

# The Evolution of Computer Bugs: An Interdisciplinary Team Work

Ole Caprani, Jakob Fredslund, Jørgen Møller Ilsøe, Jens Jacobsen, Line Kramhøft, Rasmus B. Lunding, and Mads Wahlberg

*Ole Caprani is associate professor in computer science. He has used computer controlled LEGO models for many years in teaching at all levels from school children to university students.*

*Jakob Fredslund has a PhD in computer science and is an excellent LEGO builder and the designer of several embodied agents, for example, a LEGO face that shows feelings.*

*Jørgen Møller Ilsøe has a Master's degree in computer science and is a master LEGO builder. As one of the few master students of computer science, he brought a huge LEGO model for his Master's exam.*

*Jens Jacobsen, an engineer, has designed and implemented an interface for interactive dance and other electronic devices for artistic use.*

*Line Kramhøft, a textile designer, has concentrated on the production of textiles, scenography, costumes and textile art using three-dimensional surfaces.*

*Rasmus B. Lunding, a musician and composer, has played as a soloist and in groups, has published two solo CDs as well as toured and presented compositions at an international level.*

*Mads Wahlberg is a light designer and electrical technician. He has designed and produced a number of technical gadgets for use in theatre.*

## 10.1 Introduction

An investigation of robots as a medium for artistic expression started in January 2000. As a result, insect-like LEGO robots, *Bugs*, have been created, that through movements and sounds, are able to express what to an observer seem like emotions, intentions and social behaviour. To enhance the perception of Bug behaviour as

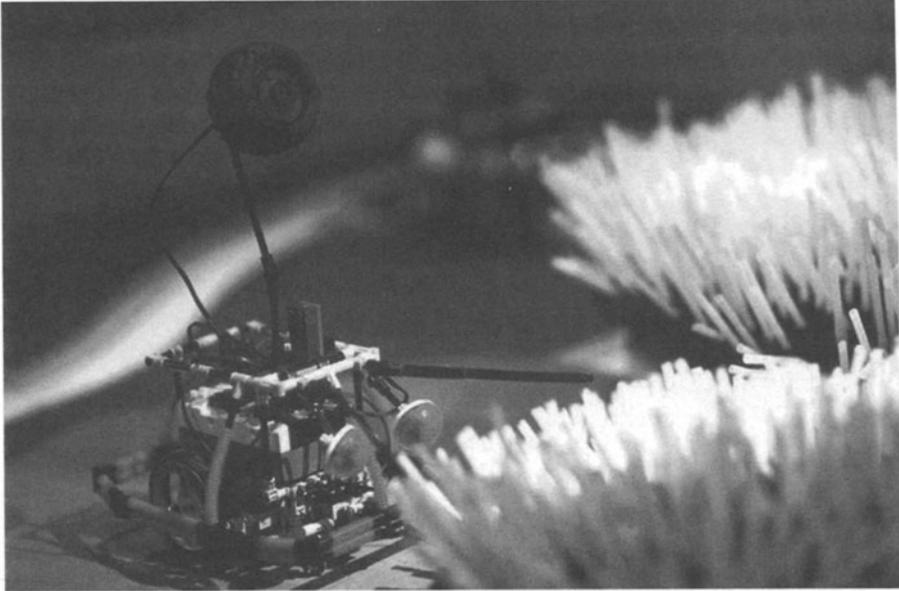


**Figure 10.1** People watch life in the Jungle Cube through peepholes in the black cloth.

aggressive, hungry, afraid, friendly or hostile, an artificial world has been created for the Bugs, an environment with a soundscape, dynamic lighting and a scenography of biologically inspired elements. This installation is called “The Jungle Cube”. Figure 10.1 shows the Jungle Cube from the outside. Through peepholes the audience can watch the life of the Bugs inside the cube.

Figure 10.2 shows a Bug inside the cube. In front of the Bug is a tuft of straws that the Bug can sense with its antennas and interact with, as if the Bug is eating from the straws. The life in the Jungle cycles through four time periods: morning, day, evening and night. There are two kinds of Bugs in the Jungle, *Weak Bugs* and *Strong Bugs*. During the day, the Weak Bugs are active, e.g., wandering around, searching for food, or eating, while the Strong Bugs are trying to sleep. During the night, the Strong Bugs are active, hunting the Weak Bugs while the Weak Bugs are trying to sleep. Strong Bugs search for a place to sleep in the morning, whereas the Weak Bugs search in the evening. When Bugs meet, they react to each other; e.g., when a Strong Bug meets a Weak Bug, the Weak Bug tries to escape and the Strong Bug attacks.

From a *technical point of view*, the installation can be described as a collection of computer-controlled elements as shown in Figure 10.3. The computer platform for an eight-channel cubic soundscape is a PowerPC computer; the rest of the elements are controlled by the LEGO MindStorms RCX computer, an Hitachi H8/3292 micro-controller embedded inside a specialised LEGO brick. Each Bug is controlled by a single RCX; the plastic straws can be moved whenever LEGO motors under the tufts are activated from an RCX-based Straw Controller. The lighting is controlled



**Figure 10.2** A Bug, an insect-like LEGO robot, in a biologically inspired environment inside the cube.

by a collection of RCX computers, each controlling three spotlights; and finally, an RCX-based Co-ordinator broadcasts infrared messages to the other RCX computers with information about the current time period: morning, day, evening or night. By means of these messages the Co-ordinator synchronises all RCX computers with respect to time period, and hence makes life happen in the Jungle Cube, cycling through the four time periods.

The Jungle Cube has evolved over a period of almost two years. Different versions have been presented to a general audience during that period. First at the NUMUS Festival for Contemporary Music in Aarhus, April 2000; then at the opening of the Center for Advanced Visualization and Interaction (CAVI) in Aarhus, March 2001; at the NUMUS Festival, April 2001; and most recently at the Nordic Interactive Conference, November 2001, Copenhagen (NIC 2001). One of the experiences from the presentations is that robots can indeed be perceived as autonomous creatures, and interaction among the robots creates an illusion of life-like behaviour in the artificial world of the installation. The main reason for this achievement is that people with different skills have been involved in the project: four people with *technical skills*, one engineer and three computer scientists; and three people with *artistic skills*, a composer, a scenographer and a light designer. This chapter describes how the multimedia installation, The Jungle Cube, has evolved as a result of the work of this *interdisciplinary* project group.

At the beginning of the project we were inspired by the robots of Grey Walter (1950, 1951), *Machina docilis* and *Machina speculatrix*. With very few sensors and actuators controlled by simple mappings from sensory input to actuator output,

these robots could produce behaviours “resembling in some ways the random variability, or ‘free will,’ of an animal’s responses to stimuli” (Walter, 1950). Furthermore, we learned that autonomous, mobile LEGO robots can be made to behave like the robots of Grey Walter or the simple vehicles of Braitenberg (1984),

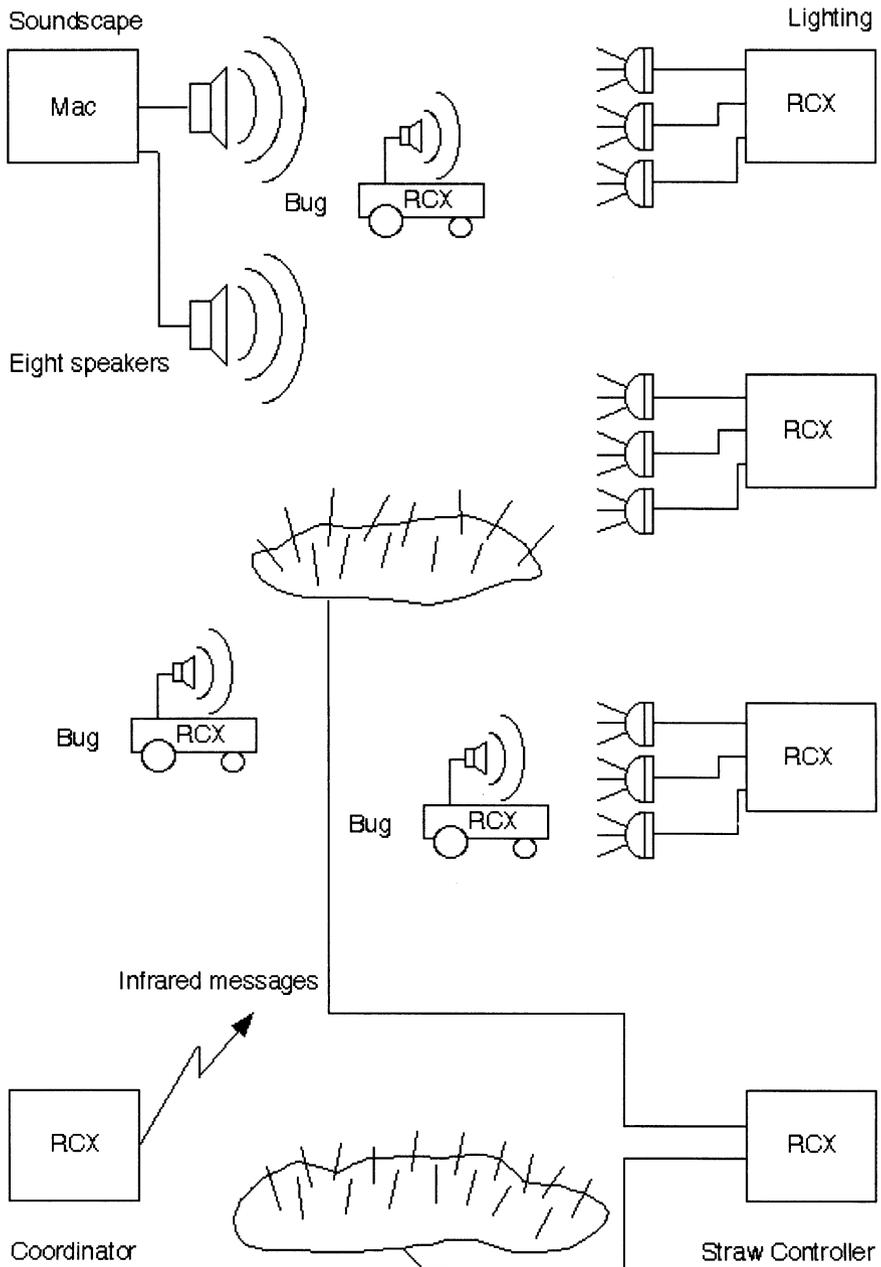


Figure 10.3 A technical view of the computer-controlled elements in the Jungle Cube.

and that a population of such LEGO robots interacting with each other and the environment can generate a great variety of behaviours, or scenarios, that are interesting for an audience to watch for quite some time. Based on this knowledge, the composer came up with an idea for a *robot scenario* with a population of animal-like robots that evolve through mutual exchange of artificial genes. The idea of using a crude model of genetic evolution to produce interesting scenarios was inspired by artificial life simulations of biological systems as described by Krink (1999).

From January 2000 until the first NUMUS Festival, the four technicians in the group and the composer were involved in the construction of the physical robots, and the programming of the robot behaviours. In this period we were mainly concerned with *technical issues*: construction of sensors, placement of sensors on the physical robot, gearing of motors, programming of a sound system, synchronisation of movements and sounds, and behaviour selection based on sensor input and the internal state of the robot. The initial robot scenario developed by the composer did, however, guide the discussions of technical solutions. From Autumn 2000 until the second NUMUS Festival, the environment of the robots evolved. This work involved the three artistic members of the group and a computer scientist. Here the technical issues still dominated: construction of a light system controlled by RCX computers, infrared communication among robots, and control of movements of tufts of plastic straws. In the last month before the presentation at NIC 2001, the *artistic issues* dominated: the lighting was integrated with the soundscape to express more clearly the beginning and end of different time periods: morning, day, evening and night; the length of the time periods were shortened to make the audience aware of the changes in Bug behaviours caused by time cycling through the four periods; rapid light changes were programmed to cause rapid changing patterns of shadows from the fixed visual elements to sweep the interior of the cube and as a result create a more dynamic visual atmosphere; the sounds of the robots were adjusted for better balance with the atmosphere of the soundscape, etc.

## 10.2 Initial Robot Scenario

From the very beginning the composer wanted to create a population of animal-like robots – creatures – and each creature should have character traits that varied dynamically between the traits of two extreme creatures: *Crawlers* and *Brutes*. A Crawler should be weak, frail and pitiful; a Brute should be strong, determined and ruthless. All creatures should have the same physical appearance; they should only differ in their behaviour, in the way they moved and in the sounds they made. The character of each creature should be determined by artificial genes, a set of parameters that controlled how a creature selected different behaviours from among their repertoire (rest, sleep, attach, flee or eat) based on stimuli from the environment (another creature or food nearby) and the internal state of the creature (Crawler or Brute, hungry, aggressive, scared or content).

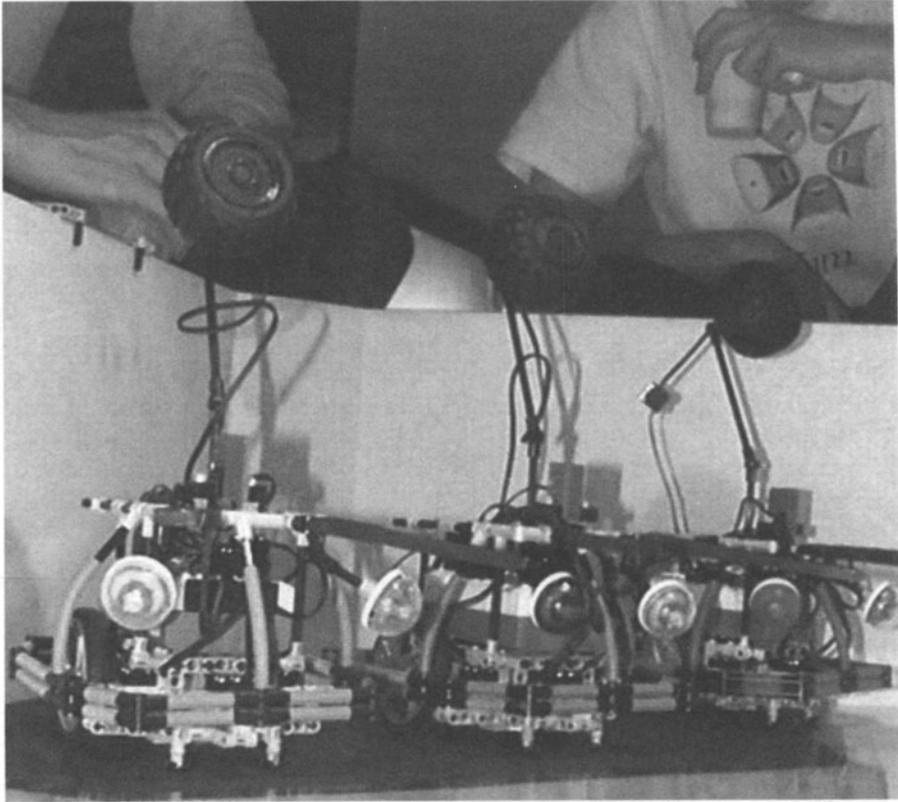
The creatures should exist in an environment where they should move around looking for a place to sleep, for the other creatures, or for food. When the creatures met they should interact, and a transfer of genes between the interacting creatures might take place. As a result the artificial genes of the interacting creatures would change, for example, through a crossover of the genes of the interacting creatures. Hence, the character traits of the creatures should change through interaction, and different scenarios would evolve depending on the character traits of the initial population and the patterns of interaction: All creatures might end up as either Brutes or as Crawlers or maybe some steady state might be reached with a few Brutes, a few Crawlers and a varying number of creatures with both weak and strong character traits.

The idea of artificial gene exchange was not realised in the Weak and Strong Bug scenario, but the idea of robots as creatures with different character traits was, as we shall see, realised in the Bug scenario.

### 10.3 From a Mobile Robot to a Bug

The four technicians on the team had previous experience with construction of physical LEGO robots controlled by programs running on the RCX. It was decided to construct the creatures of the initial robot scenario out of LEGO with the RCX as the controller platform: through sensors (touch or light sensors) connected to the input ports, the creatures could perceive the environment, and through actuators (motors or lamps) connected to the output ports, the creatures could act in the environment, while the infrared transmitter/receiver could be used to exchange artificial genes between the creatures.

After the choice of LEGO and RCX the immediate challenge was to build an animal-like creature out of LEGO bricks and develop programs to make them move and sound like animals. The result of these initial efforts looked very much like the robots of Figure 10.4. The advantage of the choice of LEGO was that the physical robots could be easily rebuilt to change the appearance of the creatures, to change the mechanical mode of operation, and to make them mechanically robust. Several small changes have since been made to the robots in Figure 10.4. The two-wheel base has been changed to a four-wheel base with two passive front trolley wheels, making it possible for the robots to move more suddenly and abruptly; and the stalk holding the speaker in the rubber tyre protruding from the top of the Bug was shortened to change the vibrations of the speaker resulting from the moves of the robot. In order to change the appearance of the robots dramatically when the ultraviolet light was turned on during the night in the Jungle Cube, the colour of the Bug eyes was altered and special red tape was put onto the coloured carapace around the robots. Additionally, the movement of the two eyes resulting from the robot movements was constrained so that the eyes moved to the left or right almost simultaneously.



**Figure 10.4** The Bugs, NUMUS 2000.

However, the choice of the RCX had its drawbacks:

- the RCX has only three independent input ports, and only primitive touch, light, temperature and rotation sensors are provided by LEGO;
- the RCX has only three independent output ports and only motors and lamps are provided as actuators;
- the sound from the built-in speaker is very weak, and bandwidth is limited;
- there is limited memory space, less than 32 k bytes;
- the programming tools that LEGO offers have limited features for data structuring and only allow slow and limited access to the 8 bit micro-controller embedded inside the RCX, the Hitachi H8/3292.

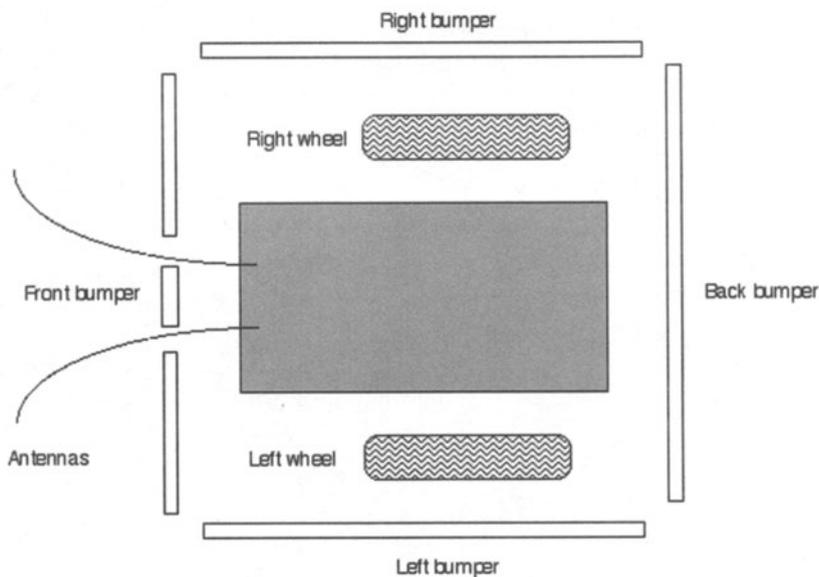
Because of this we decided:

- to develop a few customised sensors;
- to implement a sound system based on a customised speaker; and
- to use the general-purpose language C as the programming language.

### 10.3.1 Sensors and Locomotion Actuators

Figure 10.5 shows the resulting customised sensors: two antennas, based on bend sensors, and four independent bumpers, mounted on LEGO touch sensors. A push on a bumper is registered by a touch sensor. Since there are only three input ports, we had to connect several sensors to one input port. For an observer it is not easy to distinguish between a bend on the right or left antenna, so the two antennas were connected to the same input port. On the other hand, to an observer it is obvious on which side the robot encounters an obstacle, so the robot should be able to distinguish activation of each of the four bumpers. To achieve this, the resistors inside the LEGO touch sensors were changed to have distinct values. This made it possible to register activation independently on the different bumpers, even though the four touch sensors were connected to the same input port. Two LEGO light sensors were also used: one underneath the Bug to sense the colour of the floor, the other on top of the Bug to sense the ambient light level in the room. A light sensor and the four touch sensors can be used independently even if they are connected to the same input port. In total, we managed to have seven independent sensor inputs on three input ports. Through the antennas and the bumpers the Bug can sense obstacles; through the light sensors the Bug can sense coloured areas on the floor and day/night light in the room.

The movements of the Bug is accomplished by two wheels driven by two independently controllable LEGO motors. The rotation speed of the motors is also controllable. Hence, the Bug can be controlled to perform a variety of manoeuvres, for example, a very slow sneak forward or a fast wiggle.



**Figure 10.5** Sensors and actuators connected to the RCX of the Bug.

### 10.3.2 Sound Actuator: Square Wave Synthesiser

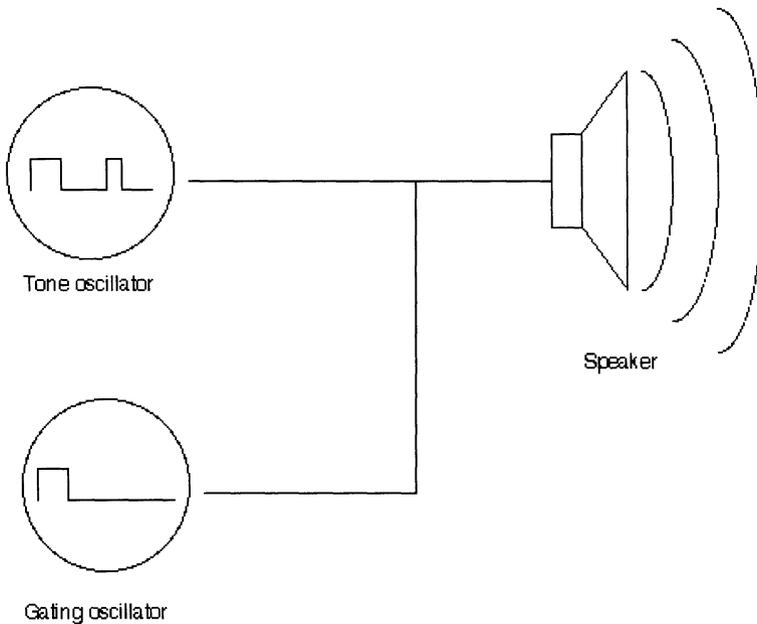
An electromagnetic speaker connected to an output port of the RCX can be turned on and off under program control. As a result, the diaphragm of the speaker will move back and forth between its two extreme positions. This is the basis for the sound generation system of the Bug. As an example, a tone with frequency 100 Hz will be generated if the speaker is repeatedly turned on for 5 msec and off for 5 msec, since  $1000/(5 + 5)$  Hz = 100 Hz. Using the accurate timers of the RCX to control the on and off periods of the speaker, a software synthesiser has been programmed, see Figure 10.6. The synthesiser has two oscillators, a *tone oscillator* and a *gating oscillator*. The basic waveform of the tone oscillator is a double square wave consisting of two consecutive single square waves. The sum of the four time periods that define the on and off periods of the double square wave determines the frequency of the tone. The timbre of the tone is determined by the relationship between the four on/off periods. The double square wave was chosen as the basic waveform because the timbral possibilities increase significantly with four time periods instead of the two time periods of a single square wave: keeping the sum of the four periods constant while adjusting the relationship between the four periods creates a tone with a surprising variation in timbre. This was already discovered and used in the late 1950s as described in Manning (1985, pp. 72–74).

The basic waveform of the gating oscillator is a single square wave. The on/off periods of the gating oscillator is used to modulate the tone by turning the tone oscillator on and off. The usage of a gating oscillator to modulate a tone oscillator also goes back to the late 1950s. The result of the gating depends on the relationship between the frequencies of the two oscillators: When the frequency of the gating oscillator is in the sub-audio range (e.g., 5 Hz), the periodicity of the gating results in the generation of a rhythmic pulse-like pattern of the tones from the tone oscillator; when the gating frequency approaches the tone frequency, the two wave forms merge into a triple- or multiple square wave form, and the result is drastically and rapidly changing tones with a rich spectrum of harmonics; and when the gating frequency gets higher, the result is a tone determined by the gating frequency. The gating modulation used in the synthesiser is an example of a general signal modulation method, called amplitude modulation, Roads (1996).

By means of short algorithms that change the six parameters of the synthesiser (the four periods of the tone oscillator and the two periods of the gating oscillator), the composer managed to program algorithms that generated a variety of insect-like musical sounds. The advantage of this sound system is that the synthesiser and the sound generating algorithms do not take up much space in the memory of the RCX.

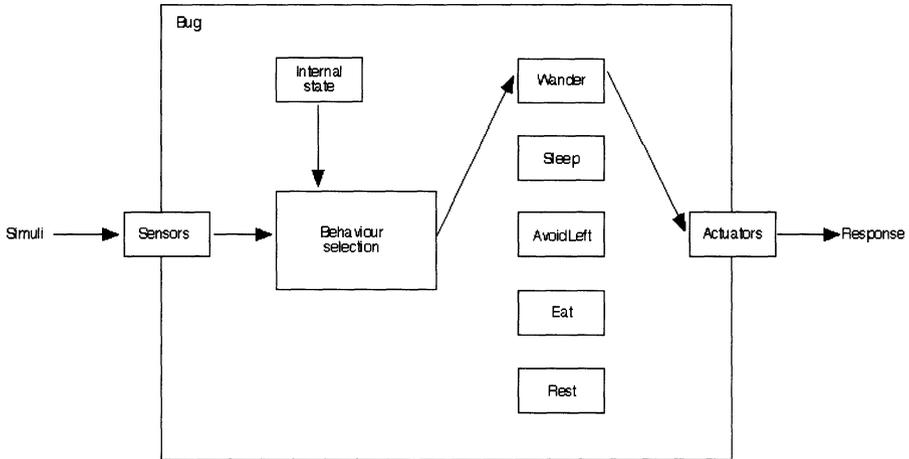
### 10.3.3 Behaviour-based Control Architecture

Now that the sensors and actuators have been described, we turn to the development of the *Bug control program*. Through a number of experiments we found a way of



**Figure 10.6** The square wave-based software synthesiser.

controlling the two motors so that the resulting movements of the Bug make it appear like an insect wandering around: first the Bug goes forward for a random period of time, between 100 msec and 500 msec; then it stops abruptly by braking the two motors; afterwards it waits for a random period of time; then by a small random angle it turns in-place left or right; and finally, it repeats this sequence of actions by going forward again. Randomly chosen speeds are used each time the motors are activated. The perception of an insect wandering around is not only caused by the Bug movements but also the accompanying insect-like sounds and the physical appearance of the Bug while moving, e.g., the way motor activation and motor braking cause the eyes, the antennas and the speaker stalk to move. Our next series of experiments were concerned with Bug reactions to obstacles. Through the bumpers and antennas, the control program gets information about obstacles encountered. There is, however, not much information about what kind of obstacle the Bug has encountered, whether a passive obstacle like a wall or another Bug. At first, we did not care much about this but tried to program two kinds of reactions: avoid the obstacle, or attack the obstacle. Furthermore, we wanted the reaction to an obstacle to be instantaneous: stop the current movements and sounds immediately and start to avoid or attack the encountered obstacle, while signalling this by generating sounds. In the beginning, we could only make the Bug look like a car bumping into an obstacle in front and then backing up with an alarm-like tone. As a result, the illusion of an insect was broken and the robot was seen as a car. Later, the composer made melodic insect-like sound sequences to accompany the movement, so instead of a car, the robot was perceived as an insect being surprised, afraid, curious, or aggressive when encountering an obstacle.



**Figure 10.7** The behaviour-based architecture of a Bug control program.

These initial experiments led us to use a *behaviour-based architecture* for the control program of the Bug (Brooks, 1986, 1991; Mataric, 1992, 1997). In Figure 10.7 the behaviour-based architecture is shown. Each Bug has a repertoire of different behaviours: wander, sleep, avoid-left, etc. Each *behaviour* consists of two independent program modules, a *locomotion module* and a *sound module*. When executed, each module generates a sequence of simple instantaneous actuator actions. Locomotion modules start or stop the motors, sound modules start or stop one or both oscillators that control the speaker.

At any given instance, only *one* behaviour is active. The two modules of the active behaviour are executed concurrently: the locomotion module moves the Bug around, while the sound module generates the accompanying sounds. Triggered by external stimuli like a bumper being pushed, or triggered by changes in the internal state, e.g., an avoiding behaviour that finishes, the active behaviour is terminated and a new behaviour is selected to be the active one. Hence, the *behaviour selection* depends on external stimuli and the internal state as shown in Figure 10.7. The selection mechanism itself is based on the concept of *motivations* (Krink, 1999). At any given instance, each behaviour is associated with a *motivation value*. The behaviour with the highest motivation value is selected to be the active behaviour. Motivation values are calculated every 50 msec by means of *motivation functions*, one for each behaviour. The motivation functions use sensor values resulting from external stimuli, and the internal state to calculate a motivation value: during the day the motivation value for the behaviour “wander” is high, during the night it is low. The opposite applies to the motivation value for the behaviour “sleep”; the motivation value for “avoid-left” is higher than the value for wander or sleep when the left bumper is activated and the value stays high until the avoid-left behaviour ends. One of the consequences of this behaviour selection mechanism is that the reaction to obstacles is delayed at most 50 msec and is perceived as immediate reaction.

### 10.3.4 Behaviours

In the above description of the behaviour-based architecture, the word behaviour has been used to denote two different things. On the one hand, it denotes an *internal mechanism* of the control program, that is, the locomotion and sound modules that generate time-extended sequences of actions to the motors or the speaker. On the other hand, it denotes an *external interpretation* of the Bug enacting these actions. During the development of the Bugs, these two aspects of the behaviour concept turned out to play a central role. To make the creatures of the initial scenario more concrete we simply tried to figure out what the creatures should do in the different situations they might be put in. As a result, the composer came up with a list of named behaviours with a description of the conditions for the different behaviours to be enacted (sleep at night, wander around searching for food during the day, go forward when bumped into from behind, etc.). Then the technicians programmed these behaviours one by one as program modules and added them incrementally to the control program. Hence, the description of the initial robot scenario was *decomposed* into a list of desired behaviours and these behaviours were then *built up* from the primitive actuator actions.

Programming the locomotion and sound modules was accomplished by developing *levels of abstract commands* to bridge the gap between the primitive low-level actuator actions and the high-level descriptions of the desired behaviours. It turned out that only one level of abstract commands was necessary to express all the desired locomotions as simple program modules. These commands were: GoForward, TurnLeft, TurnRight, and Brake. Each has an obvious implementation in terms of the primitive motor actions. The names of these commands correspond to the observed, resulting movement of the Bug when such a command is executed by a locomotion module. The introduction of these commands was not only a means to simplify the programming but also a means to develop a common language among the technicians and the composer in discussing locomotion.

The development of levels of abstract commands to program the sound modules was much harder and time consuming than the development of the one level of abstract locomotion commands. In fact, only one of the sound modules programmed for the NUMUS 2000 presentation was re-used in later versions of the Bug. There were several reasons for the slow development of the sound modules. The technicians had had no experience with programming sound generation at this low level; the composer had had no experience with square wave synthesis. Furthermore, the composer experimented a lot with individual animal-like melodic sound effects for each Bug, and sound modules that made Bugs perform small melodic pieces of music in concerto. For example, a small piece of music for three Bugs was composed to accompany the three Bugs on their way home. It was, however, difficult to make sure that all three Bugs joined in at appropriate times, because the Bugs might not be on their way home simultaneously. Finally, during autumn 2000, one of the computer scientists and the composer finally managed to find useful abstractions for sound module programming. It turned out that two levels of abstractions were enough to program all current sound modules. The lowest level is the square wave synthesiser of Figure 10.6. The synthesiser is controlled by a set of commands that start and stop

the two oscillators: StartPlaying, StopPlaying, StartGating, StopGating. The commands at the next level are all expressed in terms of musical concepts: PlayNote, PlayNoteGlissade, PlayTimbreGlissade. The first has an obvious meaning, the next plays an up or down glissade of intermediate notes between two notes, and the last command plays a glissade from one square-wave form of a note to another wave form of the same note. All commands have a simple implementation in terms of the lower level commands. At the start of this sound module development, the computer scientist wrote a number of sound modules in terms of the abstract commands; the composer listened and suggested changes. Through these experiments, we found that sounds generated by using the gating commands together with glissade commands yielded a rich variety of sound material, which were immediately useful as insect sounds. Later, the composer managed to program all the sound modules in terms of these two levels of abstract commands.

Synchronisation of movements and sounds is accomplished in two ways. The first way is to start the locomotion and sound module of the active behaviour simultaneously and to execute the two modules concurrently. The result is that, e.g., the sound accompanying the Avoid-Left movements starts immediately when a Bug bumps into an obstacle and the Avoid-Left sounds are heard while the Bug is turning away from the obstacle. The second way is to turn the speaker on and off from within a locomotion module. This can be used, e.g., to turn on the sounds accompanying the Sleep movements only during the short periods while the Bug is wiggling in its sleep.

## 10.4 Evolution of the Bug Scenario: From Crawlers and Brutes to Weak and Strong Bugs

A month before NUMUS 2000 we were still far from a realisation of the initial scenario of Crawlers and Brutes. The behaviour-based architecture had been programmed and tested with the locomotion modules for wander, hunt, avoid and attack behaviours. However, we were still struggling with the corresponding sound modules. The character traits expressed by enacting the implemented behaviours resembled the desired Crawler and Brute traits. Unfortunately, though, we had not yet implemented a parameter mechanism to change the behaviour selection or to control the behaviours dynamically. Hence, there was no way of changing the character traits of the creatures by mutual exchange of parameters. At the peak of this crisis, the composer came up with a revised scenario in which only two kinds of creatures exist: Weak Bugs and Strong Bugs. These creatures were to have *fixed* character traits. The Weak Bug should be like a Crawler; a Strong Bug like a Brute. In the initial scenario, the overall dynamics originate in changing character traits of the creatures between Crawlers and Brutes. In the revised scenario, the overall dynamics result from both the explicit passing of time through day and night and the spatial placement of home locations and food areas for the creatures. The Weak Bugs are active during the day; the Strong Bugs are active at night. Both the Weak and the Strong Bugs have a home where they sleep at night or day. The homes of the Weak and the Strong Bugs are situated in different locations in the environment. Close to the home of the Strong Bugs there is a food area which the Weak Bugs search for during the day. When they find the food, they eat a meal.

The composer described the revised scenario as a list of Weak and Strong Bug behaviours similar to the original Crawler and Brute behaviours. Moreover, most of the locomotion modules already implemented could be used in the revised scenario. Two new behaviours, GoHome and Eat, were the only behaviours we had to implement from scratch. As described in the introduction, the idea of Weak and Strong Bugs with fixed character traits are still used in the present version and so are many of the locomotion modules implemented for NUMUS 2000.

## 10.5 Evolution of the Bug Environment: From a Box to a Cube

In the description of the initial scenario of Crawlers and Brutes, we only had vague ideas about the environment of the creatures and these ideas had not been developed any further before the introduction of the revised scenario. At that point, we began to think about how to realise a *physical environment* for the Weak and Strong Bugs. The physical environment should enable the Bugs to move around and it should include elements that the Bugs could perceive through their sensors, such as day/night, food and home. An environment along these lines was constructed for NUMUS 2000 (see Figure 10.8).

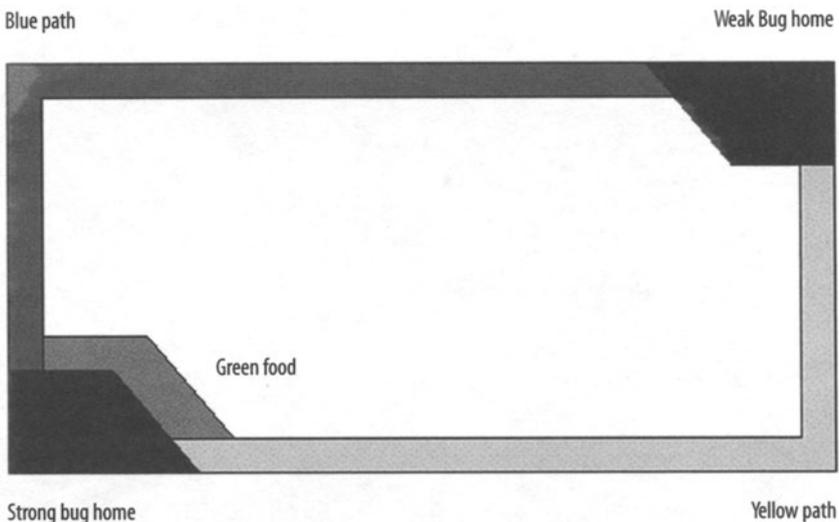


**Figure 10.8** People watch the Bug life in a box environment, NUMUS 2000.

The environment was an open wooden box measuring  $1.7 \times 1.0 \times 0.4$  m. The floor had a flat, smooth surface, so that the locomotion actuators could move the Bugs around. Two spotlights were placed above the box, a yellow and a blue one. Yellow on, blue off signified day; yellow off, blue on signified night. The duration of day and night periods was controlled by an RCX that periodically turned the two spotlights on and off. The light sensor on top of the Bug, the ambient light sensor, was used to sense the day and night light. The food area, the Weak and Strong Bug homes, and paths to the Bug homes were laid out on the floor as coloured areas (see Figure 10.9). The colours were chosen so that the light sensor underneath the Bug, the floor light sensor, could be used to distinguish the different locations. The two different coloured paths along the walls were used by the GoHome locomotion module to navigate the Bugs to their home area: when a Bug comes close to a wall it uses the path colour to decide which way to turn to head for the right home.

At the NUMUS 2000 Festival there were three Weak Bugs and three Strong Bugs. The two spotlights were controlled to yield a 10-minute overall time cycle with 5-minute day and night period.

Table 10.1 shows an overall model of the Bug responses to the stimuli from the elements of the NUMUS 2000 environment. The table contains a list of what we considered to be the main stimuli that the Bugs should react to. Also listed are the sensors the Bugs use to sense these stimuli; how the Bug control program interprets the received sensor values, i.e., how the stimuli are perceived by the Bugs; and the behaviours that are either activated or influenced by the stimuli. This list of behaviours is, on the one hand, a list of the control program modules that generate the low-level responses to the stimuli in terms of sequences of movements and sounds. On the other hand, the names in the list describe the intended external



**Figure 10.9** The coloured areas of the floor in the box environment, NUMUS 2000.

Table 10.1 Bug responses to stimuli from the box environment.

| Stimuli        | Sensed by             | Perceived as | Perceived as activate or influence behaviour in Weak/Strong Bug |
|----------------|-----------------------|--------------|---|
| Bump into wall | Touch sensor/antennas | Obstacle     | Avoid/Attack  |
| Bump into Bug  | Touch sensor/antennas | Obstacle     | Avoid/Attack  |
| Yellow light   | Ambient light sensor  | Day          | Wander/GoHome or Sleep  |
| Blue light     | Ambient light sensor  | Night        | GoHome or Sleep/Hunt  |
| Green area     | Floor light sensor    | Food         | Eat/–   |
| Blue area      | Floor light sensor    | Path         | GoHome/GoHome   |
| Yellow area    | Floor light sensor    | Path         | GoHome/GoHome   |
| Black area     | Floor light sensor    | Home         | Sleep/Sleep   |

interpretation of the low-level responses to the stimuli. Hence, the stimuli/response list is not only a model of the mapping from sensors to actuators performed by the control program, but also a model of the intended perception by an audience in terms of causality: when a Weak Bug bumps into a Strong Bug, the Weak Bug tries to avoid the attacking Strong Bug; when a Weak Bug detects food during the day, it eats; when night turns into day, the Strong Bugs search for their home and when home, they sleep.

What actually happened in the box at NUMUS 2000 was in many ways different from the intentions modelled in Table 10.1. When the spotlights were turned on and the Bugs started, the audience saw six animal-like LEGO robots moving around with no purpose, swiftly and abruptly, while avoiding and attacking each other or the walls. The audience heard a steady, high intensity animal-like sound mixed with melodic fragments. Although it was obvious that the sound came from the robots, it was impossible to locate the origin of the different sounds. The enacted scenario was perceived as disturbed creatures which were either in panic, afraid or simply confused. This emerged from frequent Bug-wall interactions and Bug-Bug interactions as a result of the wander, hunt, avoid and attack behaviours. Because of several flaws in the day/night light detection, the floor colour detection, and the behaviour selection, these four behaviours were active most of the time. The sleep, go home and eat behaviours were only activated occasionally and only for short periods of time – actually, for so short periods that activation of these behaviours was not perceived as a consequence of the day/night light or the food colour on the floor.

Even though the Bug behaviours were not enacted as intended, and hence were not perceived as part of the scenario described by the composer, we were encouraged to continue our efforts, partly because of the positive comments from the audience, and partly because several people did watch the Bugs for quite some time. During the following few months we fixed the flaws in the control program so the day/night stimuli had an obvious impact on Weak and Strong Bug activities. Furthermore, the Bugs could indeed find their way home, they slept at home, and the Weak Bugs ate on the green area. However, because of the limited physical space, the perception of the individual Bugs was still disturbed by the frequent activation of avoid and attack behaviour.

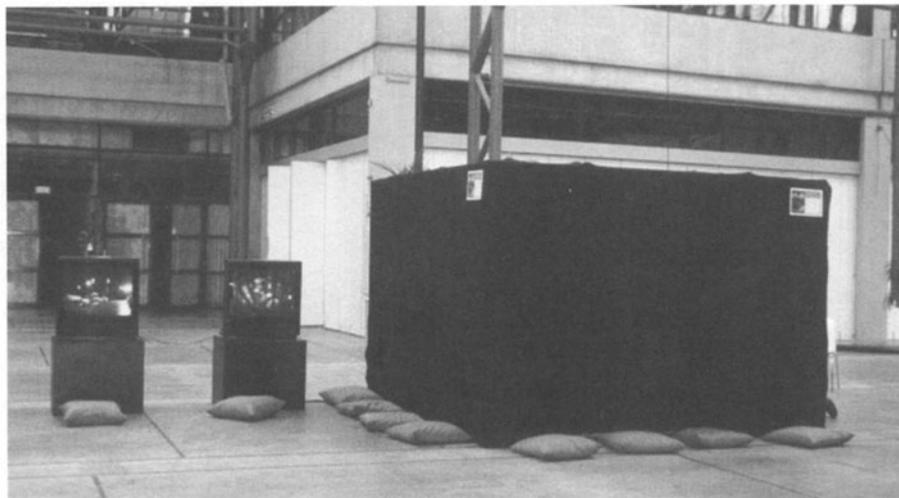
In the beginning of the summer, 2001, two of the computer scientists and the engineer left the project; the composer and a computer scientist continued to discuss how to improve the environment of the Bugs. During September 2001, the composer came up with an idea for a new Bug environment: *a closed cube*. The idea was, first of all, to make enough room on the cube floor for the Bugs to move around without frequent collisions. Second, peepholes in the cube should enable an audience to watch the Bugs in a jungle-like environment created in the interior of the closed cube. The composer and the computer scientists designed a physical cube together with a local theatrical design and production company. In November, the company finished the physical cube.

### 10.5.1 The Cube

The cube measures  $3 \times 3 \times 2$  m. It is constructed as an iron skeleton covered with black cloth containing peepholes for the audience. At CAVI 2001 and NUMUS 2001 we used a cloth with holes positioned, so the audience had to stand up to put their heads through the holes (Figure 10.10). In the present version, NIC 2001 (Figure 10.1 and Figure 10.11), the holes are positioned so the audience can sit more comfortably on cushions around the cube. Furthermore, the audience gets a better view of the Bugs from the head positions closer to the cube floor. Also added to the NIC 2001 version are two TV monitors outside the cube. Video on the monitors show the Bug life inside the cube. Two static cameras inside transmit the video to the monitors.



**Figure 10.10** The Jungle Cube, NUMUS 2001.



**Figure 10.11** The NIC 2001 Cube with cushions and two TV-monitors.

In the first version of the black cloth, two holes for the arms were positioned below each peephole. The idea was to enable the audience to influence the activities of the Bugs and other elements inside the cube. Through the armholes the audience should be able to reach for an electric torch hanging down inside the cube, switch it on and point it at, e.g., a Bug to make it react to the light.

It turned out that we did not make any use of the armholes. The reason was that we could only think of panic as the Bug reaction to torch light, and if most Bugs were panicking because of torch lights pointing at them, it would prevent the audience not using an electric torch from experiencing the rest of the Bug behaviours. The armholes caused some confusion at the CAVI and the NUMUS presentations: people thought that waving arms through the hole had an influence on the Bug behaviours and children thought it was meant to enable them to touch the Bugs.

### 10.5.2 The Scenography

At first, only the composer and the computer scientist discussed the appearance of a jungle-like environment inside the cube. We imagined a mix of natural and artificial elements like stones, plants and RCX controlled, non-mobile, animal-like LEGO elements with movable speakers. We also began to work on non-mobile elements such as a speaker built into an angular, plastic tube.

After these initial discussions and experiments, the composer suggested involving a scenographer. At the first meeting with the scenographer, we used a demonstration of the Bugs in the box and a video from NUMUS 2000 to initiate discussions about the scenographer's involvement in the project. Based on this and the revised scenario, the scenographer came up with a model of the interior of the cube (see



**Figure 10.12** The scenographer's model of the cube interior.

Figure 10.12). The idea was that the cube should only contain artificial elements with an organic appearance to enhance the insect-like appearance of the Bugs. Furthermore, there should only be elements that the Bugs could sense and that seem natural to an audience as something the Bugs can react to.

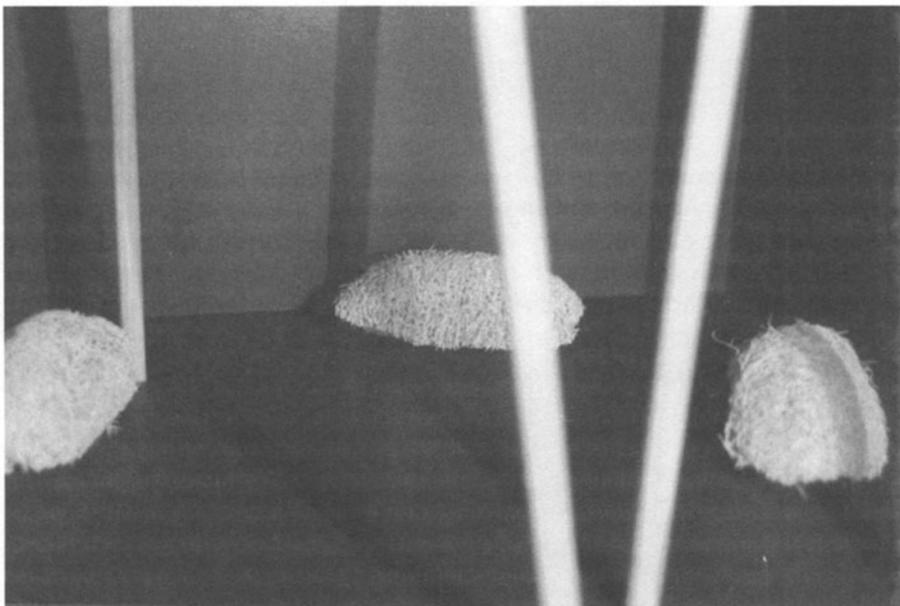
First of all, the model gave us a shared understanding of the scenographer's conception of the visual appearance of a jungle-like environment inside the cube. Secondly, the involvement of the scenographer directed our focus from the single Bug behaviours onto the design of an environment in which these behaviours would make sense to an audience. At the beginning of the discussions, the scenographer only had a vague understanding of the Bug sensors and the possible Bug perception of an environment through the sensors. Hence, the scenographer simply accepted that

there should still be, as in the box environment, coloured areas on the cube floor, and these areas should still be perceived by the Bugs as home and food locations. Also accepted was the idea of navigating the Bugs to home and food locations by means of coloured pathways on the cube floor.

### 10.5.3 The Lighting

As can be seen from Figure 10.13, the visual appearance of the elements depends on the light setting. The scenographer did not have the technical skills required to set up and control the light inside the cube, so a light designer with experience in stage-lighting control was involved in the project. The primitive on/off RCX control of the yellow and blue spotlights of the box environment inspired the light designer and the computer scientist to experiment with a similar autonomous RCX-based light control mechanism, but with numerous theatrical spotlights and the conventional stage-light fading mechanism instead of the crude on/off control of the two box lights.

The control of the light intensity of a spotlight by a program on an RCX turned out to be quite a technical challenge. The spotlights need a 12 V power supply, the RCX output ports can supply 9 V. This means that the power to the spotlights had to come from a 12 V power source controlled by signals from the RCX output ports; a signal of carefully timed on and off periods on the output port was sent to an electrical circuit that used the signal and a 12 V power supply to turn on a spotlight with intensities ranging from almost dark to full light; and the intensity is controlled by the relationship between the duration of the on and off periods of the



**Figure 10.13** Model of cube interior with light setting.

RCX signal. The light designer built the electrical circuits, while the computer scientist programmed the fading algorithms and the overall control of the cube light from the RCX-based Co-ordinator and Light Controllers (see Figure 10.3).

The overall control of the light setting is expressed in light control concepts known to the light designer: For each time period there is a table of light cues and time intervals between cues; for each cue, the table contains up and down fading times common to all spotlights and the target light percentage for each spotlight. The entries are used by each Light Controller to fade up or down a single spotlight depending on its current light percentage and the target percentage. After implementation of the light control system, the light designer simply filled in and edited the four tables with light cues, fade up and down times, and light percentages to test different jungle light settings. Besides the spotlights, the light designer also installed on/off controllable ultraviolet neon tubes to be used during the night period.

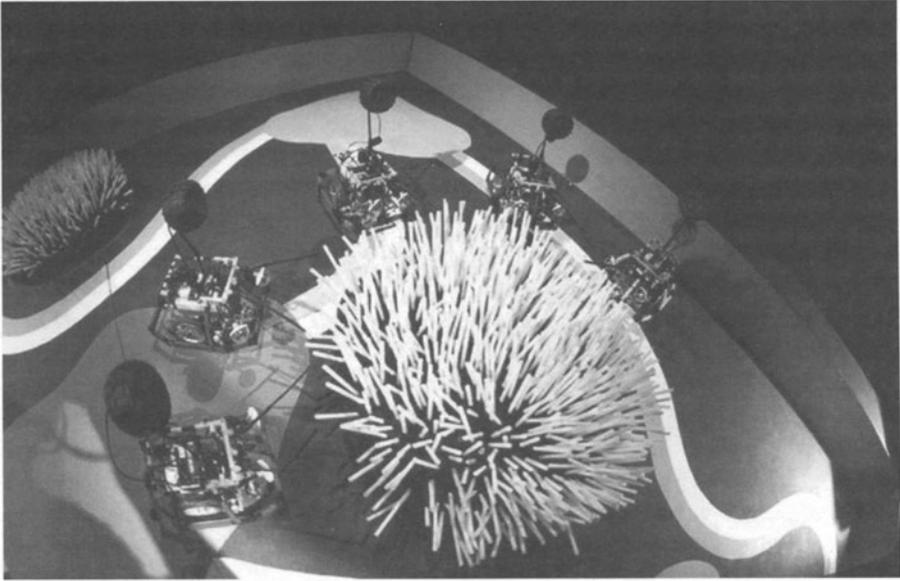
### **10.5.4 The Soundscape**

The composer's jungle-cube idea included a soundscape as part of the jungle-like environment inside the cube. The acoustics of the closed cube covered with cloth made it possible for a person with both ears inside the cube to spatially position sounds from differently located speakers inside the cube. A speaker was placed in each of the eight corners in the cube. The composer used the eight speakers to create an eight-channel soundscape in which sounds can be positioned in each of the eight corners or sounds can be moved around inside the cube. To create the musical material for the soundscape, the composer used Max/MSP to manipulate sounds and to spatially position the sounds, and the composer used ProTools to assemble the musical material into a soundscape, and to control the playback of the soundscape. These platforms run on the PowerPC and they are well known to the composer.

To integrate the soundscape with the insect-like Bug sounds and the organic visual appearance inside the cube, the composer chose sampled insect sounds as the basic sound material for composing the soundscape. The resulting soundscape has some resemblance with a natural jungle soundscape, although none of the individual sounds are immediately recognisable as natural sounds.

### **10.5.5 Co-evolution of Bug Scenario and Cube Environment**

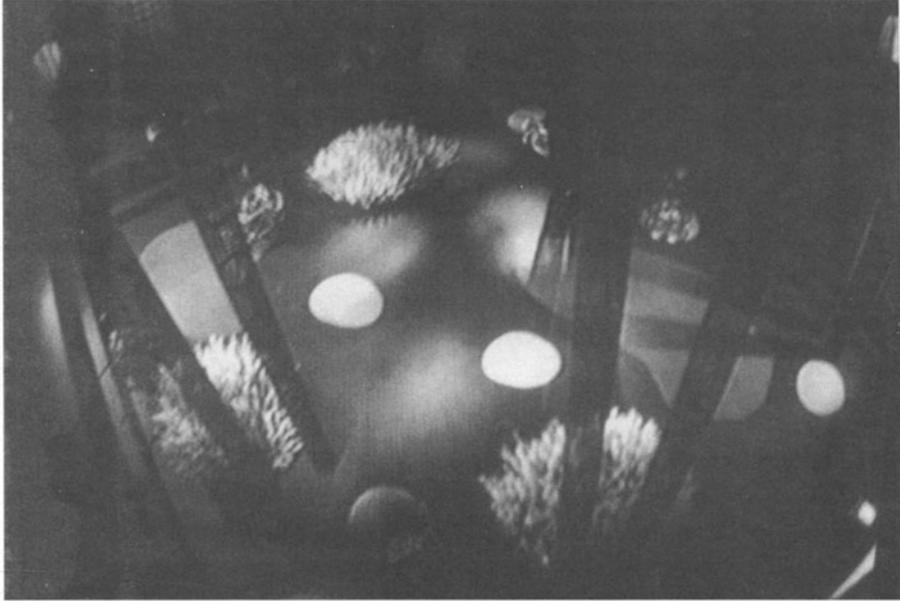
The Bug scenario and the cube environment evolved through the three presentations at CAVI, NUMUS and NIC, e.g., the visual appearance changed as illustrated in Figures 10.14–10.16. The three presentations represent stages in the process of integrating the scenography, the lighting, and the soundscape to create a jungle-like visual and sonic atmosphere inside the cube. The three presentations also represent stages in our attempt to integrate Bug behaviours and cube environment so the Bug behaviours enacted in the cube environment make sense to an audience.



**Figure 10.14** The floor of the Jungle Cube, CAVI 2001.



**Figure 10.15** The floor of the Jungle Cube, NUMUS 2001.



**Figure 10.16** The floor of the Jungle Cube, NIC 2001.

After the internal cube had been modelled (Figure 10.12), and after the overall structure of the technical platform had been designed (Figure 10.3), there were numerous discussions about the Bug scenario in the new environment, mainly between the composer and the computer scientist. As a result, the composer refined the Bug scenario. The main change was the addition of the two time periods, morning and evening. This explicit time separation of the day, with active Weak Bugs, and the night, with active Strong Bugs, was intended to enhance the difference between the two kinds of Bugs. The composer also suggested having four Weak Bugs and two Strong Bugs. This emphasises the two Strong Bugs as lonely hunters, and the flock of Weak Bugs as the hunted. Furthermore, the introduction of four instead of two time periods was also intended to make it possible to exploit the dynamics of the lighting and the soundscape to express the passing of time more precisely than the crude yellow/blue light on/off periods used in the box environment. Each of the four time periods could then be expressed with different visual and sonic atmospheres.

### **10.5.6 Integration of Scenography, Lighting and Soundscape**

Design and creation of a scenography, light setting and composition of a soundscape are well-known to the three artistic people. As a consequence, the visual and sonic atmosphere presented inside the cube already at CAVI communicated a jungle-like environment to the audience. The lighting was, however, not well integrated with the scenography and the soundscape. The reason was that the time-consuming develop-

ment of the technical platform for the light control prevented the light designer from spending enough time with fine-tuning of the light tables. As a result, the lighting was too stationary: only the ultraviolet light was used during the night period, even though the soundscape clearly expressed different sonic atmospheres within the night period; the spotlights used in the morning period were too constant and too bright so changing shadows from fading spotlights illuminating the fixed elements of the scenography were not used to express, e.g., a sunrise.

Unfortunately, the light designer did not have time to change the lighting until we started working on the NIC presentation. At that time, the scenographer and the light designer made a thorough change in the visual appearance, as can be seen in Figure 10.16. They integrated the colours and the shapes of the scenographic elements with the lighting: coloured filters and cardboard with patterns of holes were placed in front of spotlights. Furthermore, a dynamic visual atmosphere was created by having periods of rapid light changes causing rapid changing patterns of shadows to sweep the interior of the cube, and these periods were integrated with the soundscape. For example, the sunrise in the jungle morning was expressed as yellow and red spotlights fading up, accompanied by soft, high-pitched and calm sounds, while the jungle night was expressed as a short period of down fading and flashing spotlights focused inside plastic pipes accompanied by low-pitched, raw sounds moving slowly around in the eight speakers, followed by a long period of only ultraviolet light accompanied by sounds moving fast around the top four speakers like the sound of a storm.

At the CAVI and the NUMUS presentations, the audience had to spend 15 minutes with their heads through the peepholes to experience the four time periods. Because of several minutes with almost no Bug activity in the CAVI and the NUMUS scenarios, and because of the stationary lighting, most people spent only a few minutes watching the jungle life. At the NIC presentation, the time periods were shortened so people only had to spend 6 minutes to experience all periods of the jungle life, and these 6 minutes were made comfortable by the cushions outside the cube. Furthermore, the TV monitors outside the cube and the sonic atmosphere coming from the Bug sounds and the soundscape inside the cube were meant to stir the curiosity of people passing by so they would put their heads into the Jungle Cube.

### **10.5.7 From Path-controlled Navigation to Random Search**

On the floor at CAVI (see Figure 10.14), coloured pathways were used for navigating the Bugs to their homes or to food. A lot of experiments were performed to choose appropriate floor colours. On the one hand, they should be chosen so they could be distinguished by the Bug control program through the floor light sensor and perceived as different locations; on the other hand, they should be part of the colours of the scenography. Five colours were used on the CAVI floor: a black ground, several green areas to signify Weak Bug food locations as well as Strong Bug homes, dark grey areas as Weak Bug homes, and two-coloured pathways of pale grey/white lines

connecting green areas and dark grey areas. The two-coloured pathways made it possible for the Bugs to choose direction either towards home or towards food. To distinguish five colours was, however, just on the edge of what could be reliably perceived with the floor sensor. As a consequence, the path navigation did not work: when Bugs started on the pathways by following the coloured lines they simply turned off the pathway shortly after, and found neither home nor food.

The floor was changed for the NUMUS presentation, as can be seen in Figure 10.15. The main difference was that the pipes and the tufts of straws were now used, instead of green areas, to represent food. In the middle of the floor a large dark grey area represented the Weak Bug home, and the two-coloured pathways now connected the home with the three tufts. The Strong Bugs had no home; they simply slept somewhere on the floor. The reduction of the number of colours to four instead of five, made the Bug colour interpretation more reliable than on the CAVI floor. As a result, a Bug could use the pathways for reliable navigation. Unfortunately, it did not look very insect-like when it followed a pathway. It looked more like a car driven by a drunken driver.

From the very beginning, the scenographer did not like the two-coloured pathways because they dominated the visual appearance, and they only made sense to the Bugs, not to an audience. After all, there are no two-coloured Bug pathways in a jungle. So when we started working on the NIC presentation, the composer suggested to leave out the pathways and use only coloured areas to represent the homes: two large dark grey areas as Weak Bug homes, and several small, pale grey areas as Strong Bug homes. Instead of path navigation, home and food were searched for by random walk. It turned out that random walk was just as efficient as the NUMUS path navigation for the Bugs to find their homes and food. The small size of the Strong Bug homes made sure that the two lonely hunters also slept alone, whereas the large size of the Weak Bug homes had the effect of gathering the Weak Bugs for the night.

### **10.5.8 From Obstacle, Colour Sensing to Making Sense of the Environment**

At CAVI, the Bugs avoided or attacked not only the walls and each other but also the pipes and the tufts. The behaviour that emerged from Bug interaction with the straws and the pipes could be perceived as a Bug eating a meal. After CAVI, the scenographer and the computer scientist discussed how to change the environment so that the Bugs could sense the pipes and the straws as food. A very simple solution was found. We prevented the antennas from touching the walls by placing a low barrier along the four walls. This barrier could be sensed through the front bumper without touching the wall with the antennas. As a consequence, the antennas could be used for sensing tufts, pipes and other Bugs. In order to distinguish tufts and pipes from Bugs, infrared communication was used: whenever an obstacle is sensed through the antennas and the obstacle identifies itself, it is a Bug; otherwise it is a tuft or a pipe. The combination of front bumper sensing, antenna sensing and communication made it possible for the

Bugs to react differently to walls, tufts and pipes, and other Bugs. At NUMUS, the Weak Bugs interpreted the pipes and the tufts as food and enacted an eat behaviour, moving the antennas from side to side in the straws of a tuft or scratching a pipe.

The space of possible sensor values resulting from external stimuli is called the *sensor space* of a Bug. The values in the sensor space are interpreted by the control program to make sense of the environment. The space of possible interpretations is called *perception space* of a Bug. For example, in the box, the perception space included day/night and home. The infrared messages from the Co-ordinator make sure that the current time period is available to the Bug. Infrared messages are also used to identify another Bug encountered in front. Since this use of communication can be considered a substitution for sensors, we also include the current time period and the identity of an encountered Bug in the perception space of the Bug.

The perception space of a Bug has changed a lot since the box environment. In the box, for instance, a Weak Bug could perceive (day/night, food, home, path to home, front/left/right/back obstacle). In the NIC environment the perception space has been enlarged to (morning/day/evening/night, food, home, wall, Weak Bug in front, Strong Bug in front, left/right/back obstacle). The enrichment of the perception space has been achieved by engineering the environment by placing a barrier along the walls. A well-known biological assumption is that animals adapt to the environment. The Jungle Cube might be said to have evolved the other way around: the environment has been adapted to the Bugs. The changes to the environment have always been made with the audience in mind: *the perception and the subsequent reaction of the Bugs should be meaningful to an audience.*

## 10.6 Discussion

The initial robot scenario of Crawlers and Brutes has acted as a framework for our investigation of robots as a medium for artistic expression. Within this framework, we have experimented with robots enacting animal-like behaviour. Hence, we relied on biomorphism to trigger associations in the audience which made the robot behaviour meaningful.

In our experiments we were inspired by behaviour-based artificial intelligence (Brooks, 1991). This line of research within artificial intelligence takes inspiration from biology. Around 1984, Brooks proposed “looking at simpler animals as a bottom-up model for building intelligence”, and “to study intelligence from the bottom up, concentrating on physical systems (e.g., mobile robots), situated in the world, autonomously carrying out tasks of various sorts”. This research has resulted in a variety of mobile robots, also called agents, and flocks of mobile robots or multi-agent systems (Maes, 1994; Mataric, 1997). These agents and multi-agent systems produce what in the eye of an observer is intelligent behaviour. The

behaviour emerges from interaction of primitive components that can hardly be described as intelligent:

It is hard to identify the seat of intelligence within any system, as intelligence is produced by interactions of many components. Intelligence can only be determined by the total behaviour of the system and how that behaviour appears in relation to the environment (Brooks, 1991).

In the mind of an observer, the behaviour-based approach to robotics creates an illusion of intelligence when robots are carrying out tasks like navigation in an office or collecting empty soda cans. Our first aim, within the initial scenario framework, was to create the illusion of animal-like behaviour in LEGO robots. It turned out that the behaviour-based approach, with motivation functions used for behaviour selection, were fruitful in creating the illusion. At our first presentation, NUMUS 2000, the Bugs were indeed perceived as animal-like robots by the audience. Furthermore, the curiosity of the audience was stirred by the fact that small, autonomous and mobile LEGO robots could enact several situations where they were perceived as fighting, mating or searching. Bug–Bug and Bug–box interactions produced these situations. They were not produced by any particular behaviour module, but *emerged* from rapid switching between modules within each Bug, and the dynamics of the interaction of the individual behaviour modules with the world in which the Bug was *situated*. Hence, the explanation for the observed intelligence of a soda can-collecting robot also applies to the observed aliveness of the Bugs in the box. But there is more to it. Collecting empty soda cans is a well-defined task; it is easy to tell whether the robot succeeds. Creating animal-like behaviour is not as well-defined a task as the tasks researchers normally make their agents do. It turned out that the combination of movements and sounds was crucial in our effort to express animal-like behaviour and to create the illusion of creatures with goals, intentions, and feelings. If the speaker is disconnected the illusion almost breaks down.

Our next choice was to investigate how *traditional media of artistic expression* like scenography, lighting and soundscape can be used to create an environment in which the Bug behaviour appears as more meaningful to an audience than in the box environment. In the CAVI and the NUMUS 2001 presentations, we tried to make the robots enact the intended Weak and Strong Bug scenario with the confrontations of Weak and Strong Bugs as the main generator of dramatic situations. At first, the confrontations were expressed as abrupt and sudden movements accompanied by almost annoying, high-pitched Weak Bug sounds and low-pitched Strong Bug sounds. The result was very dramatic. The number of confrontations taking place could, however, not be controlled. As a result, a sequence of frequent confrontations was annoying both to watch and listen to. Later, we expressed the confrontations less dramatically with the result that the conflict between the Weak and Strong Bugs was not perceived by the audience. As an alternative, we considered using the idea of a central Stage Manager (Rhodes and Maes, 1995), to monitor the number of confrontations and to command the Bugs to express these confrontations more or less dramatically depending on the current confrontation frequency. We abandoned this approach in order to investigate *randomness and emergence as generators of interesting situations*.

Before the NIC presentation, we had a few discussions on how different artistic intentions might be expressed by the Bugs in the cube. We finally abandoned the idea of expressing the conflict between the Weak and the Strong Bugs and of regarding this as the main element of the scenario to communicate to the audience. Instead, our choice was to balance all the expressive elements in the Jungle Cube to create an installation of an ambient, visual and sonic atmosphere – the mimesis of insect life in a jungle. Each of the elements can attract attention from the audience, for example, when the straws start to move, when the soundscape starts the storm, when the lights flash rapidly in the pipes, or when a Weak Bug suddenly sounds scared and lonely because it cannot find a home for the night.

Bøgh Andersen and Callesen (1999) have drawn on “literature, film, animation, theater and language theory”, to describe the idea of using agents as actors “that can enact interesting narratives”. In our framework, we have chosen biology and ethology as inspirations for the scenarios. Instead of telling stories, as suggested by Bøgh Andersen and Callesen, we have used behaviour-based control programs with a simulation of biological behaviour selection as the generative system producing the situations that the audience watches.

Other people have used biological models as a generator of artistic expressions: when Felix Hess created *Electronic Sound Creatures* he was inspired by the sound of frogs (Chadabe, 1997). Hess created small machines with a microphone listening, a little speaker calling, and some electronics. He created the machines so they “react more or less to sounds the way frogs do”. For Hess, “it was a synthetic model of animal communication”. To other people “it was art”.

## 10.7 Conclusion

In an essay on agents as artworks, Simon Penny (1999) has discussed two different contexts in which production of artworks based on agent technology has taken place: an artistic context and a scientific context. Penny argues that the two different contexts will inevitably lead to two different approaches to agent design, “the approach to production of artworks by scientifically trained tends to be markedly different from the approach of those trained in [visual] arts”. As a case example of the different approaches, he compares two works in the Machine Culture exhibition at SIGGRAPH 93 (Penny, 1993): *Edge of Intention* by Joseph Bates and Luc Courchersne’s *Family Portrait*. Penny generalises the difference in approach:

(with apologies to both groups): artists will klunge together any kind of mess of technology behind the scene because coherence of the experience of the user is their first priority. Scientists wish for formal elegance at an abstract level and do not emphasise ... the representation schemes of the interface.

Furthermore, Penny discusses “the value of artistic methodologies to agent design” and describes how artistic methodologies seem “to offer a corrective for elision generated by the often hermetic culture of scientific research”.

In our project, we have tried to avoid the two extreme approaches described by Penny. Throughout the whole project, artistically and scientifically trained people have been involved. As a consequence, we have avoided the two pitfalls of klunging together “any mess of technology”, and of neglecting the audience. The technicians of the team used their mechanical, electrical and programming skills to come up with robust and re-usable technical platforms: the mechanics of the LEGO robots only required minor changes after the first NUMUS presentation; the electrical components in the customised sensors, the robot speaker, and the 9 V to 12 V power converter were not changed after initial testing; and the behaviour-based architecture and the motivation selection have acted as the architectural pattern for the control program since the first NUMUS presentation. Furthermore, we did emphasise “the representational schemes of the interface”. By involving artistically trained people we have managed to integrate scenography, lighting, soundscape and the Bug movements and sounds in the Jungle Cube installation to conjure meaning in the mind of the observer.

The development of different installations, as carried out in this project, might be called *artistic prototyping*: We have tested the potentials for artistic expressions made possible by the technical platforms – for example, the robots – by creating scenarios and environments as installations based on the platforms; and, as usual in artistic practice, the installations have been presented to a general audience. The reaction from the audience has clearly indicated that robots can indeed be used for artistic expression. Whether the latest prototype, the NIC Jungle Cube, can be described as art is for other people to decide.

## Acknowledgment

The Jungle Cube was sponsored by Kulturministeriets Udviklingsfond and the Center for IT-Research.

## References

- Bøgh Andersen, P and Callesen, J (1999). Agents as Actors. In L Qvortrup (ed.), *Virtual Interaction: Interaction in Virtual Inhabited 3D Worlds*, London: Springer.
- Braitenberg, V (1984). *Vehicles: Experiments in Synthetic Psychology*. Cambridge, MA: MIT Press.
- Brooks, RA (1986). A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, RA-2, 14–23.
- Brooks, RA (1991). Intelligence without Reason. In *Proceedings, IJAI-91*, 569–95, Sydney, Australia.
- CAVI, Center for Advanced Visualization and Interaction: [www.cavi.alexandra.dk](http://www.cavi.alexandra.dk).
- Chadabe, J (1997). *Electric Sound*. Englewood Cliffs, NJ: Prentice-Hall.
- Krink, T (draft). *Motivation Networks: A Biological Model for Autonomous Agent Control*.
- Krink, T (1999). *Cooperation and Selfishness in Strategies for Resource Management*. In *Proceedings MEMS-99, Marine Environmental Modelling Seminar*, Lillehammer, Norway.
- LEGO MindStorm: [www.mindstorms.lego.com](http://www.mindstorms.lego.com).
- Maes, P (1994). Modeling Adaptive Autonomous Agents. *Artificial Life Journal*, 1(1–2).
- Manning, P (1985). *Electronic and Computer Music*. Oxford: Clarendon Press.

- Mataric, MJ (1992). Integration of Representation into Goal-driven Behavior-based Robots. *IEEE Transactions on Robotics and Automation*, 8(3), June, 304–12.
- Mataric, MJ (1997). Behavior-based Control: Examples from Navigation, Learning, and Group Behavior. *Journal of Experimental and Theoretical Artificial Intelligence*, special issue on Software Architectures for Physical Agents, 9 (2–3), 323–36.
- Penny, S (1993). Machine Culture. In *ACM Computer Graphics SIGGRAPH93 Visual Proceedings*, special issue, 109–84.
- Penny, S (1999). Agents as Artworks and Agent Design as Artistic Practice. In K Dautenhahn (ed.), *Human Cognition and Social Agent Technology*, John Benjamins Publishing Company.
- Rhodes, B and Maes, P (1995). *The Stage as a Character: Automatic Creation of Acts of God for Dramatic Effect*. Working Notes of the 1995 AAAI Spring Symposium on Interactive Story Systems, Stanford: AAAI Press.
- Roads, C (1996). *The Computer Music Tutorial*. Cambridge, MA: MIT Press.
- Walter, WG (1950). An Imitation of Life. *Scientific American*, May, 42–45.
- Walter, WG (1951). A Machine that Learns. *Scientific American*, August, 60–63.