Chapter 5

Artificial evolution in virtual and real world settings

Daniel C. Dennett

The law of uphill analysis and downhill synthesis (Braitenberg, 1984) states that it is much easier to deduce the behavioural competence of a system whose internal machinery you have synthesized than to analyse the machinery of a black box whose behavioural competence you have observed. This is particularly the case when studying the mind (Dennett, 1994*a*): downhill synthesis can shed light on issues that are difficult to understand by reverse engineering, i.e., trying to analyse an existing system. Artificial Life uses the bottom-up approach to model cognitive competence by examining how simpler systems could have evolved into more sophisticated ones; Artificial Intelligence, by contrast, starts at the top by attempting to build from scratch machines capable of complex tasks. The two methods produce important differences in the architecture of the systems, partly because of the limited knowledge of the designer.

Artificial Life projects usually depend on a form of genetic algorithm, in which bit strings serve as genomes, formal recipes for performing some task or building some device. These are randomly mutated and set in competition against each other, the winners being permitted to replicate and advance to the next round of competition. I illustrate the power and limitations of this method by comparing three projects: two involve selection in an entirely simulated or virtual environment, the third in a real environment. The comparison demonstrates the importance of making the evolutionary setting as close as possible to the real world, allowing for the myriad physical effects of actual environments. Some of the startling results of these projects confirm the power of bottom-up strategies, particularly as the models produce solutions to problems that a human engineer would be unlikely to consider. In Artificial Intelligence, the importance of using the real world, rather than a simulated one, as the challenge has been recognized by those who have moved away from 'bedridden' systems (Dennett, 1979) to build robots that have to move and act in the world. The state of robotics is represented here by Cog, a humanoid robot that is being developed using inputs from the environment interacting with a basic structure.

Computer-generated models of evolution

Evolution both of body form and behaviour occurs even in a model where organisms do not interact and neither the genome nor the developmental program receives any input from the environment. In the simulation program Evolved Creatures (Sims, 1994*a,b*), a set of simple creatures was generated using a series of random numbers as the genetic code. The code determined the phenotype, i.e., the size, number and arrangement of the articulated blocks that made up the body, which had joint sensors and muscles, and the organization of the simple nervous system. These virtual creatures competed for the best locomotory ability either in a virtual liquid representing the sea or on a virtual solid plane that served as a land surface. The winners of these competitions were repetitively bred and mutations were simulated by making random changes in the original genetic code.

In spite of the extreme oversimplification of the creatures and their environments, different body plans emerged and, in later generations, some familiar characteristics, such as symmetry (or near-symmetry) evolved several times. Some individuals were unable to move as a result of a mutation that damaged the nervous system but others evolved unique and unexpected solutions. Some developed the ability to swim when placed in water. Selection was accomplished by an objective, automatic test: each candidate was placed in the simulated liquid space and allowed to behave according to its genotype for a fixed period. The distances from the origin — if any — covered by all the competing genotypes were measured and the winners advanced to the next generation. Within a few dozen generations, several efficient, graceful swimming forms evolved.

When put on land, various forms of locomotion appeared within 15 generations. The blindly automatic selection process was nicely revealed by a lineage that simply measured the distance, of any body part, from the origin: it was 'born' at the origin as an upright tower of connected blocks and simply fell over! In subsequent generations it was taller and taller at birth and even adjusted its entirely rigid form to execute a sort of somersault when it landed, approximately doubling its distance from the origin. The loophole in the selection regime that allowed this creature to evolve was closed in subsequent competitions, making the selective environment somewhat more realistic.

Selection for other behaviours included jumping, phototaxis and competing for control of a cube, rather like the face-off that starts a hockey game. The computer-generated solutions were often completely different from anything that a computer programmer would have devised. Bizarre shapes that utilized friction or angular momentum in novel ways were among the previously unimagined solutions discovered by this evolutionary process.

This virtual world has several shortcomings. Because the physical forces of the real world are largely absent from the program, many limitations are missing, e.g., energetic demands, resource limitations, wear-and-tear and cost/benefit complexity. Another problem is that neither the genome nor the developmental program receives an input from the environment, so interventions, such as changing the length of the genome, were required to enable new abilities, such as phototaxis, to evolve. Lastly, the system contains no noise, which turns out to be an important lubricant in the evolutionary process (see below).

Some of these limitations are taken into account in another simulation, ECHO, which is a general platform for exploring artificial life (Holland, 1995). As in Sims' work, the creatures that have evolved in this simulated environment had to find efficient solutions to oversimplified and idealized problems: locomoting, finding 'food' and 'mates' and avoiding predation by other virtual inhabitants of their virtual world. But the world in this simulation is significantly more realistic: the basic building blocks in ECHO are not the body segments used in Evolved Creatures but resources that can be employed in many ways. They may become body parts but can also form part of the genome, represent energy for moving and acting, or be material resources from which offspring can be built.

The genome is in the virtual world, so that both the developmental process and the length of the genome are visible to selective forces, thus opening up opportunities for environmental novelty to intrude spontaneously into the selection process. This means ECHO is open-ended in a way that Evolved Creatures is not. Evolution requires an abundance of undesigned bits and pieces to serve as raw material for incorporation into designed bits and pieces.

Although sexual reproduction was built into ECHO, one of the biological features that has emerged in simulations is an elaboration of mating systems. Varieties of parasitism, symbiosis and mimicry, and even some crude forms of communication, have also appeared. In spite of the simplicity of ECHO, the fecundity of its evolutionary process is striking: the solutions have an ingenuity that is manifestly not imbued by the creators of the system and that combine biological familiarity and novelty. If we discovered life on a distant planet, we would expect to find both convergent evolution, due to underlying deep similarities posed by environmental problems, and entirely novel forms arising from the differences in environment. Both types of solution are abundant in the alien simulated worlds of Artificial Life.

Entering the real world

The most serious limitation of computer simulations is the lack of input from the real world. This is being overcome by using robots, to which real-world physics apply, programmed using genetic algorithms. A popular robot platform for such experiments is the tiny Khepera, designed and built by Francesco Mondada at the Federal Institute of Technology in Lausanne, Switzerland. It is about the size of an ice-hockey puck, with wheels, three eye spots and vibrissae for collision detection. A simple neural net connects the sensors to the wheels (see *Fig. 25*) and contains a program that allows several individuals to flock and compete. The bit strings that make up the genome completely determine the phenotype. The locations of the eye spots and whether the connections between the sensors and wheels are excitatory or inhibitory is under genetic control. All the connections in the controlling neural net are subject to change over the course of evolution but are not plastic in the individual, unlike the connections in many neural nets that 'learn'.

To enable Khepera-type robots to evolve efficiently, a robot simulator suspended from a gantry has been built (*Figs 24, 25*; Harvey *et al.*, 1997). It was driven by a developmental program similar to that of Evolved Creatures (Sims 1994*a*,*b*). The workings of the Khepera were simulated in the controlling computer at the same time that the real sensory and motor interactions with the world were accomplished by the robot-interface (*Fig. 24*). This obviated the need to accomplish the developmental program specified in the genome by physically rewiring the robots and relocating their eye spots. As each candidate genotype emerged, the computer faithfully built a simulated nervous system following the instructions in the genome, with just three pixels from the built-in TV camera representing the eye spots. Then the computer positioned the gantry at the point of origin in the middle of the table and the genotype was given trials similar to those used by Sims (1994*a*,*b*) but in real time and space.

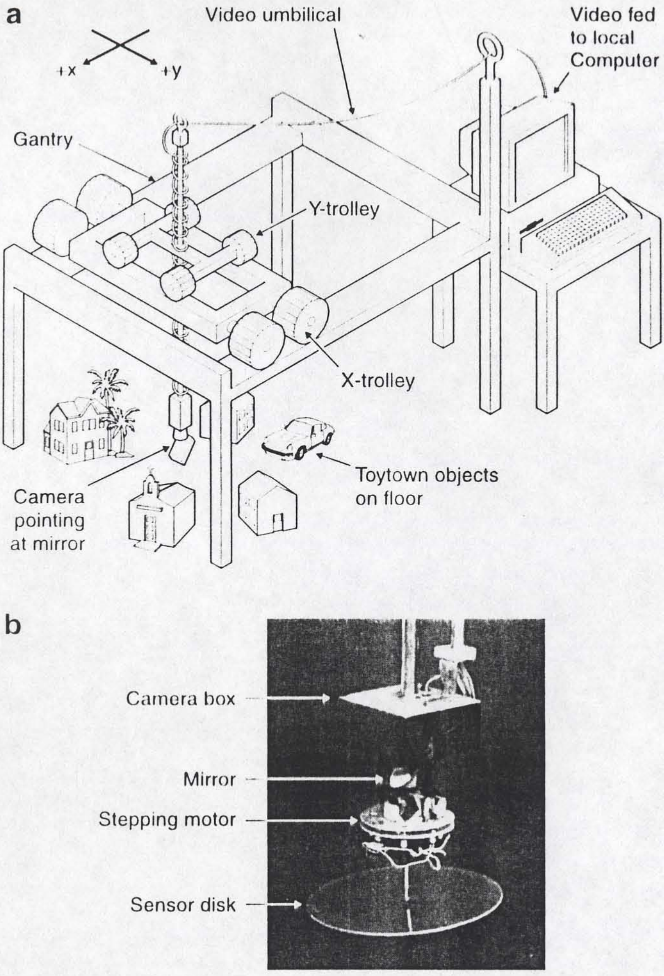

***Figure 24**. The gantry robot, a system for testing the effects of phenotypic differences between simple robots in the real world. **a**, the robot consists of an array of sensors carried on a platform that moves on wheels in* x *and* y *coordinates. Information from the sensors, including the video camera, is fed to a computer where it provides the input to a simulated nervous system. **b**, detail of the array of sensors: the camera inside the top box points down at an inclined mirror, which can be turned by a stepping motor. The plastic disk suspended from a joystick senses bumps. Modified from Harvey et al., 1997.*

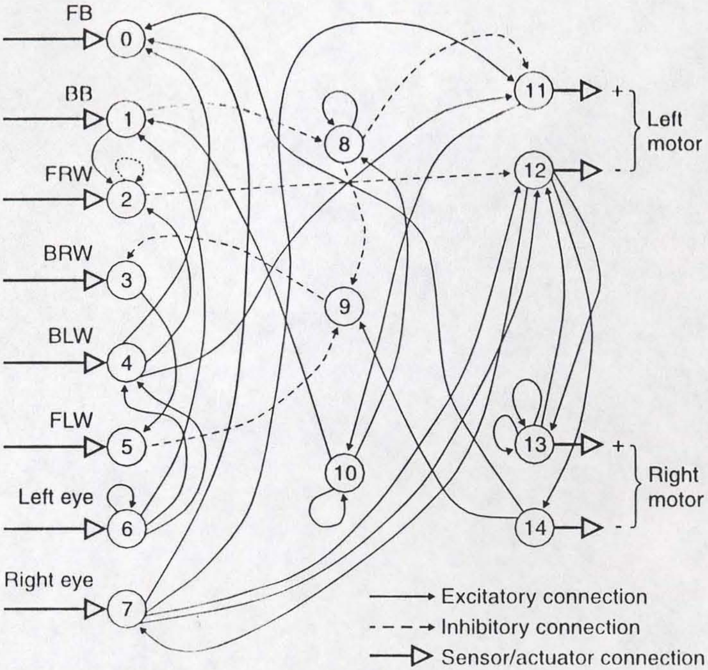

Figure 25. The three-layer neural network of a Khepera like that used to control the gantry robot. Units on the left are initially designated as input units: BB, FB, back and front bumpers; BRW, BLW, FRW, FLW, back and front, left and right whiskers. Units on the right output to the motors for the left and right wheels. Centre column contains hidden units. Modified from Harvey et al., *1997.*

The fitness of the phenotypes specified by the different genotypes was tested in a walled environment, e.g., for phototaxis, they had to learn to avoid a triangle painted on one wall and move towards a square on the other. Each individual was evaluated on three trials and selected on the worst one, in simulation of nature — 'you are only as good as your worst day'. Those with superior performance were allowed to mate, i.e., their genomes were subjected to crossover and mutation in a step of the genetic algorithm and the resulting genomes became part of the next generation.

The robots evolved in some unexpected ways. The nervous systems of the winners developed complex functions that were difficult to determine using reverse engineering and known design principles. Most complex human artefacts, such as engines, computers, aeroplanes and assembly lines, are composed of many single-purpose subsystems carefully isolated from each other to prevent unintended side effects from interfering with their operations. Like biologically designed systems, these robots largely ignored such principles. When Harvey and his colleagues attempted to give some of their robot genotypes a head start by hand-designing first-generation candidates, evolution frequently discarded their handiwork and replaced it with better, but largely inscrutable, solutions. Although these did not fit human ideas of elegance or efficiency, they were very effective, a good example of Leslie Orgel's Second Rule: "Mother Nature is smarter than you are" (see Dennett, 1995).

Noise was very important for effective evolution as it helped the non-plastic nervous system to work better. Although the robots were initially designed with three eye spots, which was thought to be near the functional limit, in several lineages one eye spot came to 'look' in an irrelevant direction, throwing away its input and introducing more endogenous noise into the system. This noise kept the system from settling into sub-optimal states and so was more valuable to the robot than the external information that could have been obtained through a 'seeing' eye spot.

The power of such a simple set of transducers to exploit information in the environment should not, however, be underestimated. In one evolutionary run, conducted by the computer over several days of continuous trials and generations, diurnal and nocturnal subspecies emerged. The ancestors of the diurnal variety had mainly been tested when daylight from the lab windows contributed to the ambient light, whereas the nocturnal version arose from those tested at night. This entirely unintended and unanticipated outcome is a dramatic demonstration of the importance of a real world as the selective environment.

Cog, a humanoid robot

Cog, a life-sized humanoid robot (*Fig. 26*), is being developed at Massachusetts Institute of Technology by Rodney Brooks and colleagues as a tool for studying both evolution and the human mind (see Dennett, 1994*b* and the Cog website*). Cog has two arms, using series elastic joints with a variable spring constant regulated by fast feedback motors. Although it has no legs, it can move its body with the same degrees of freedom as a human body. Four microphones serve as ears and the two eyes have both foveal (narrow angle) and parafoveal (wide angle) cameras. The eyes saccade below the 100msec range at three to four fixations per second, approximately the same rate as in humans. Parts of the body are covered with touch-sensitive plastic skin and detectors including strain gauges and heat sensors approximate to an innate pain system.

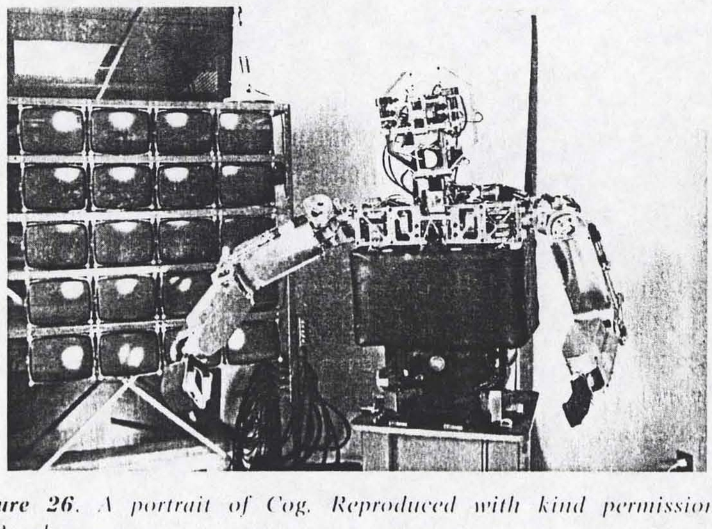

Figure 26. *A portrait of Cog. Reproduced with kind permission of R. Brooks.*

*Cog website: www.ai-mit/projects/cog/Text/cog-shop.html

The original distributed competitive nervous system consisted of 128 Mac2 computer central-processor chips connected in parallel, with no central executive*. The programs that ran on this hardware, written in a new parallel dialect of Lisp, were partly written by hand and partly evolved using either genetic algorithms or various connectionist learning algorithms. Comparing the efficiency and grace of hand-coded versions of some of these programs with those designed by one of these evolutionary processes encourages the use of the evolutionary methods wherever possible.

Cog's movements depend on feedback and it can perform actions requiring sophisticated motor control, such as manipulating a 'Slinky' toy between its hands, swinging a pendulum back and forth, and hammering a nail into a board, with the joints absorbing much of the shock. It can recalibrate its movements to accommodate for 'growth' and wear-and-tear. For example, the software rapidly compensated for a new set of arms that differed in length and weight from the previous ones and for changes in eye position caused by slippage in the equipment.

Having equipped Cog with hardware and basic software, the intention is to see how it develops from its present 'infancy'. Projected developments are set out in *Figure 27*. The distributed nervous system is provided with separate modules to accomplish different tasks, an organization reminiscent of domain-specific learning (see Gallistel, this volume). Communication between the modules is not designed in advance but emerges with time through their interaction. We hope that the systems that have evolved to solve certain problems will become modified to be available for solving other problems. We anticipate that an intelligent Cog will not have a central processing module but that the task-related modules will develop to access each other to some degree, e.g., to enable a temporal problem to be mapped on to

* This hardware has been superseded in the summer of 1998 by a second-generation architecture with greater power and ease of use.

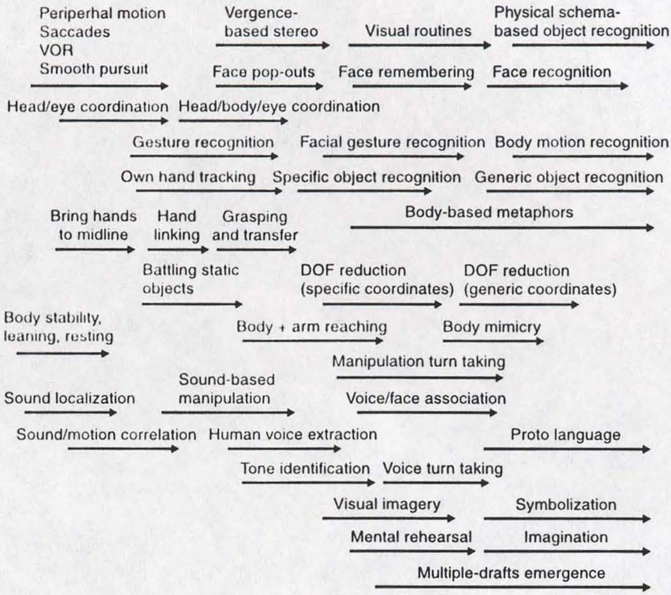

***Figure 27**. A scheme of Cog's projected development. Tasks progress from simple on the left and at the top to more complex and abstract on the right and at the bottom. Reproduced with kind permission of R. Brooks.*

another dimension, such as the use of an analogue clock to determine time elapsed. Humans achieve this mapping effortlessly and continually use it as a useful crutch for thinking. Cog may discover some mappings of its own that will enable it to exploit its own competences.

The results from Cog may not resemble the very complex behaviours, task performance or consciousness that characterize the functions of the human nervous system but there may be several ways to build a consciousness. In one sense Cog is already a success: according to a Chinese legend, a sage was fishing in the river with a straight pin; this curious news reached the emperor, who was so puzzled that he went to see the man and asked him what he expected to catch; the sage replied, "You, my dear friend". Cog is a preparation for examining a problem, a stimulus to think about cognitive problems from a different perspective, without assuming *a priori* which are the hard problems.