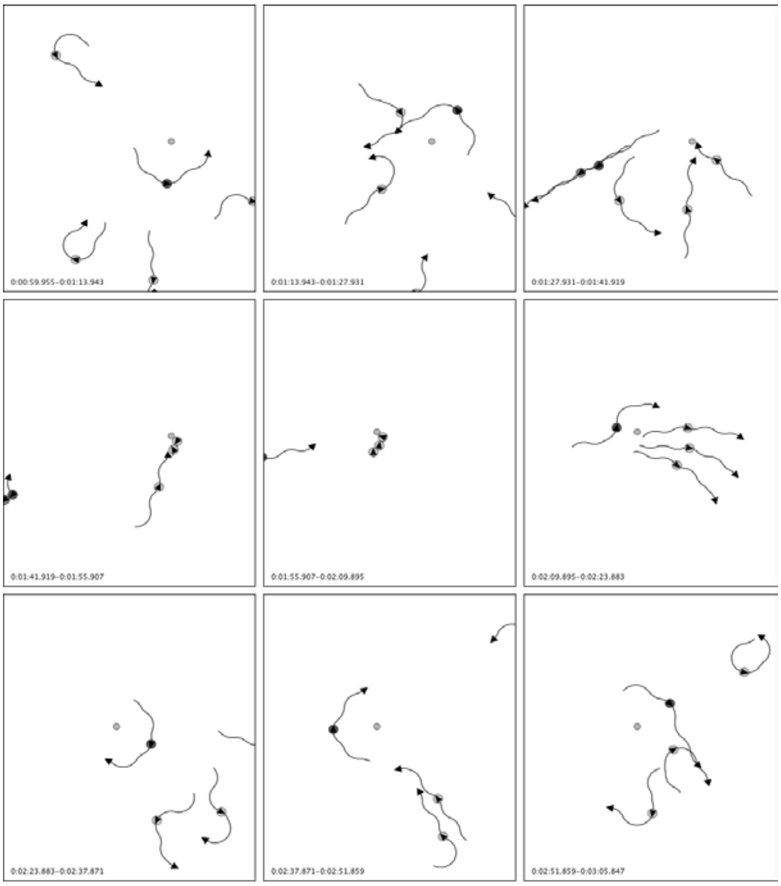

Figure 5*: Behavior of the Decoy Unie*

4.1. Results

The rain forest environment is the most complex of all tested environments because it has many obstacles and food sources. This had major consequences on the behavior of the robots. Because of the very large number of obstacles the Wumpus was not able to catch any of the Unies in any of the simulations except when it was placed near to the Unie at the start of the simulations (random deaths). The Unies were not very successful either, as they were constantly hindered by the obstacles when approaching the food sources.

In the desert environment, the number of Unies dying of hunger increases with simulation time because more time has passed for the Unies to completely deplete their energy (see table 2). Unies mainly die of hunger when they are too far away from a food source to smell it. The Wumpus always dies of hunger after having eaten all the Unies. Most Unies die after being chased away from the food. This may or may not give other Unies the opportunity to eat (which is another defining feature of decoy behavior).

Table 3 presents the absolute and relative occurrences of escape and sacrifice behavior (being the two variants of decoy behavior) in both the desert and rain forest environments. The number of times we saw escape behavior is zero. This is not completely surprising because the Wumpus is much faster than the Unies. Unless it is hindered by an obstacle, it will always catch the Unie. Although there would have been chances for a Unie to escape in the rain forest, a chase never occurred due to the richness of the food and the many obstacles preventing the Wumpus to detect a Unie near the food source it was guarding. In the desert environment we did see a number of chases, resulting sometimes in decoy behavior of the sacrifice type. Still, in many instances the Unies simply do not get enough time to eat the food. It is likely that the amount of escape and sacrifice behaviors would grow considerably once the speed of the Wumpus is decreased. At this point in time, however, this has not been simulated and tested yet.

	Time of death [s]		Type of death [%]		
	Average	Standard deviation	Random	Hunger	Chased
Unie					
First	34.0	33.8	40	0	60
Second	94.6	15.2	0	0	100
Third	201.0	73.1	0	20	80
Fourth	242.8	57.3	0	20	80
Fifth	350.5	81.2	0	40	60
Sixth	450.5	69.1	0	40	60
Wumpus	737.5	126.6	0	100	0

***Table 2**: Time and type of death in the desert environment*
Presented are the times of death (in seconds after the start of the simulation) and the type of death of each Unie and of the Wumpus. The labels first, second, etc. do not denote a specific Unie, just the sequence in which they died. One of the death types is labeled 'Chased', indicating that the Unie dies after being chased and caught by the Wumpus after it tried to approach the food.

	Occurrence			
	Desert environment		Rain forest environment	
	Absolute	Relative [%]	Absolute	Relative [%]
Escape	0	0	0	0
Sacrifice	6	25	0	0
Useless Death	18	75	0	0
Total	24	100	0	100

***Table 3**: Decoy behavior in the desert and rain forest environment*

5. Conclusion

This research project indicates that behavior that appears to be based on plan-related, central or global representations can actually be the result of the interaction of (cognitively) simple systems with each other and their environment.

The apparent cooperative behavior we observed in 6 simulations is not the result of some collective plan formation or role-division. The Unies do not use a representation of their world to guide their actions nor do they actively communicate with each other to determine the role each one has to play. They only react to the input from their senses. Yet, apparently cooperative behavior did occur. Interestingly, it seems that *scarcity* leads to cooperation. In the rain forest, Unies were not as much pressed to 'cooperate' in order to get to the food, as they were in the desert environment. In this sense, then, it is the nature of the environment that shapes the overall behavior of the simulated robots. This is in line with current ideas [3, 4] in cognitive science that emphasizes the importance of the embodied *embeddedness* of creatures for understanding their behavior.

References

[1] Brooks, R. (1991). Intelligence without representation. *Artificial Intelligence*, 47, 139-159.

[2] Brooks, R. (1999). Cambrian Intelligence. Cambridge, MA: MIT Press.

[3] Chiel, H.J. & Beer, R.D. (1997). The brain has a body: adaptive behavior emerges from interactions of nervous system, body and environment. *Trends In Neurociences, 20*(12), 553- 557.

[4] Clark, A. (1997). *Being There: Putting brain, body and world together again.* Cambridge, MA: MIT- Press.

[5] Russell, S.J. & Norvig, P. (1995). Agents that Reason Logically. *Artificial Intelligence: a Modern Approach.* (151-184). Englewood Cliffs, NY. Prentice Hall, Inc.

[6] Simon, H.A. (1969/1996). *The sciences of the artificial.* Cambridge, MA, USA. Massachusetts Institute of Technology.

[7] Willems, D.J.M. (2002) *Simulation in Behavioral Robotics: The user manual.* Nijmegen, The Netherlands. http://enterprise2.cogsci.kun.nl/robotics/UserManual/manual.html

[8] Willems, D.J.M. (2003) *Apparent Cooperative Behavior in Simulated Reactive Robots.* Nijmegen, The Netherlands. MA-Thesis, University of Nijmegen

Notes

[i] The names are mainly based on the Wumpus world, a simple environment used by Russell and Norvig [5] for demonstrating the virtues of logical reasoning in artificial agen ts.