Design for a Brain Revisited:
The Neuromorphic Design and Functionality
of the Interactive Space 'Ada'

Kynan Eng, David Klein, Andreas Bäbler, Ulysses Bernardet, Mark Blanchard, Marcio Costa[1], Tobi Delbrück, Rodney J. Douglas, Klaus Hepp, Jonatas Manzolli[1], Matti Mintz[2], Fabian Roth, Ueli Rutishauser, Klaus Wassermann, Adrian M. Whatley, Aaron Wittmann, Reto Wyss and Paul F.M.J. Verschure

Institute of Neuroinformatics, University/ETH Zurich, Switzerland
[1]Nucleo Interdisciplinar de Comunicacao Sonora, University of Campinas, Brazil
[2]Psychobiology Research Unit, Tel Aviv University, Tel Aviv, Israel

SYNOPSIS

While much is now known about the operation and organisation of the brain at the neuronal and microcircuit level, we are still some way from understanding it as a complete system from the lowest to the highest levels of description. One way to gain such an integrative understanding of neural systems is to construct them. We have built the largest neuromorphic system yet known, an interactive space called 'Ada' that is able to interact with many people simultaneously using a wide variety of sensory and behavioural modalities. 'She' received 553,700 visitors over 5 months during the Swiss Expo.02 in 2002. In this paper we present the broad motivations, design and technologies behind Ada, and discuss the construction and analysis of the system.

KEY WORDS

Reprint address:
Kynan Eng
Institute of Neuroinformatics
University/ETH Zurich
Winterthurerstrasse 190
CH-8057 Zurich, Switzerland
e-mail: kynan@ini.phys.ethz.ch

1. INTRODUCTION

Mankind has a long history of building successively more elaborate approximations of natural organisms. Early examples from the 1700s include the humanoid automata of the brothers Droux and Jacques de Vaucanson's mechanical duck. Modern artificial organisms, such as the robot dog Aibo (Sony) and humanoids SDR-4X (Sony) /3/ and Asimo/P3 (Honda) /44/, display more advanced capabilities than their predecessors. Each case is an example of the application of some of the known operating principles of an organism to the instantiation of a particular artefact. Since all known animals above a certain level of complexity have brains, it is only natural that some sort of approximation to a brain often finds its way into these artificial systems.

It could be argued that building this sort of system is merely the mechanical application of a known principle, without generating new knowledge. Indeed, the above examples were mainly constructed for entertainment purposes. In the 20th century, however, a more deliberate effort was commenced to construct machines, such as Grey Walter's turtles /58/, in order to advance our understanding of the brain (see /13/ for an overview). In this tradition Ashby published his influential proposal 'Design for a brain' in 1952, in which he proposed that the principles of cybernetics could be applied to the study of the brain /5/. This pioneering work explored different applications of feedback control to neuronal circuits. Absent from this proposal, however, was the physical realisation of such a control scheme in a real-world behaving

system. The process of building a real system provides a reality check that can be used to eliminate potential models. Real-world systems must satisfy constraints such as power consumption, physical appearance and real-time behavioural performance that many models do not consider /57,62/. In particular, the construction of a real-world artefact requires that an explicit link between neuronal control structures and overt behaviour be elaborated. In addition, being forced to address real-world constraints can also have side benefits in the form of new technologies that are developed to address specific problems, yet find different fields of application.

The exercise of building artificial organisms allows multi-level testing to a degree that would not be possible in a natural organism. For example, many animal experiments in neuroscience try to link neural activity with observed behaviour by *in vivo* cell recording during behavioural experiments. While recording techniques are constantly improving, the range of data that can be collected from any one experiment is small since it is difficult to record from multiple sites simultaneously. As well as this, the need for recording electrodes attached to cumbersome equipment makes experiments with free-roaming subjects even more difficult. If we then want to add further simultaneous data recording using techniques such as optical imaging, fMRI or EEG scans, the technical problems become insurmountable. We encounter the problem of our measurement techniques interfering with each other, as well as with the functioning (or even the lifetime) of the organism itself. Artificial organisms offer a way to avoid these problems. They can be designed to allow extensive data recording without affecting normal operation. When designed to incorporate models of brain function based on what has been learned from natural systems, artificial organisms also serve to help validate or invalidate different theories. In addition, the use of a synthetic approach allows the systematic manipulation of components of the nervous system and the environment in which the system operates that cannot be achieved using biological systems. Another important feature of this approach is that it supports highly controlled repeatability of experiments that is difficult to achieve in complex behavioural paradigms.

These considerations led us to construct 'Ada', an artificial organism in the tradition of many that have come before. However, 'she' has one important difference from any known organism: she is an interactive space, developed for the Swiss national exhibition Expo.02 located in Neuchâtel. Conceptually, she can be seen as an inside-out robot with visual, audio and tactile input, and non-contact effectors in the form of computer graphics, light and sound. Visitors to Ada are immersed in an environment in which their only sensory stimulation comes from Ada herself (and other visitors). Like an organism, Ada's output is designed to have a certain level of coherence and convey an impression of a basic unitary sentience to her visitors. She can communicate with them collectively by using global lighting and background music to express overall internal states, or on an individual basis through the use of local light and sound effects.

The Ada project builds on the tradition of Grey Walter and Ashby, but extends it further towards the development of real-world artefacts that show a complexity that approaches that of real biological systems. We argue that it is through the construction of such systems that we can advance our theories on brain function /57/. A key feature of this approach is that it allows the incorporation of a multi-level perspective from neurons to circuits and behaviour, while allowing full experimental control and analysis. In this respect we envision an approach based on a constructive synergy between experimental and synthetic methods.

To realise Ada, several simultaneous lines of research and development were pursued. Topics under investigation include:

- Auditory processing and localisation;
- Multi-modal visual and tactile tracking of humans;
- Communication using visual and sonic cues;
- Real-time homeostatic control systems;
- Human-machine interaction via whole-body locomotion of visitors;
- Large-scale sensory and behavioural integration;
- Learning of behaviour.

Development of Ada commenced in late 1998 and ramped up to a maximum team size of about 25

people. The main milestones along the way were a series of about ten increasingly complex public demonstrations and tests that started with an interactive music composition system at the computer fair Orbit98. The technical development team came from many different disciplines, ranging from biological sciences through engineering and physics to musical composition. There was also a team of architects, artists, publicists, scenographers, site managers and guides for handling the production and operation of the exhibit. A total of over 100 people were directly involved in the construction and running of Ada. The exhibit ran continuously for 10 to 12 hours a day during 5 months from 15 May to 20 October 2002. During this period 553,700 visitors interacted with the space.

This paper presents an overview of Ada's sensors and effectors, system architecture and behaviours. Three aspects of the system involving neural processing are discussed in detail: auditory processing, learning of behaviours for influencing visitor positions in the space, and a homeostatic scheme for behaviour selection and emotional expression. An overview of the operational results and a description of the data collection strategies are also given. We show how data collected at many different levels can be combined to gain a coherent picture of Ada's operation. The system presented here demonstrates how the construction of real-world artefacts can facilitate the development and evaluation of theories on brain function. Moreover, Ada shows that we now have the technological capabilities to construct artefacts of a level of complexity approaching that of biological systems.

2. SENSORS AND EFFECTORS

Figures 1 and 2 show an overview of the Ada exhibition layout and the main Ada space. The sensors and effectors within Ada consist of (including auxiliary exhibition areas) 15 video inputs, 367 x 3 tactile inputs, 367 x 3 temperature sensors, 9 audio input channels, 46 mechanical degrees of freedom, 17 output audio channels, 367 x 3 floor tile lights, 30 ambient lights and 20 full-screen video outputs. All of these inputs and outputs can be addressed independently, giving a rich
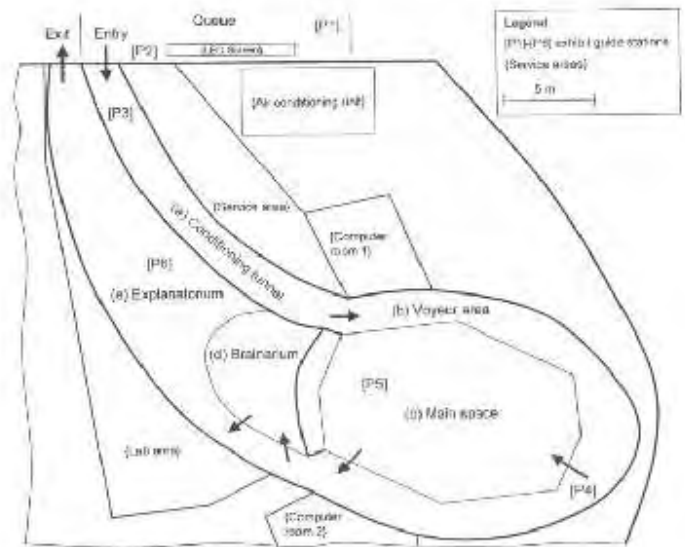

Fig. 1: Floor plan of the Ada exhibit. The regions are labelled (a) to (e), respectively. (a) Conditioning tunnel: Visitors were introduced to Ada's components. (b) Voyeur corridor: Semi-transparent mirrors allowed a view of what happened inside Ada self. (c) Ada self: The Ada main space for visitor interaction. (d) Brainarium: Room with six monitors providing information and real-time graphical displays showing the current dynamics in Ada's control system. (e) Explanatorium: Visitors were provided with background information on the exhibit. The arrows indicate visitor flow. The total surface area of the exhibit was 427 m², of which the main Ada space occupied about 160 m².
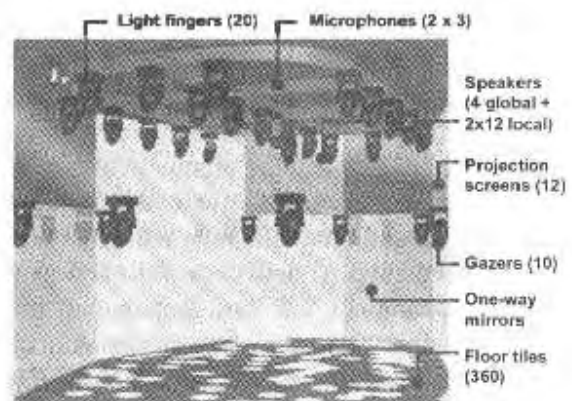

Fig. 2: Overview of Ada main space. The main space is about 160 m² in area and 5 m high. The main sensory and effector components are indicated. The numbers in parentheses specify the amount present. See text for further explanation.

array of sensory modalities and output possibilities. Ada has the following sensory capabilities:

- **Vision:** Pan-tilt cameras called *gazers* (Fig. 3) are available to Ada for focused interactions with specific visitors. The cameras have zoom and digital filtering capabilities that are controlled on-line.

- **Hearing:** There are clusters of three fixed microphones each in the ceiling plane at 5 m above the floor surface. Using these sensors Ada is able to localise sound sources by triangulation. In addition, basic forms of sound and word recognition and pitch and musical key extraction are available.

- **Touch:** Ada has a 'skin' of 0.66 m wide hexagonal pressure-sensitive floor tiles /15/ (Fig. 4) that can detect the presence of visitors by their weight. Each contains a microcontroller and is connected to an industrial automation bus called Interbus.

- **Temperature:** Within each floor tile is an 8-bit temperature sensor. In Ada, this sensor was used only for measuring the internal floor tile temperatures to ensure that the floor did not overheat.

As well as sensing, Ada can also express herself and act upon her environment in the following ways:

- **Visual:** Ada uses 12 LCD projectors to create a 360° surround projection above the mirrored wall that surrounds the space, called BigScreen. These projectors collectively show a single, unified display of 3D objects covering multiple screens in real-time. Key elements of BigScreen for visual communication are the background image, a closed tube reflecting the dynamics of a network of coupled oscillators, and independent graphics windows that can contain still images or live video that can move with smooth transitions between screens. There is also a ring of ambient lights for setting the overall visual tone of the space. Local visual effects can be created using the red, green and blue neon lights in each floor tile in Ada's skin. This output modality can express Ada's internal states, interaction dynamics or localised specific effects for guiding and visitor feedback.
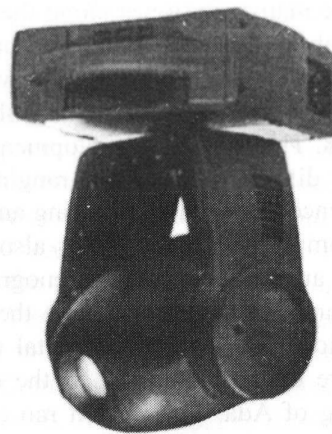

Fig. 3: Martin MAC 250 light finger, normally mounted hanging from a bar. The light finger has a 250 W lamp, a 2-axis 16-bit pan-tilt unit and multiple light colour/shape filters. The gazer is based on exactly the same mechanism, except that the light is removed and replaced with a CCD zoom camera block.
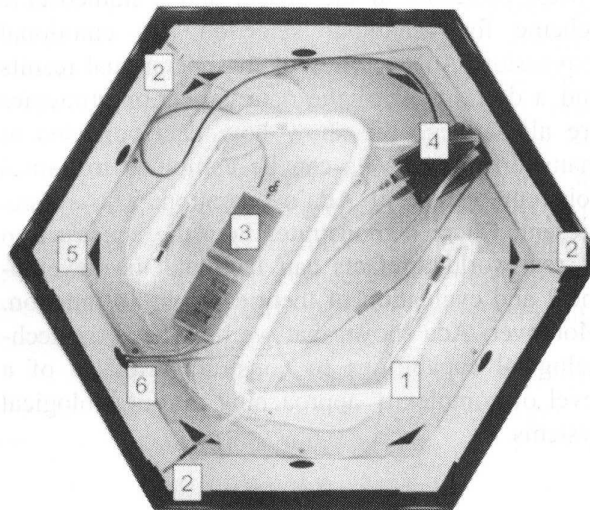

Fig. 4: Floor tile, shown without the glass top. The tile is about 66 cm across and 15 cm deep. Red, green and blue neon tubes (**1**) take up most of the internal space. Three pressure sensors with cables protruding from them (**2**) can be seen near the corners around the edge of the tile. The white box near the centre of the tile is a computer-controllable neon dimmer (**3**). The circuit board at the top-right (**4**) contains a power transformer, microcontroller and network communication to the host computer. Holes can be seen for drainage (**5**) and cable routing (**6**) in the bottom and sides of the tile, respectively.

- **Audio:** Ada is able to generate a wide range of local and global sound. These sounds can be distributed across the entire space or localised using a matrix mixer. She expresses herself using sound and music composed in real-time on the basis of her internal states and sensory input. The composition is generated using a system called Roboser /60/ and performed using a sampler.

- **Touch:** Ada has 20 16-bit pan-tilt light fingers (Fig. 3) for pointing at visitors or indicating different locations in the space. They are standard theatre lights on a serial bus called DMX, which is also used to control the ambient lights and the gazers. Light fingers can emit light of different colours and with different mask (gobo) and focus effects.

A complete list of all devices and manufacturers can be found in Appendix 1.

3. NEUROMORPHIC DESIGN PRINCIPLES

We describe Ada as a neuromorphic system. This means that the same design principles were used to build Ada as are seen in natural organisms. In this section, we briefly survey some of the literature on natural nervous systems, and extract some key concepts that were used in the design of Ada.

3.1.Modularity and encapsulation

The nervous systems of all higher animals, while highly distributed, are not anatomically homogeneous. Rather, they exhibit a high degree of modularity. Even the peripheral nervous system contains modules: far from being a simple conduit for commands to muscles, the spinal cord of many animals can perform sets of coordinated muscle contractions. For example, electrical stimulation of certain regions of the frog spinal cord results in a smooth leg trajectory to an equilibrium point in its workspace. The leg can be interpreted as being under the control of a force field varying in both space and time /7/. It has been suggested that collections of these spinal modules serve as motor primitives, which are used in superposition by higher brain areas to produce a large variety of

movements without having to deal with complex inverse kinematics /37/. This encapsulation of the inverse kinematics problem - a very difficult procedure using Cartesian geometry - may serve to reduce the state space for motor learning to make it more tractable.

Evidence exists that the modularity of motor action extends even further than this. Experiments with monkeys show that complex, coordinated movements can be induced by localised micro-stimulation of the motor cortex /24/. This result suggests that even very complex high-level behaviours can be regulated by the modulation of relatively small populations of neurons. Another example of this design principle can be found in the cerebellar mechanisms underlying classical conditioning. In this case only a few neurons in the deep cerebellar nuclei will control the activation of a complete conditioned response /26/. This is only possible because the motor commands at this level of organisation are independent of their translation to the detailed kinematics and dynamics of the skeletal motor system.

These results suggest that the motor system is designed around a hierarchy of encapsulated modules in which increasingly more abstract and elaborated behavioural patterns are encoded. In this context we hypothesise that at least three levels of encapsulation must be distinguished: specific movements of individual effectors define actions, patterns of actions define behaviours, and sets of behaviours define behavioural modes.

3.2. Data reduction, abstraction and representations

The physiology of the monkey inferotemporal cortex indicates that neurons in this region form abstract representations of visual objects /32/. This representation is considerably abstracted from the sensory input, and involves a large degree of data reduction to a very compact representation. Such a scheme may have two uses: firstly, as an efficient way of storing behaviourally relevant data, and secondly, as an important component of a decision-making mechanism. Decision-making requires that operations are performed on representations that pertain to the complexity of the actions the organism can generate /57/. This complexity is far lower than that provided by the information that is

processed by its sensory systems. For instance, navigation behaviour partly depends on the place field representations found in the hippocampus /42/. Place field responses are very sparse, yet they integrate complex information from both external and idiothetic cues. As far as decisions on navigation are concerned, however, the representation of a location in space they provide is a data abstraction that pertains to the task. In contrast, the detailed representation of the local and global properties of (for instance) the visual cues that give rise to this representation are not relevant to the decision-making process.

These considerations led us to implement Ada's sensory processing in such a way that the high-level abstract representations of events and objects that are created are constrained by Ada's tasks. These representations directly pertain to the actions and behaviours Ada can generate, i.e. visitors and their actions.

3.3. Behavioural modulation

The psychologist Abraham Maslow formulated a system of behavioural motivation for humans based on *needs* /33/. According to him, needs came in two main types: *deficit* needs and *being* (self-actualisation) needs. Deficit needs form a hierarchy in order of necessity for survival: physiological, safety, love, and esteem. The deficit needs operate in terms of *homeostasis*, inducing behaviours to remove the deficit when a particular need is not satisfied. Once all deficit needs are satisfied, the organism can start to realise self-actualisation, which is not a deficit need because achieving self-actualisation does not remove the need to continue achieving it. The concept of homeostasis has been widely applied to the study of the brain, for instance in relation to the mechanisms underlying sleep /8/ and feeding /46/. In general, homeostasis expresses a regulation process that aims at keeping a set of essential variables in a specified range. In the case of sleep, it has been suggested that the regulation of the metabolic and energetic needs of a living system contributes to the cyclic control of its overall activity level between stages of sleep and wakefulness. It has been further generalised to the process of maintaining the organism as a whole in the notion of autopoiesis /50/.

Although Maslow's formulation was primarily directed at humans, and it is questionable whether needs such as 'esteem' and 'self-actualisation' could be measured reliably, there is some evidence to support his concept of homeostasis to select between possible behaviours in order to reduce needs. For instance, it has been suggested that the midbrain central grey region may function as a modal command region for eliciting certain behaviours, as seen in lesion studies in rats /17/. Such a region could form the basis of a system for behaviour selection based on the modulation of activity in relatively small populations of neurons by a homeostatic goal-achieving and/or need reducing control system. Hence, we hypothesise that behaviours are organised around behavioural modes that are defined by the mechanisms for homeostatic control, that support self-preservation, and that this requires the modal organisation of these behavioural subsystems.

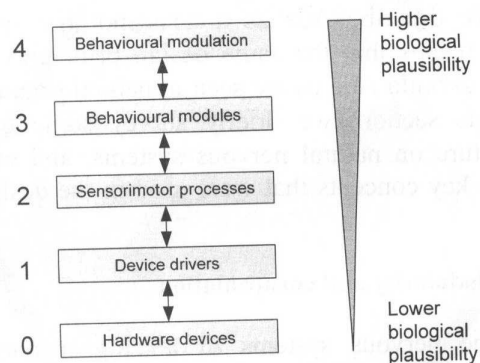## 4. METHODS: DESIGN IMPLEMENTATION

Fig. 5: Ada architecture overview showing layered structure. The lower levels are generally less biologically plausible than the higher levels, with notable exceptions (e.g. the auditory system).

Ada's architecture can be roughly sketched out as a series of levels (see Fig. 5), with a gradient of decreasing biological plausibility as the proportion of traditional procedural code increases. Each level contains modules that communicate with other modules in the same layer, as well as with modules in adjacent layers. The metaphor used is that of distributed brain-like computation, characterised by tight coupling within individual modules and loose coupling between modules. The underlying

software is a hybrid mixture of simulated neural networks, agent-based systems and conventional procedural or object-oriented software. The types of computations performed at each of the different levels are summarised in Table 1.

Different communication protocols are used to connect the components of the system: a TCP/IP socket-based protocol for the simulated neural networks, and an asynchronous message-based middleware for data transfer between agents and behavioural modules. Lower levels closer to the hardware have more specialised protocols. The system is programmed using a mix of C, C++, Java

and a neural simulation environment called IQR421 /6/.

Figure 6 shows the actual implementation of the system architecture used in Ada. The different elements of the architecture are discussed in more detail in the following sections.

Like the nervous system of real animals, Ada's architectural design is highly distributed, with about 100 processes running in parallel. The layering concept provides a useful framework for thinking about the design process and controlling the proliferation of connections between processes. Actions and behaviours are implemented as low in

TABLE 1

Descriptions of functionality and types of software found at different levels in Ada

| Level	Functionality	Software
4: Behavioural modulation	Goal function evaluation, behaviour mode selection, emotional model	Simulated neurons (IQR421 /6/)
3: Behavioural modules	Coordinated high-level interactions	Simulated neurons, software agents
2: Sensorimotor processes	Filtering of raw input data	Procedural or object-oriented code
1: Device I/O drivers	Interface to hardware	Procedural or object-oriented code
0: Hardware devices	Motor control, sound production, sensor reading, light setting	On-device logic and/or procedural firmware
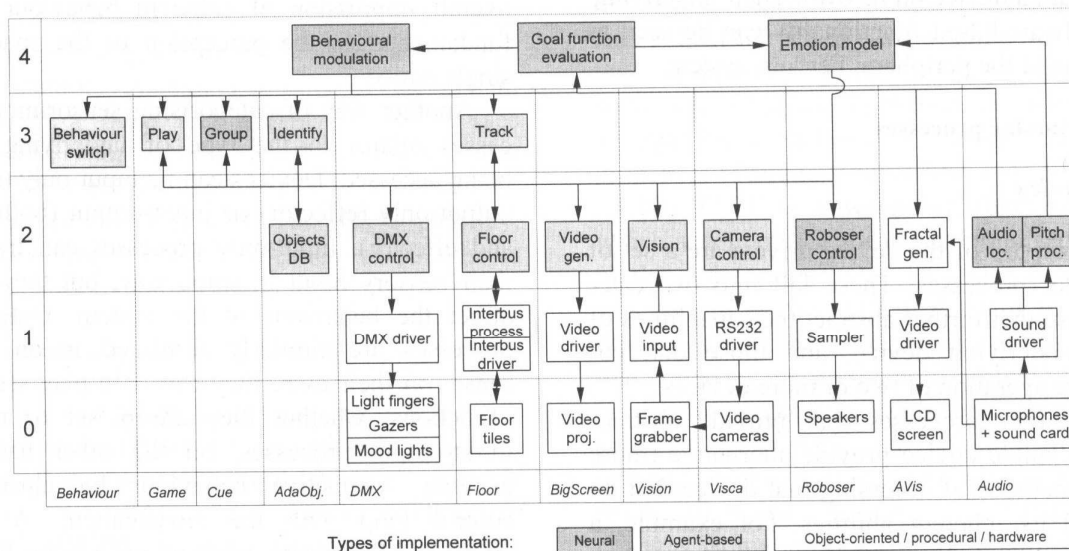
Fig. 6: Ada system architecture, organised into conceptual layers and servers. Numbers in the left column correspond to the layers depicted in Figure 5. See text for explanation.

the processing hierarchy as possible, so that they can be deployed directly by upper layers without explicit knowledge of their internal workings; i.e. we invoke the principles of encapsulation and modularity. One simple example of this can be seen in the floor tiles, which contain simple reactive behaviours even at the hardware level that can be either used or overridden by subsequent layers.

4.1. Device drivers

Many different types of hardware devices are used in Ada. For the system to achieve coherent overall behaviour, all of these devices need to be controlled in parallel and in real-time. As is the case in many other large modern artificial systems, this is achieved by interfacing all of the components to a common computational substrate; i.e. a computer cluster. Various combinations of communications protocols and software drivers are used, as summarised in Appendix 1. Every device driver resides on a *server*, a set of programs that controls the operations related to that device.

Highly complex sensorimotor capabilities may exist for particular devices, but their scope of control is strictly local. The device drivers abstract these capabilities to make the problems of high-level control more easily tractable. At the same time, they implicitly define what is physically possible with a device on its own. Their underlying, local, hardware-based functionality can be seen as an analogue of the peripheral nervous system.

4.2. Sensorimotor processes

4.2.1. Overview

Sitting on top of the device drivers are a set of sensorimotor processes. These fall into two categories: those that further abstract the capabilities of a single underlying device, and those that co-ordinate the operation of two or more devices.

Sensorimotor processes that provide abstractions for a single device provide alternative (more compact) methods of accessing that device, but do not extend its inherent abilities. For example, a DMX-controlled light finger has pan-tilt settings that explicitly set joint angles on the light mechanism, so it is possible to make a light finger point to

any point in its working envelope by specifying two numbers. However, the frame of reference of the behaviours that Ada generates is a 3D Cartesian space, requiring all sensors and effectors to be mapped, implicitly or explicitly, to this common frame of reference for efficient operation. For instance, the gazers, light fingers, localised sound output, BigScreen video windows, floor tracking and floor effects must all be aligned to the same physical space. Hence we encapsulate the necessary kinematics to allow the light finger to be accessed using Cartesian coordinates. In addition, humans also generally find it easier to think about the problem in a 3D Cartesian space. A single device may have many encapsulations, providing convenient access to different subsets of its capabilities, but the encapsulations add nothing intrinsic to what the device can do.

In contrast, sensorimotor processes that access multiple devices can produce qualitatively different behaviour that extends the overall capabilities of those devices *as viewed by an outside observer*. The key to extending observed device capability lies in the temporal coordination of device operation. For example, if a process simultaneously directs two light fingers to illuminate the same location, an observer will draw the conclusion that there is some sort of connection between the two events. Many such paired events build up an overall impression of coherent behaviour that is fundamental to the perception of the space as a single entity.

Another way of categorising sensorimotor processes relates to the type of underlying device being accessed. Devices can be input-only (sensor), output-only (effector) or input-output (both sensor and effector). Input-only processes can transform their sensory input in some way, but they cannot affect the behaviour of the system. Output-only processes are similarly restricted in an inverse sense: on their own, they can only play sequences of actions, whether they are preset or random. Input-output processes, on the other hand, can generate interactive behaviour by closing the control loop with the environment. A simple example of interactive behaviour can be found in the floor control process, which can use the floor as both a sensor and effector to light up tiles that are

loaded above a certain threshold. In practice, there are no instances of sensorimotor processes in Ada that directly access the device drivers of multiple input and output devices. While in principle there is nothing to prevent a sensorimotor process from doing both input and output interfacing, in a large system it quickly gets unwieldy from a modularity and software maintenance viewpoint. This type of distributed interaction is reserved for the next level.

Sensorimotor processes reside on *servers*, according to the device drivers with which they communicate. Each server controls all of the devices of one particular type, and handles resource allocation requests from higher-level processes. Sensorimotor processes only access data or devices on the server that they belong to. Table 2 lists the most important sensorimotor processes and the server to which each one belongs.

4.2.2. Biologically inspired auditory processing: An attentionally modulated stereausis model

As an example of neural processing occurring at the sensorimotor level, we take the Audio server. Ada's auditory system provides information about the most interesting visitors in the space. Visitors become 'interesting' from an auditory perspective by clapping their hands, speaking, or whistling. The task of the auditory system is thus to detect, classify, and localise salient sounds made by the visitors. It does so in an acoustically cluttered environment, using a simplified model of the mammalian auditory system. The core of the system consists of a *What* and a *Where* pathway (see Fig. 7). The *What* pathway, which is modelled after the monaural auditory pathway, is concerned with the identification of sounds. The *Where* pathway, being concerned with the location of sounds, is modelled after the binaural auditory pathway.

The What *pathway*

The *What* pathway detects patterns in a spectrotemporal representation of the microphone signal. Similar to its biological counterpart, it does so using neurons with specific spectrotemporal receptive fields (STRFs) /1/. Each neuron in our system codes for a different type of sound. The types of

sound include phonetic elements (i.e. certain vowels and consonants), claps, and pitches. In order for a sound to be detected, 1) the neuron corresponding to that sound must be active (membrane potential above threshold) and 2) the neuron's activity level must be higher than any other active neuron in the population (winner take all).

The spectrotemporal representation and STRFs

The spectrotemporal representation of the audio signal is similar to that produced by the subcortical monaural auditory system of mammals. In the mammalian cochlea, an incoming sound is passed through the biological equivalent of a bank of band-pass filters /14,59/. As a result, neurons in the auditory nerve respond to a restricted range of sound frequencies, which varies systematically from low to high frequencies as a function of the place in which the neuron innervates the cochlea. This 'tonotopic' organisation is a key feature of the mammalian auditory system, and is preserved through several nuclei to the primary auditory cortex (AI) /41/.

Our spectrotemporal representation is produced via the short-time Fourier transform /12/, which is computed in real time using the fast Fourier transform (FFT). The FFT of the signal is computed about 22 times per second. Only the amplitude of the FFT is subsequently used. The linear frequency axis is then warped to a logarithmic frequency axis, thus approximating the tonotopic axis of the cochlea. Finally, the amplitudes are compressed in a logarithmic fashion, in accordance with the properties of auditory nerve fibres /41,59/.

The STRF is a weighting function, or filter, which acts upon the spectrotemporal representation of the audio signal /16/. The form of each STRF was chosen as the optimal detector for a specific type of sound, based upon recordings containing that sound in the presence of realistic background noise. In accordance with neurons in primary auditory cortex (AI), the temporal-filter characteristics of the STRFs were all band-pass. As a result, all neurons respond only when the sound changes. Some, however, respond to faster changes and others respond to slower changes /30/.

TABLE 2

Summary of servers and most important sensorimotor processes

Server	Device	Sensorimotor processes
Floor_server	Floor tile network	Read raw load (input) Read calibrated adaptive-thresholded load (input) Determine loaded tiles (input) Determine tiles with isolated persons (input) Set single tile colour (output) Set floor colour (output) Set global floor effect (output) Set local tile effect (output) Interactively set tile effects for loaded tiles (input/output)
Vision_server	Gazer camera	Read image (input) Read video stream (input) Record image (input) Record video stream (input)
Visca_server	Gazer camera	Set camera zoom (output) Set image parameters (output) Set digital effects (output)
DMX_server	Ambient light	Set colour (output)
DMX_server	Gazer pan-tilt unit	Set Cartesian position (output)
DMX_server	Light finger	Set Cartesian position (output) Set colour wheel (output) Set 'gobo' light filter (output)
BigScreen	Graphics output	Set background texture (output) Set foreground object texture (output) Set lighting parameters (output) Set tube parameters (output) Set tube perturbations (output) Set still grabbed image parameters (output) Set live video window parameters (output)
Audio	Sound input card	Determine frequency spectrum (input) Determine sound location (input) Determine sound type (input)
Roboser	Sound output	Set MIDI parameter (output) Set local sonic event (output) Set sound library (output) Set style (output)
Object_server	(internal database)	Save visitor data Read visitor data Append to visitor data Delete visitor data
*Game_server	–	–
*Cue	–	–
*Behavior_server	–	–

* Some servers do not have any sensorimotor processes; they use the sensorimotor processes of other servers to implement their functionality.

The Where pathway

In the auditory brainstem of mammals, information from both ears is combined in order to estimate the location of sound sources. For acoustic frequencies below approximately 5 kHz, sound location within the azimuthal plane is based largely upon the computation of inter-aural time delays (ITDs) /41/. Similarly, in the *Where* pathway of our system, sound location is based upon the computation of inter-microphone time delays. However, the computation and utilisation of ITDs by Ada diverged significantly from the mammalian solution because, unlike mammals, Ada does not have a head, her ears were located 5 meters above the localisation plane, and she was required to accurately localise sounds in two dimensions.

Three microphones were arranged in an 'L' pattern, as shown in Figure 7. A sound originating from a given location on the ground produces a specific pair of ITDs between microphones 1 and 2 and microphones 2 and 3. In other words, for a given observed pair of ITDs, the location of the sound source can be estimated by comparing this pair of ITD measurements. The ITDs were inferred from the short-time cross-correlation between the signals from spatially separated microphones. These cross-correlation functions are also computed around 22 times per second, using the same data that are used for the FFT mentioned above.

After the pair of inter-microphone cross-correlation functions are computed, their outer product results in a 2D matrix of neural activities. Each element of the matrix corresponds to a certain point on the ground. The neuron with the maximum activity will thus signal the location of the sound source.

Synchronisation of the What and Where pathways

The system as a whole can be imagined as having two layers that we will call cortical and subcortical. In the subcortical area, the *What* features (the spectrotemporal representation) and the *Where* features (the ITDs) are continuously being computed. The cortex is normally silent. There are gates between the subcortical and cortical areas that are normally closed. Each gate corresponds to a different kind of sound (i.e. speech, claps, whistles).
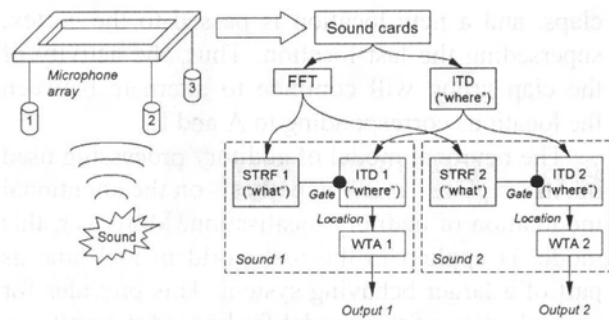

Fig. 7: Overview of the auditory processing scheme. Sound from the microphones is processed in separate '*what*' and '*where*' pathways. Information about '*what*' gates the flow of information about '*where*' to the final winner-take-all (WTA) operation and the output. See text for explanation.

Activity in the cortex is triggered by activity in the *What* pathway. When a given sound is detected, a corresponding gate is opened for a short time. While this gate is open, location activity is passed onto a corresponding cortical cell group where it is integrated and subjected to a winner-take-all operation. The winning activity codes for the most likely location of the sound that activated the *What* pathway.

Only certain *What* cells can open the cortical gates: those that respond to fast changes in the spectrum. There are two consequences of this. First, localisation performance is enhanced, because the transient parts of signals are the easiest to localise. Second, multiple sounds that are interleaved in time can nevertheless be separated.

To consider a specific example, imagine that one visitor is speaking while another is clapping. The transient elements of the speech repeatedly open the gate to the 'speech' cortical area, where the corresponding location maps are passed and held for a short time. Simultaneously, the location maps corresponding to the claps are passed to a different cortical area and held for a short time. In this way, the locations corresponding to two distinct and ongoing sound events are separated and simultaneously presented to higher processing centres.

In another example, two people are clapping at different locations. When person A claps, the clap gate opens briefly. The location map computed from the onset of the clap is passed into the cortex, where it is held for a short time. Next, person B

claps, and a new location is passed to the cortex, superseding the last location. Thus, the activity of the clap group will continue to alternate between the locations corresponding to A and B.

The neuronal model of auditory processing used in Ada explores a new hypothesis on the attentional modulation of auditory localisation. Moreover, this model is applied in the real-world in real-time as part of a larger behaving system. This provides for an evaluation of this model far beyond the artificial conditions in which audition is usually explored. In the Ada architecture the auditory system is an input sensorimotor process that provides a very significant abstraction from the input. From two sets of three microphones, sending 16-bit input into six sound card channels at 44 kHz, the auditory system can categorise and localise the sound - a data reduction of several orders of magnitude. The encapsulation of the complex operations involved is vital for simplifying the design of the control and behavioural modules at the next level of the system.

4.3. Behavioural processes

4.3.1. Overview

To build behaviours, Ada uses a set of behavioural processes to coordinate her sensorimotor processes. Ada has four of these basic behavioural processes:

1. **Track:** Isolate the trajectories of individual persons and groups. Information from the floor tile pressure sensors is used to determine the location, speed, direction and weight of persons. The limited resolution of the tiles means that it is not always possible to distinguish individual paths, so in some cases Ada only knows about the presence of groups of people at certain locations.

2. **Identify:** Test the individual responses of persons to cues and reward them if they are responsive. Once a person has been tracked successfully for a certain period of time, their *compliance* is tested. The test stimulus takes the form of a flashing tile next to a tracked person. Each time a person follows a compliance cue their measured compliance increases; once it

reaches a certain threshold they are given a reward. The reward has three components: a pulsating coloured pattern is shown on the floor around the person, light fingers are directed at the person, and an image, live video and recorded trajectory of the person are displayed on the BigScreen. The images on the screen automatically move around to align themselves with the current estimated direction of motion of the person.

3. **Group:** Try to actively influence the position and distribution of persons. Grouping cues of various colours and types are shown to try to bring many persons to a single location. Ada tries to learn to deploy cues that people are most likely to respond to by measuring their responses to different randomly chosen cues.

4. **Play:** Deploy an interactive game. There are four types of games that Ada can play:

   - *Football:* Visitors try to step on an animated bouncing ball on the floor tiles. The ball speeds up whenever it bounces off a person to make the game more difficult.

   - *Pong:* A variation on the old Atari game. Two teams each collectively control a paddle, whose position is determined by the centre of gravity of each team. The paddle is used to strike a bouncing ball towards the opposition's goal.

   - *Boogie:* A dance game for situations in which the space is very crowded. Pulsating patterns are shown on the floor, which change in frequency and colour as the power spectrum of the floor pressure data changes with people's dancing tempo. Individual highlighted tiles are connected to percussive sound effects.

   - *Gunfight:* People try to 'shoot' each other. Jumping from a tile to an adjacent tile generates a bullet in the direction of movement. Jumping on the spot creates a shield, making the person temporarily impervious to bullets but also disabling their ability to shoot. Persons who leave the space or are shot more than a certain number of times are eliminated from the game; the winner is the last one remaining.

Each behavioural process integrates a set of behaviours distributed across several servers, controlled by a *process* residing on one of the servers. They differ from the sensorimotor processes in that they do not access the device drivers directly, and they communicate with each other between servers.

There are four special servers containing behavioural processes that do not have corresponding sensorimotor processes: *object_server, game_server, cue* and *behavior_server. Object_server* is an object-based database that contains Ada's persistent representation of persons and groups of people. *Game_server* controls the running of the floor-based games that Ada can play. *Cue* contains methods for trying to influence the position of groups of people in the space by generating different patterns on the floor. *Behavior_server* coordinates the switching of behavioural processes to ensure an overall coherent effect.

What the visitor sees in Ada at any point in time is a mixture of the behavioural processes in different states of operation. These states of operation, called *behavioural modes*, regulate the processing and output of each behavioural process. There are six behavioural modes: *Sleep, Wake, Explore, Group, Play* and *End (Leave)*. Appendix 2 contains an overview of how the behavioural modes affect the functions performed by each behavioural process, and Section 4.4 describes the way in which behavioural modes are switched.

4.3.2. Neural adaptive action selection using DAC

As an example of how one of these behavioural processes works, we examine the *Group* process in more detail. It is based on a neural model of classical and operant conditioning called Distributed Adaptive Control (DAC) /53/. Given a set of appetitive and aversive unconditioned stimuli (US+/US-) and conditioned stimuli (CS), DAC selects unconditioned responses (UR) based on a set of predefined (US+/-, UR) reflex mappings. If a CS is consistently paired with a US, DAC will acquire a conditioned response (CR). In this way DAC directly implements the notion of stimulus substitution, which is a standard interpretation of the classical conditioning paradigm.

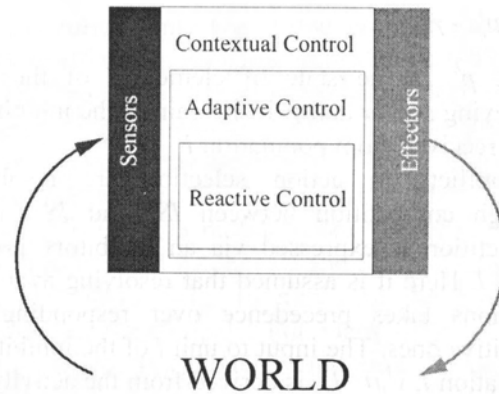Figure 8 shows an overview of the DAC architecture. DAC consists of three, tightly coupled,

Fig. 8: Overview of the DAC architecture. Explanation in the main text.

layers of behavioural control: reactive, adaptive, and contextual control. The reactive control layer provides a behaving system with a prewired repertoire of reflexes, which enables it to interact with its environment and accomplish simple automatic behaviours. The activation of any reflex, however, also provides cues for learning that are used by the adaptive control layer. Adaptive control provides the mechanisms for the adaptive classification of sensory events and the reshaping of responses supporting simple tasks, and can be seen as a model of classical conditioning. The sensory and motor representations formed at the level of adaptive control provide the inputs to the contextual control layer, which acquires, retains, and expresses sequential representations using systems for short-term and long-term memory. These representations are used to control ongoing behaviour in the context of behavioural plans /51,57/. For Ada, we restrict ourselves to using the reactive and adaptive control levels of DAC, called DAC0 and DAC2, respectively.

Reactive control (DAC0)

The reactive control structure (DAC0; Fig. 9) is implemented by four populations of simulated neurons that are interconnected by prewired fixed connections. The sensor readings of the appetitive unconditioned stimuli, $US+$, and the aversive unconditioned stimuli, $US-$, are projected onto two populations of internal state neurons, IS^+ and IS^-. The input, v^k_i, of cell i in IS population k is defined by:

$$v_i^k = p_i^k - \gamma^k I \qquad (1)$$

where p_i^k is the state of element i of the US conveying sensor and γ^k is the gain of the inhibitory input received from population I.

Conflicts in action selection are resolved through competition between IS^+ and IS^-. This competition is expressed via an inhibitory population I. Here it is assumed that resolving aversive situations takes precedence over responding to appetitive ones. The input to unit i of the inhibitory population I, $v_i^I(t+1)$, is derived from the activity of the aversive internal state population IS^-:

$$v_i^I(t+1) = \alpha^I v_i^I(t) + \gamma^I \frac{1}{M^{IS^-}} \sum_{j=1}^{M^{IS^-}} o_j^{IS^-}(t) \qquad (2)$$

The activity of population I is determined using Equation (2), thresholding with θ^I.

The activity of unit I in IS population k, o_i^k, is defined by thresholding the integrated input v_i^k:

$$o_i^k = H(v_i^k - \theta^k) \qquad (3)$$

where H is the Heaviside or step function and θ^k defines the activation threshold of the units of IS population k.

For both US-IS mappings one sensory element will activate only one neuron in IS^+ or IS^-, respectively. v of populations IS^- and IS^+ is subsequently thresholded in order to generate their activity. If any neuron in IS^- is suprathreshold it will depolarise the membrane potential of the inhibitory population I, v^I. If v^I is above threshold it will inhibit the activity in IS^+. In this way a prewired preference relationship is defined between appetitive and aversive internal states, according to the rationale that it is generally more important to avoid aversive states than to seek out appetitive states.

Both IS populations will depolarise specific neurons in population UR, which represents particular actions. If none of the neurons of the internal state populations are active, the reactive control structure will resort to its default action of *exploration*. The input, r_l, of unit i in the UR population is defined as:

$$r_l = \sum_{k=1}^{K} \sum_{i=1}^{M^k} y_{li}^k o_i^k \qquad (4)$$

where K denotes the number of IS populations, M^k is the size of IS population k, and y_{li}^k is the strength of the connection between cell i of IS population k and cell l of the UR population.

After updating their inputs the UR units compete in a winner-take-all fashion. The winning unit's activity is again thresholded, θ^{UR}. If its activity is above threshold it will induce a particular motor action, a conditioned or unconditioned response. If no motor unit is active the control structure will trigger a default exploration behaviour.
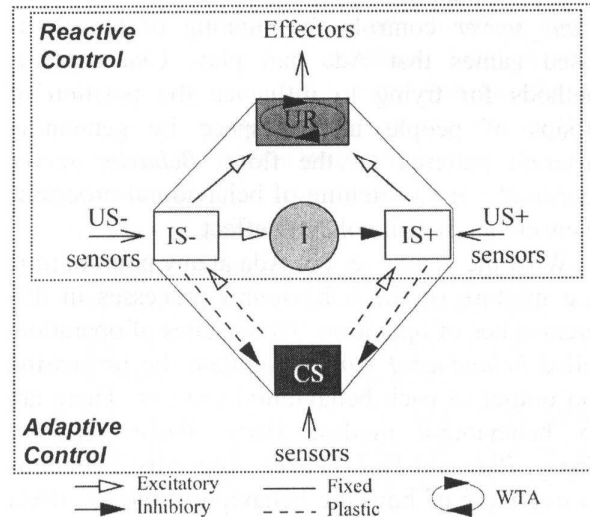

Fig. 9: DAC reactive and adaptive control layers. The components and their interactions are described in the text. Adapted from /51/.

Adaptive control (DAC2)

The adaptive control structure (DAC2) is defined on the basis of the reactive control structure (DAC0). It adds the components dealing with the processing of representations of CS events, and their association with URs. The population CS receives encoded events from initially neutral stimuli. The activity, u_j, of unit j in population CS is derived from the state, s_j, of element j of the input sensor:

$$u_j = e^{-\gamma^s s_j} \qquad (5)$$

where γ^s defines the slope of the transduction function.

CS in turn excites populations IS^+ and IS^-, in a modified form of equation (1). In this new form,

the input, v^k_i, of cell i in IS population k is defined by:

$$v^k_i = \sum_{j=1}^{N} w^k_{ij} u_j + p^k_i - \gamma^k I \qquad (1')$$

where N is the size of the CS population, w^k_{ij} is the efficacy of the connection between CS cell j and cell i of IS population k. p^k_i is the state of element i of the US conveying sensor and γ^k is the gain of the inhibitory input received from population I.

The synapses forming the connections between these populations are modifiable using a predictive Hebbian learning rule /53/. This method embeds a local learning rule in a recurrent circuit. After the inputs, v^k, of the IS populations are updated (Equation 1′), the IS populations in turn recurrently inhibit the CS population. The resultant activity, u'_j, of unit j in the CS population now is defined as:

$$u'_j = u_j - \gamma^r e_j \qquad (6)$$

where γ^r is a gain factor modulating the effect of the recurrent inhibition and e_j is the recurrent prediction defined by:

$$e_j = \sum_{k=1}^{K} \sum_{i=1}^{M^k} \frac{w^k_{ij} v^k_i}{M^k} \qquad (7)$$

where K denotes the number of IS populations, M^k is the size of IS population k, w^k_{ij} is the strength of the connection between cell i of IS population k and cell j of the CS population, and v^k_i is the integrated activity of unit i of IS population k. e is the *predicted CS* given the state of the IS populations and will be referred to as a CS *prototype*. The connections between unit j of population CS and unit I of IS population k now evolve according to:

$$\Delta w^k_{ij} = \eta^k v^k_i u'_j \qquad (8)$$

where η^k defines the learning rate of the connections between population CS and IS population k. Despite the possibility of u' attaining negative values, w is kept greater than or equal to 0 at all times. The representations of CS events constructed in this way will ultimately express the average CS state conditional on particular IS states. We have shown that this solution allows the use of local

learning rules, while preventing problems such as overgeneralization, primacy, and saturation /57/.

DAC has been investigated using formal approaches /52/ and robots /53,55-57/, and has been shown to be compatible with formal Bayesian models of human decision-making /51/. In one task, for instance, a mobile robot was required to associate coloured patches on the floor of an arena with actions in order to minimise the travelled distance between targets. For many different task configurations, the model would find the shortest route between targets in this robot equivalent of a random foraging task /55/. The DAC architecture has established itself as a standard in the field of artificial intelligence and behaviour-based robotics /3,11,13,25,40,54/. The principles investigated at the level of reactive and adaptive control have been translated into biophysically detailed models of key structures involved in classical conditioning, i.e. the experience dependent reshaping of receptive fields in the primary auditory cortex /28,45/ and the adaptive timing of responses by the cerebellum /26,29,54/

Allen Newell, one of the founders of Artificial Intelligence, defined general intelligence as the ability to make anything a task /38/. This implies that the cognitive architecture underlying intelligent behaviour must be capable of coping with a large range of possible task configurations. Any model of such an architecture must therefore be shown to generalise easily to different task domains. Ada allowed us to investigate this generalisation for aspects of the DAC architecture. Although DAC was originally developed as a model of classical and operant conditioning applied to mobile robots, it can be used in more general learning situations. Here, we investigate the application of DAC to the learning of cues for guiding visitors to a particular location. In both cases, however, the DAC architecture · is employed as a model of adaptive action selection as first studied by Thorndike over 100 years ago using the puzzle box /48/. The generic learning problem consists of two elements. The first element is stimulus identification, i.e. deciding which event in the world is behaviourally relevant. The second element is adaptive action selection - the process of deciding which action allows the system to achieve its goals.

To learn how to guide visitors, Ada must choose a cue, show it to a visitor, and evaluate the visitor's subsequent action. If the visitor follows the cue and moves towards the goal location then it was effective; if the visitor moves away then it was not. This provides a mapping of visitor movement towards the goal position as appetitive ($US+$) and movement away as aversive ($US-$). Ada can choose from four different colours of cue (red, green, blue, white) and two types of cue (single flashing tile, travelling 'bullet' towards goal), giving a library of eight different cues. Thus there are eight different types of $US+/UR$ and eight corresponding $US-/UR$ stimuli - one for each cue in the library. For the results presented here, the goal was placed in a fixed location that was usually avoided by visitors, but in principle it could be in dynamically changing positions.

In principle, the initially neutral CS could be composed of many multi-modal components combining tactile, audio and visual information. However, for simplicity we chose to set the CS to be the general level of 'crowdedness' of the space, with four neurons coding increasing levels of visitor crowdedness in the space. The rationale behind this choice was twofold. Firstly, the CS should ideally be provided by passive visitor interaction - the visitor should not have to perform 'special' actions to provide Ada with the necessary information to provide cues. Secondly, we expected that the visitor density could affect the likelihood that they would be able to see and react to different cues (e.g. due to occlusion by other visitors in crowded situations), hence making different cues more appropriate for different visitor density scenarios.

Figure 10 illustrates the operation of the *Group* process. Topographic maps of visitor locations are continually fed into the *Ranger* module, and the motion of the most 'salient' visitor (as determined by the *Identify* agent) is tracked. An error signal calculated by the *Ranger* module, indicating motion towards or away from the goal position, is fed into the *Cue* module. Once activated by *behaviour_server*, the *Cue* module converts and gates the reinforcement signal into *DAC* as $US+$ or $US-$ according to the current UR (i.e. the last

selected cue). At the same time, the visitor density coding from *floor_server* is gated into *DAC* as the *CS*.
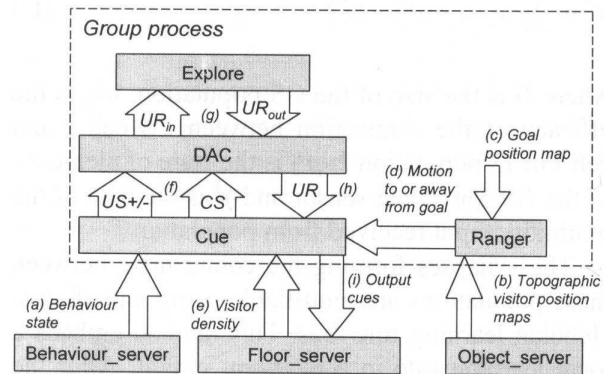

Fig. 10: Overview of neural pathways within the group process. The process is activated when signalled by the behaviour_server (**a**). The topographic location of the most salient visitor (**b**) is compared with the goal position (**c**) and a motion error signal is generated (**d**). The motion error signal and the current visitor density (**e**) form the US+/- and CS signals, respectively (**f**). The UR coming from DAC (**h**) can then be output to produce the selected cue (**i**). If no US or strong enough CS is available, a default explore behaviour is triggered (**g**) which maps a random action onto the output.

Within *DAC*, the association of *US* and *CS* takes place. The resulting *UR* (or *CR*, if caused by a *CS*) will either be a forced action or a default explore action. If an explore action ($UR = 0$) occurs, the *Explore* cell groups are activated to provide a randomly chosen *UR*, which is then mapped back into *DAC* to provide a 'real' action. Finally, the resulting *UR* passes through *Cue* into *floor_server*, where the cues seen by the visitors are generated.

The predefined reflexes for the learning scheme are very simple. Two basic rules apply that define the essence of trial and error learning: if something works, do it again; otherwise try something else at random. This basic learning process provides for the latching of cues that were successful. In addition, these cues will be associated with the *CS*, allowing DAC to identify those cues that are effective and ineffective given the crowdedness of the space. DAC uses a weight matrix to initialise these rules.

4.4. Behavioural modulation and emotional model

Ada has to solve the general problem of behavioural control and action selection. Behaviours and actions can be called adaptive when they allow a behaving system to achieve particular goals, despite being in an initially unknown environment. In the earlier example of the DAC architecture the goals are defined in relation to particular tasks. In the case of Ada these goals serve a maximal and consistent interaction with visitors. Conceptually, Ada is an artificial organism that tends to homeostasis, along similar lines to that described by Maslow, by trying to maximise her own goal functions, which we interpret as her 'happiness'.

In this respect we assume that the homeostatic control of goal functions feeds into a hedonistic evaluation of the state of operation of the system /46/. This means that the system as a whole must implicitly or explicitly compute its level of happiness, which can then be used to determine whether certain actions contribute to this goal. As a first approximation we can write:

$$H = f(g_s, g_r, g_i)$$

where H = overall goal or 'happiness';
g_s = survival; g_r = recognition; g_i = interaction.

Survival is a measure of how well Ada satisfies her basic requirements, which are to maintain a certain flow of visitors over time and to keep these people moving with a certain average speed. *Recognition* quantifies how well Ada has been able to track and collect data about people, as a precondition for more advanced interactions. This process can be seen as Ada 'carving' objects out of the world of her sensory data, which is implemented as a progressive filtering of the sensory data and the creation of objects in her internal database once certain criteria of persistence and coherence have been satisfied. *Interaction* measures the number of successful human interactions in which Ada has been involved, with more complex interactions such as games being weighted more highly.

As a system, Ada has the goal of maximising the value of H. There are multiple strategies for achieving this: for example, Ada could encourage high visitor throughput, but in doing so have very few possibilities for recognition and interaction (g_s high, g_r and g_i low). Alternatively, Ada could also achieve an equivalent value of H with only a few visitors in the space, but with high recognition and interaction with each visitor (g_s low, g_r and g_i high). The actual computation of H occurs in multiple ways: an explicit top-level calculation is performed using simulated neurons, and in parallel individual behavioural processes calculate their own contributions to the components of H.
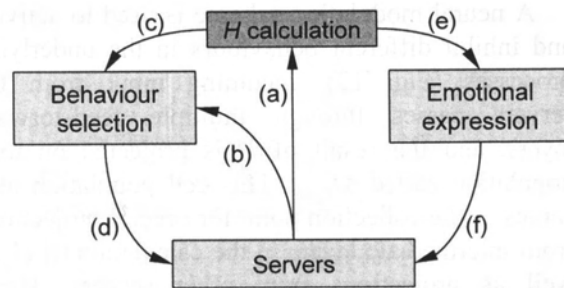

Fig. 11: Overview of neural pathways at the behavioural modulation level in Ada. Signals from the servers feed into the calculation of the homeostatic variable H (**a**). Behavioural mode selection (**d**) is determined by the resulting value of H (**c**) and other events from the servers (**b**), as well as internal dynamics within the behaviour selection module itself. Emotional expression parameters are also calculated from the value of H (**e**) and feed back to output elements of the servers (**f**).

The value of the homeostatic variable H and its underlying goal functions is used in two ways: to select behavioural modes, and to define emotions (Fig. 11). This explicit link between behaviour and emotion via H has the advantage of automatically providing a degree of coherence between the actions and expressed emotions of the organism.

The following two sections describe the operation of the behaviour selection system and the emotional expression model in more detail. Sample data from the operation of the system are shown in Section 5.2.

4.4.1. Behavioural selection

The results of the H calculation are combined with other high-level inputs and the state history to select the most suitable overall set of behaviours for Ada - her current *behavioural mode*. As

mentioned in Section 4.3, Ada was given six basic behavioural modes: *Sleep, Wake, Explore, Group, Game* and *Leave*. Strictly speaking, behaviour selection occurs at multiple levels - for example, the floor tiles display colours that depend on the local effects in use, as well as the overall state of the space. However, we will mainly concern ourselves with the determination of how the behavioural mode affects the operation of the processes at the next level down in the architectural hierarchy.

A neural modulation scheme is used to activate and inhibit different behaviours in the underlying processes (Fig. 12). Incoming input from the servers passes through multiple feed-forward layers, and the result of H is projected on to a population called M_{input}. This cell population also serves as the collection point for several projections from intermediate layers in the calculation of H, as well as projections from other servers. Time-delayed components in M_{input} can also be introduced by using the M_{timer} population as a delay line.

At this point, all non-linear terms in the behaviour selection have essentially been calculated and accumulated at M_{input}. From here, M_{input} drives two cell populations equally: $M_{competition}$ and M_{select}. Excitatory and inhibitory interactions between these two populations lead to a biased competition between the neuronal representation of behavioural modes. A small amount of recurrent excitation in M_{select} forces either a 'hard' or 'soft' winner-take-all (WTA) on the population. The resulting active cell(s) in M_{select} then define the current behavioural state of Ada. Each cell in M_{select} corresponds to a different operating configuration of all of the underlying processes, i.e. the set of possible behaviours that the lower-level processes can deploy. The extent to which the WTA operation needs to be 'hard' depends on the subjective evaluation of how the behaviours interact and/or interfere with each other.

In practice, during a live exhibit with a high visitor flow rate, the behavioural control was run in a 'hard' mode with an underlying cycle to ensure that all visitors could see the space in a short time. Some other modes existed for test modes and gazer calibration. The behavioural modes were used to switch the behavioural modules into different

operating states, so that the overall effect would provide an entertaining visitor experience. Table 3 summarises how each behavioural module was modulated by the different behavioural modes.
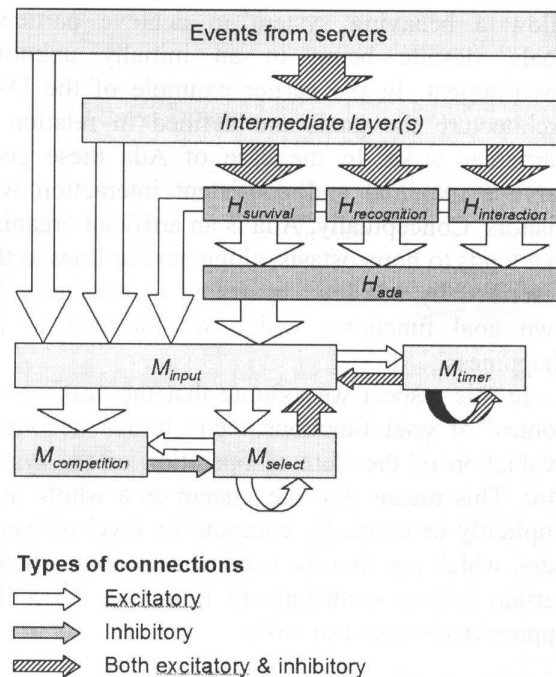

Types of connections
⟹ Excitatory
⟹ Inhibitory
⟹ Both excitatory & inhibitory

Fig. 12: Neural behavioural mode selection scheme used in Ada. The M_{input} population receives input from the various servers, as well as preprocessed data from the intermediate layers and the current values of the components of H. These data are combined and projected on to the $M_{competition}$ and M_{select} populations, with the winning neuron(s) in the M_{select} population encoding the current behavioural state(s). Recurrent connections to M_{input} from both M_{select} and M_{timer} allow various combinations of state transitions with different timings.

4.4.2. Emotional model

Ada has two components to her emotional expression: her *mood* and her *emotions*. Both depend directly on the current levels of her goal functions (H), as depicted in Figure 13. To distinguish between Ada's moods and emotions we adopted a model of affect in which mood changes on a time scale of hours or days, and emotions change on a time scale of seconds to minutes /18/. Exhibition-related requirements meant that the time course of Ada's behaviour needed to be rather

TABLE 3

Modulation of activity of behavioural modules depending on behavioural states

Mode	Track	Identify	Group	Play
Sleep	Simple reactive patterns only; one colour for all visitors	Off	Off	Off
Wake	Visitors given different-coloured floor tiles	Off	Off	Off
Explore	Visitors given different-coloured floor tiles	Probe for 'interesting' visitors; deploy light fingers and gazers	Off	Off
Group	Visitors given different-coloured floor tiles	Probe for 'interesting' visitors; deploy light fingers and gazers	Try to direct visitors to a certain location in space	Off
Play	Running in background to support games	Off	Off	Play selected game depending on number of visitors in space
Leave	Tile effects show path to exit of space for each visitor	Off	Off	Off

compressed in comparison with humans, so we chose to set mood activity to change on the order of tens of seconds to minutes, whereas emotions were set to change in the order of seconds.

Ada's mood system has the parameters of *arousal* and *valence*, a bipolar description of affective space common in the literature /43/. The arousal parameter is set by the current behaviour mode, and corresponds to the general level of activity of the system. The valence parameter represents the status of *H*. Emotions are synthesised on input from the three components of *H*. *Joy* is set by the goals of *Survival* and *Interaction* in the sense that *Joy* is high if *Survival* or *Interaction* approach maximum achievement. The *Sadness* parameter increases if either *Recognition* or *Interaction* decreases from maximum achievement. *Anger* is excited if *Survival* decreases from maximum achievement, while *Surprise* is triggered by a sudden increase in *Recognition*.
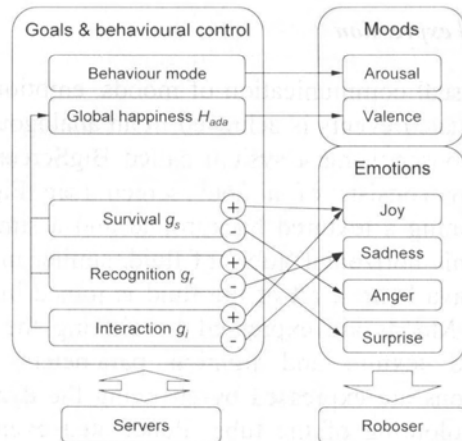

Fig. 13: Ada's mood and emotion synthesis. The current behaviour mode set *Arousal*, whereas global Happiness set *Valence*. The emotions *Joy*, *Sadness*, *Anger* and *Surprise* were set by the status of achievement of the high-level goals *Survival*, *Recognition* and *Interaction*, as indicated by the arrows. (+), approaching maximum goal achievement' (−), moving away from maximum goal achievement. Adapted from /61/.

Audio expression

Audio expression is achieved with the *Roboser* system, a real-time music composition and performance system that accepts input from a variety of sources to guide a composition process. The Roboser composition engine synthesises a stream of MIDI data upon simulated neural input /60/. Roboser composes music on up to twelve performance tracks in real-time. During performance in Ada, the outputs of the Roboser tracks are performed using a sampler, resulting in a complex soundscape. Different MIDI parameters on different tracks are set in real-time to achieve the desired musical effect for each mood and emotional state, according to an extended version of a scheme outlined by Gabrielsson and Juslin /21/. On top of the mood/emotion soundscape is a set of localised sound effects corresponding to the actions of the space. Typically, these sounds are punctuate, emotion-independent audio events. A second Roboser system running in parallel with the emotional soundscape handles these sounds, with a matrix of speakers in the space providing for the localisation of selected sounds.

Visual expression

Visual communication of moods, emotions and punctuated events is achieved in an analogous way to Roboser using a system called BigScreen. The display consists of a 360° screen (see Fig. 14) containing a textured background and a simulated dynamic horizontal 'tube' of fluid, similar in effect to a lava lamp if all of the fluid is joined into one blob. Moods are expressed by altering the background texture and lighting parameters, while emotions are expressed by changing the dynamics and colouring of the tube. Punctuated events are represented by insertions of energy at certain points along the tube, causing rippling waves to appear. Events related to individual visitor interactions can cause the appearance of live gazer video windows or saved snapshots that can move around the screen.

4.5. Computational infrastructure

Ada runs on a 100 Mbit network of 31 PCs (AMD Athlon XP 1800+, 1.0 Gb RAM, Tyan motherboards, SuSE Linux 7.3). Driver cards are used for DMX and Interbus communications. In addition, 40 frame grabbers (Hauppauge, Hauppauge, NY, USA), four sound cards (M-Audio, Arcadia, CA, USA) and 11 dual-headed 3D accelerated graphics cards (Matrox, Dorval, Canada) are installed. Laptops on a wireless LAN (Cisco, San Jose, CA, USA) enable system testing and tuning to occur while walking around in the main space.

4.6. Data logging

Data can be logged at several different parts of the system simultaneously at ~5 Gb per hour, not including digital video tape (DV) or raw digital sound (WAV) data. The data are automatically transferred to a central repository on the network each evening for backup on to DVD-R. A time-server keeps all timestamps across the cluster synchronised to within 100 ms, which is sufficient for most types of analysis. A brief description of the types of data logged is given in Table 4. Every level of the system is covered by some form of computerised data recording, as well as several external hand-recorded statistics concerning system reliability and visitor flow.

5. RESULTS

5.1. Ada in operation

5.1.1. System reliability and public response

Most current models of neural computation need to work only in relatively well-controlled laboratory conditions, for periods of time much shorter than the lifespan of real organisms. For Ada, however, every major component was contractually required to work safely and reliably with large numbers of naïve users, for up to 12 hours per day, over a period of 5 months. The visitors also expected to enjoy themselves while passing through the exhibit. This demanded a much more rigorous approach to operational reliability than is normally required of a research laboratory. From 1998 to 2002, ten increasingly large public tests were run to evaluate the feasibility and scalability

TABLE 4

Description, level and types of data logged in Ada - All different levels of the system architecture are covered

Description	Level	Type
Floor raw load data: sensor values and RGB neon output	1: Device driver	Text
Floor server data: positions of all currently loaded floor tiles	2: Sensorimotor	Text
Tracking data: onset, path and endpoint of visitor tracks on floor	3: Behavioural module	Text
Neural network internal states for behaviour selection and emotion (it was possible to track all neuronal states at any level)	4: Behavioural modulation	Text
Camera view (up to 4 simultaneous gazers or overhead cameras)	0: Hardware device	DV
Gazer view of tracked visitors	1: Device driver	MPEG
Ada generated musical composition and sonic events	1: Device driver	MIDI
Sound in space	External behaviour	WAV
Miscellaneous hand-held video footage	External behaviour	DV
Visitor questionnaires gauging responses to the Ada experience	External behaviour	Text
Visitor guest book	External behaviour	Text
Operator log book for tracking system reliability issues	External behaviour	Text
Hand-counted visitor flow data (15 min intervals)	External behaviour	Text

of the underlying technologies, stability and performance of the different models, gauge visitor impressions, and test different interaction scenarios. The two key issues that stood out from the results of the tests were the need for effective visitor flow control, and the importance of communicating Ada's intentions clearly through the use of effective cues and visitor pre-conditioning sequences. One direct consequence of this experience was the decision to employ guides to actively inform Ada's visitors as much as possible about what they would see in the exhibit.

The system reliability data were tracked for the entire duration of Expo.02, using logs kept by the system operators and exhibit guides. Anecdotal evidence of visitor enjoyment was recorded in the visitor guest book. In addition, over 800 visitors were asked to fill out a questionnaire measuring their interpretation and evaluation of Ada. These visitors were selected under a variety of controlled manipulations of Ada's operation, e.g. disabling particular modalities or manipulating visitor density, and visitor behaviour.

Ada ran for over 1700 hours on 159 consecutive days with an uptime of better than 98.3%, where uptime was defined as having a system that functioned well enough to enable a normal flow of visitors through the exhibit. Discounting outages due to deficiencies in building services that were beyond the control of the project team, the overall system uptime was over 99.1%. This result compared favourably with other exhibits at Expo.02 which did not have the same level of technical complexity as Ada. Nine stable versions of the Ada software were released during Expo.02, incorporating incremental improvements in user functionality and data logging facilities. On any given day, either the latest development version or the stable version could be run, depending on the demands for testing and experimentation. Overall, the satisfactory operational result was partly the result of accumulated experience during the developmental process.

The public reaction to the exhibit was overwhelmingly positive, despite queue waiting times of up to 90 minutes. Surveys conducted by the Swiss Expo.02 organisation indicated that Ada was

one of the five most popular attractions out of over 60 at the Expo. An online poll /27/ also found that Ada was the most popular of the IT-related exhibits at the Expo. Anecdotal evidence from the visitor guest book also indicated a mainly positive reaction. The visitor queue length of >30 minutes outside the exhibit for the entire duration of the Expo suggested that the final attendance of 553,700 could have been higher if the capacity of the space had been larger.

5.1.2. Visitor flow control

Due to the extremely large number of visitors who wanted to see Ada, it was necessary to control their flow very rigidly to avoid problems with over-crowding. This was necessary both for safety reasons and to ensure that each visitor had a certain minimum amount of space with which to interact with Ada. Table 5 summarises a typical visitor experience in the space and Figure 14 shows a typical scene in the main space.

During normal operation, the main space received about 25 visitors at a time, giving a nominal throughput of about 300 visitors per hour and an instantaneous exhibit occupancy of 125 visitors. The tracking system worked as reliably for wheelchairs and children weighing more than about 20 kg as it did for adults.

5.2. Behavioural control

All actions of Ada and her allocation of resources and the processing of subsequent information are modulated by her behavioural modes, as
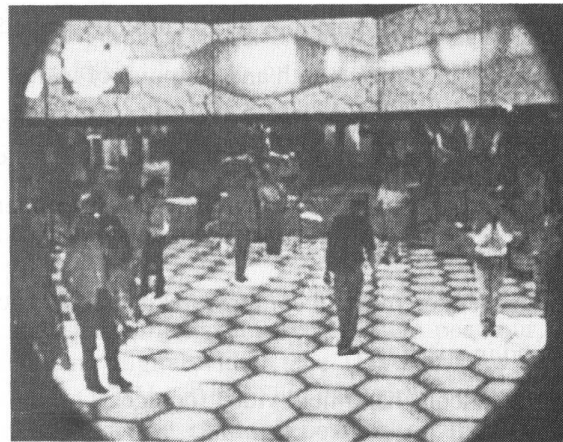

Fig. 14: A typical live user interaction scene within Ada, as seen through one of Ada's gazers. Visible are floor tiles, a visitor being highlighted by a light finger (centre left), a dynamic 3D visualisation on the BigScreen (top) and a live gazer video on the screens (top left). The visitor at the centre right has just clapped her hands and Ada has produced a flower pattern on the floor tiles below her feet in response. The visitor in the upper middle part of the image is being challenged by one of Ada's probes, a flashing white tile to her immediate left.

TABLE 5

Typical visitor experience at the Ada exhibit

Region	Visitor experience	Time (min)
Queue (outside)	'Brainworkers' educational video on big screen (10 min)	0-90
Conditioning tunnel	Sequential introduction to individual sensors and effectors	5
Voyeur corridor	View group of visitors interacting with Ada, listen to guide's explanation	5
Main space	Interact with Ada	4-6
Brainarium	View 'observation room' screens and look back into main space	5
Explanatorium	Art by H.R. Giger depicting the merging of biology and technology, guest book, videos with statements and explanations by non-scientists and scientists, credits poster	5

described in Section 4.4. The modes are set by a number of goal functions defined in a neuronal control model. A first question is whether this control model provides a robust control system that can handle the many unexpected events triggered by the visitors. Data were logged at all levels of the system for up to 90 minutes in length, and analysed for consistency and reliability.

This core module of Ada operated without a single failure during the entire Expo. We analysed the dynamics of this control module in relation to several internal and external measures.

Figure 15 shows a selection of data that were logged during 1 hour of normal operation that give a global description of Ada's control system and interactions. From this small selection it is already
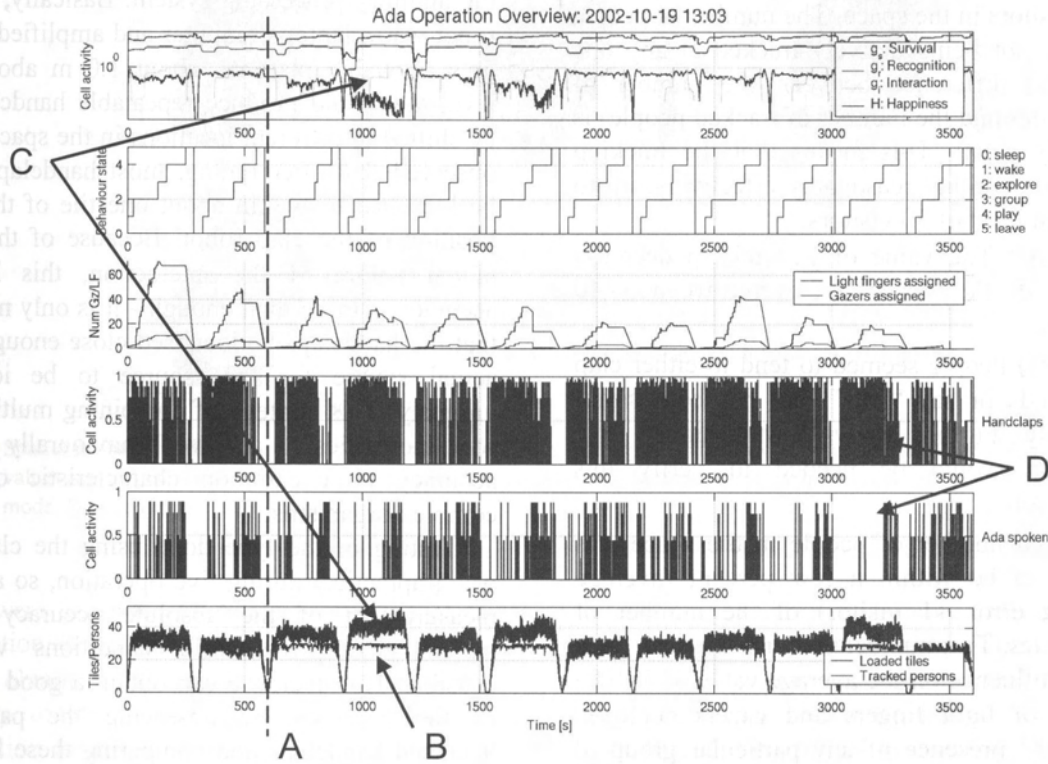
Fig. 15: Multi-level overview of 1 hour of live operation of Ada. The labels A to D are discussed in the main text. Selections from several different levels of the system architecture are shown. From top to bottom, the traces are:

- Levels of g and overall happiness H (note: log scale).
- Behaviour mode over time. The characteristic repeating staircase shows Ada cycling through the different modes; the length of each mode changes depending on visitor interactions. Each behaviour cycle lasts an average of about 5 min.
- Numbers of light fingers and gazers deployed during each cycle. Note that these values show the total number of light fingers and gazers deployed over the course of the cycle, not the actual number active at each time. When the space enters explore mode, compliant visitors have light fingers and gazers assigned to them. This continues through group mode, and ends at the commencement of game mode. The values are reset at the end of each cycle.
- Cell activity for a simulated neuron indicating detection of a handclap in the space. There is some variation in the handclap rate between individual cycles, and 'bursts' can be seen in the trace. This would support the observation that visitors sometimes tended to copy each other; i.e. if one person clapped their hands then it was quite likely that many others would follow suit. People who clapped their hands were given a reward of the floor tiles around them flashing for a short time; this probably induced them to continue clapping as well as encouraging others to try the same thing.
- Cell activity for a simulated neuron indicating detection of the spoken word "Ada". These events generally occurred much less frequently than handclaps.
- The number of loaded tiles (visitor detected) and the number of visitors that are actively being tracked. There was a limit built into the tracking system that would allow a maximum of 30 visitors to be tracked simultaneously.

possible to notice some features of the system as a whole. For example:

- (Fig. 15A) Ada tends to have maximal values of H when there are relatively few people in the space. This occurs during *Sleep* mode when a group is leaving, and another group is in the process of entering.

- (Fig. 15B) There is a limit of 30 people that can be tracked simultaneously, even if there are more visitors in the space. The number of people who are not being actively tracked at any one time (the difference between the number of loaded tiles and the number of tracked people) is relatively small. This means that the tracking system is working reasonably effectively, up to its inbuilt limit of 30 visitors.

- (Fig. 15C) The value of g_s tends to decrease sharply if the space is overcrowded (>40 visitors).

- (Fig. 15D) People seemed to tend to either clap their hands or say "Ada", but not both simultaneously. Further analysis with higher time resolution would be needed to verify this observation.

- The actual number of people in the space was verified to be within a few percent (average counting error <1 visitor) of the number of loaded tiles. This number does not seem to have a large influence on the average value of H. The number of light fingers and gazers deployed during the presence of any particular group of visitors does not affect the average value of H.

The dynamics of these traces, such as the variability of g and H, and the varying lengths of the behaviour modes, show that Ada is not an automaton. Rather, she is an open system flexibly interacting with her dynamic world. The verification of Ada's 'reasonable' operation, as shown above, is the first step on the way to a more in-depth analysis of individual components presented in the following sections.

5.3. Auditory processing

In certain behavioural modes (*Wake*, *Explore* and *Group*), floor occupancy data are combined with the localisation data to produce a floor effect as a reward for visitors whose handclaps are successfully detected. The reward, in the form of a 'flower' of light around the visitor's feet, is only shown if the handclap is localised to within a certain distance of a visitor. This functionality requires topographic maps of the floor and the handclap localisation to be overlaid as accurately as possible. During the system tuning phase before the start of the Expo, we used a 'clap robot' to calibrate the auditory processing system. Basically, the clap robot was a laptop computer and amplified speaker on a movable platform about 1.0 m above floor level that could produce repeatable handclaps and be shifted to different locations in the space. It was observed that after tuning, most handclaps would be localised to within about one tile of the actual position of the clap robot. Because of the multimodal nature of the interaction, this level of accuracy is more than enough - it is only necessary that the handclaps be localised close enough to the actual source for that source to be identified uniquely. This strategy of combining multiple sensory modalities to achieve behaviourally relevant accuracy is a common characteristic of many complex organisms.

Testing of the type done using the clap robot was impractical during live operation, so a precise measurement of the absolute accuracy of the system under real-world conditions was not possible. However, we can obtain a good estimate of the accuracy by observing the pattern of localised handclaps and comparing these locations with the actual positions of visitors from the floor data. Figure 16 shows the spatial probability distributions for detected handclaps, segmented by behaviour mode and averaged over 60 cycles (nearly 5 hours). The plot showing the average for all behaviour modes (lower left panel) shows two peaks, roughly corresponding to the locations of the two microphone arrays. This result seems sensible, as claps originating closer to the microphones will generally have a higher intensity and thus a higher chance of being detected. Due to the extremely challenging acoustic environment with a high level of reverberations and ambient noise, not all handclaps are localised to valid spatial positions where tiles are positioned. About 13% of all localised handclaps corresponded to invalid locations,
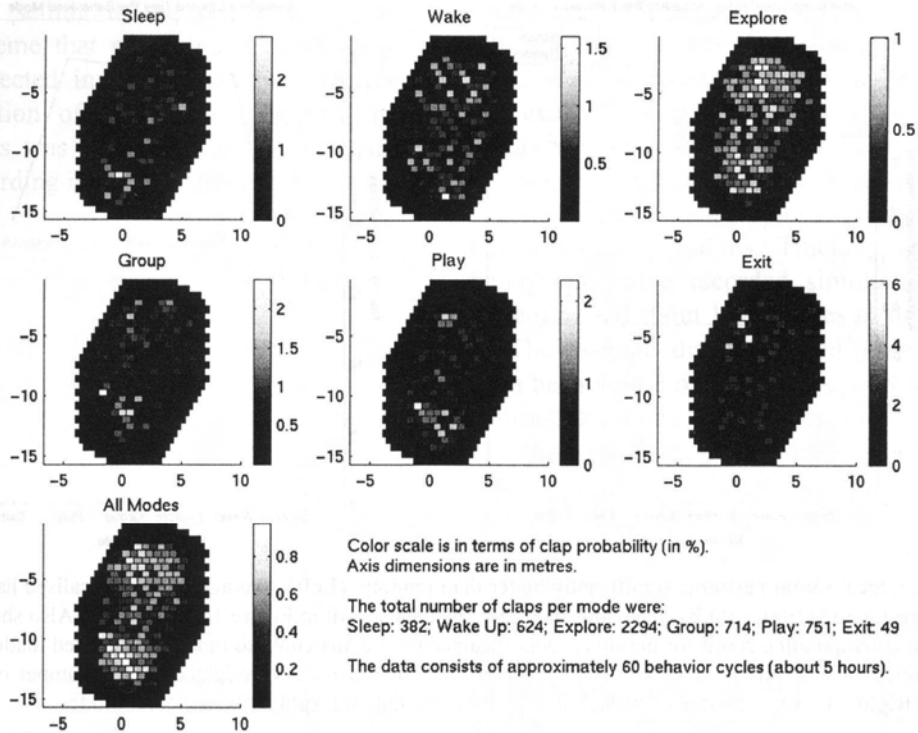
Fig. 16: Spatial probability distributions for detected handclaps in Ada, split by behaviour mode and averaged over 60 cycles. All values are in percent. The overall distribution plot is weighted by the number of handclaps occurring in each behaviour mode. Detailed explanation in the text.

which were discarded in Ada's processing of this information and in the subsequent analyses.

To obtain an estimate of the localisation accuracy of the microphones, we compare the localisation achieved by our attentional stereausis model to a hypothetical random scheme based on a Monte Carlo sampling technique (n = 100). This can be achieved by determining the distance from each localised point to the nearest occupied tile, and comparing it with a random sound localisation distribution. Figure 17 shows the average distance from each localised handclap to the nearest loaded tile, both for the actual handclap data and for random localisation data. For the random case, it is expected that the average distance decreases as the space becomes more crowded, with an overall average of about 1.15 m (the distance between tile centres is 0.66 m). For the real data, the average error across all behaviour modes is about 0.72 m, a significant improvement. It can be seen that for the random localisation case, there is an inverse

relationship between the number of loaded tiles and the average localisation error. This relationship is effectively removed by the use of the real localisation data, indicating that the accuracy of the matching is not strongly affected by the number of visitors in the space. The slightly higher average error for the *Sleep* mode is probably due to the visitors tending to be near the entrance or the exit of the space, which is where the microphones are least effective at localising visitors.

The rate of handclap detection can be seen in Figure 18. Visitors' handclaps were detected at about 0.5 Hz over the entire recording period (weighted by behaviour mode). However, during *Wake*, *Explore* and *Group* modes the detection rate increased to over 0.7 Hz, while it was below 0.35 Hz for the other modes. This indicates a large influence of the space on this aspect of the visitors' behaviour. In fact, the behavioural effect was probably significantly larger - it was observed that visitors' combined clapping patterns, once initiated,
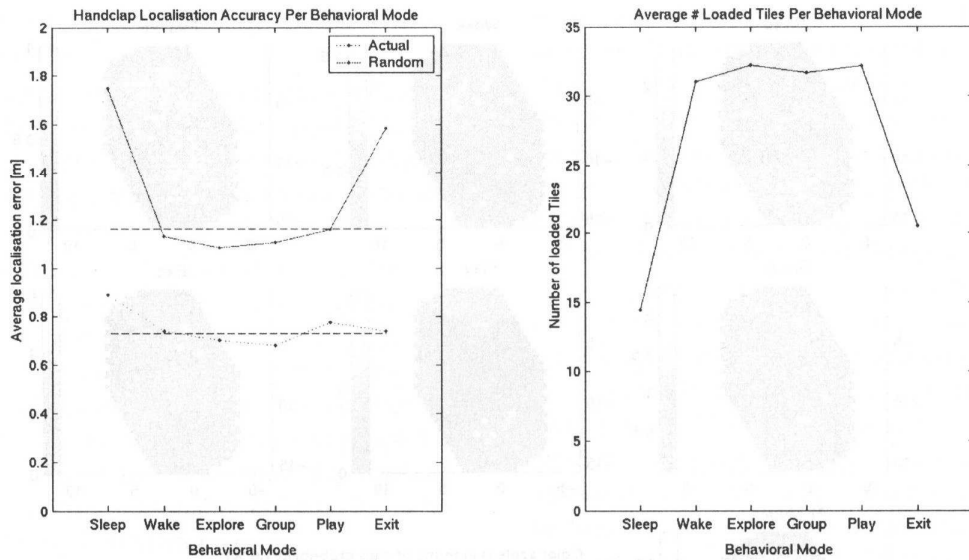
Fig. 17: Handclap localisation performs significantly better than random. **(Left)** Average error of localised handclaps (distance to the nearest loaded tile), split by behaviour mode for the data shown in Figure 16 (solid line). Also shown is a comparison with the corresponding result for purely random localisation, i.e. matching to randomly selected loaded tiles (dotted line). The dashed lines show the average distance for all behavioural modes, weighted by the number of handclaps in each mode. **(Right)** Average number of loaded tiles for the same data set, split by behavioural mode.

could reach a rate of several Hz. The handclap detection system saturates at high handclap rates and is not able to detect handclaps above a certain rate, but it was still able to produce correct output for a subset of the visitors. It was observed that visitors who clapped their hands and received feedback from the floor tended to continue clapping, a clear indication that they understood the correlation between their action and the response of the system.

Figure 19 shows the cumulative probability of a handclap localisation occurring within a certain distance of a loaded tile. The two curves compare the actual handclap localisation data with the data that would occur for a purely random localisation distribution. It is clear that the handclap localisation is working significantly better than random - 18% of handclaps fall exactly on a loaded tile, compared with 9% for the random case. This improves to about 90% of all handclaps localised within 1.32 m (two tile diameters) of a loaded tile, compared with only 64% for the random case.
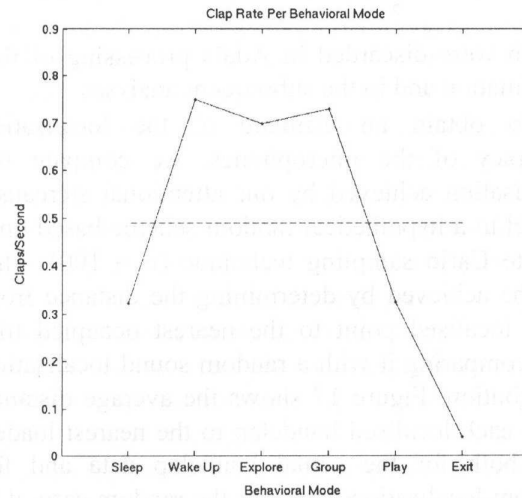

Fig. 18: Behaviour mode affects detected handclap rate. Average claps per second per behavioural mode (solid line) and for all modes (dashed line).

It can be seen in Figure 16 that no handclaps were recorded on the left-hand side of the space (minimum X-coordinate). This was due to an in-

correct threshold setting in the simulated neural transmission scheme that prevented the recording of handclaps detected in that region but did not affect the operation of the rest of the system. However, no bias was introduced to the data, as only the data recording itself was affected.
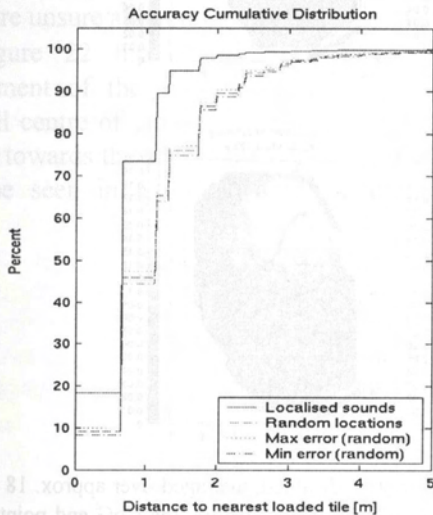

Fig. 19: Cumulative probability distribution of handclap localisation occurring within a certain distance of a loaded tile, compared with the same measure for a purely random handclap localisation scheme. The max/min errors for the random scheme indicate the ± 2 standard deviation limits for the Monte Carlo simulation (n = 100).

5.4. Neural adaptive action selection

Ada faces a number of unpredictable elements in her environment that require learning. One of these is that it is not known *a priori* what cues Ada should generate on the floor so that visitors will respond appropriately. We have rephrased this problem in the context of classical and operant conditioning, and adapted the reactive and adaptive control layers of the DAC architecture to this learning situation, as described in Section 4.4.

Groups of approximately 30 visitors were exposed to the normal exhibit behaviour cycle of about 5 minutes total duration. During this time, the space was set in *Group* mode for about 30 seconds and all tracked visitors were exposed to cues generated by DAC. These cues were all directed to one corner of the space, which previous experience had shown that visitors were very unlikely to visit

(on average) during a typical stay in the space. For one group of visitors the learning mechanisms of DAC were disabled, while for a second group it was enabled. After each trial with a visitor group in the learning-enabled condition, the synapse weights evolved by the DAC module were recorded. Every trial started with zero synapse weights between the CS and IS populations. Tracking and tile occupancy data were recorded simultaneously. Trials were repeated about 15-20 times in succession.

The average distribution of floor load during each behavioural mode for 18 cycles shows that the visitors distribute themselves in a highly mode-specific way (Fig. 20). The contours represent the fraction of the total duration of the behaviour mode that each location in space was occupied. During *Sleep*, new visitors enter the space from the lower left corner while the previous group exits (upper left corner). In the two following modes, *Wake* and *Explore*, we observe that the visitors tend towards an increasingly uniform distribution in space, suggesting less grouping and an increase of locomotion. In *Group* mode, however, visitors reduce their speed of movement and cluster more, before changing back to a more uniform distribution during *Game* mode. In the last mode, *End*, visitors accumulate at the exit of the Ada space. Throughout the entire cycle, visitors have a tendency to stay away from the entry area to the space. In general we see that the behaviours that Ada generates during the different modes induces marked differences in the distribution of visitors and their locomotion patterns.

When we compare the visitor distribution for the conditions in which the DAC learning system is enabled or disabled, we observe a change in the distribution during *Group* mode. In the control condition visitors generally do not visit the lower right corner of the space. The goal location of the *Group* process was placed in this region, the rationale being that an effective cue should draw people towards that point. In fact, the centre of gravity of the visitor distribution does shift towards the lower right corner when the DAC module is enabled. Figure 21 shows a comparison between the two cases by plotting the difference between the visitor distributions for test runs in which the DAC learning system of the *Group* process was switched
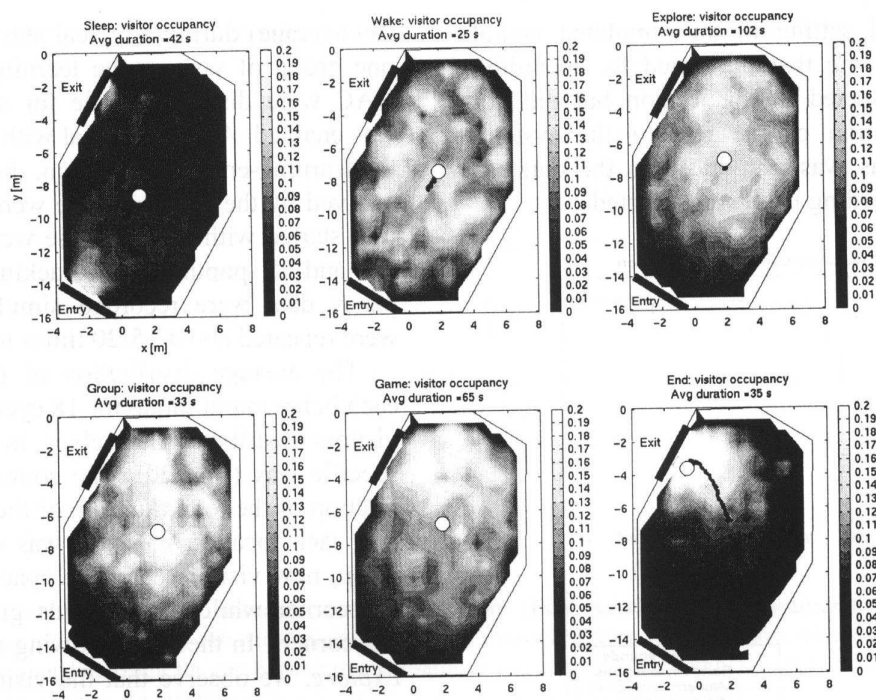
Fig. 20: Floor load distribution contour for a visitor cycle with the cue learning system disabled, averaged over approx. 18 cycles. The black trace on each plot is the movement of the centre of gravity (CoG) of the visitors; the CoG end point of the mode is indicated by a white circle. The entrance to the space is in the lower-left corner of each plot; the exit is in the top-left corner
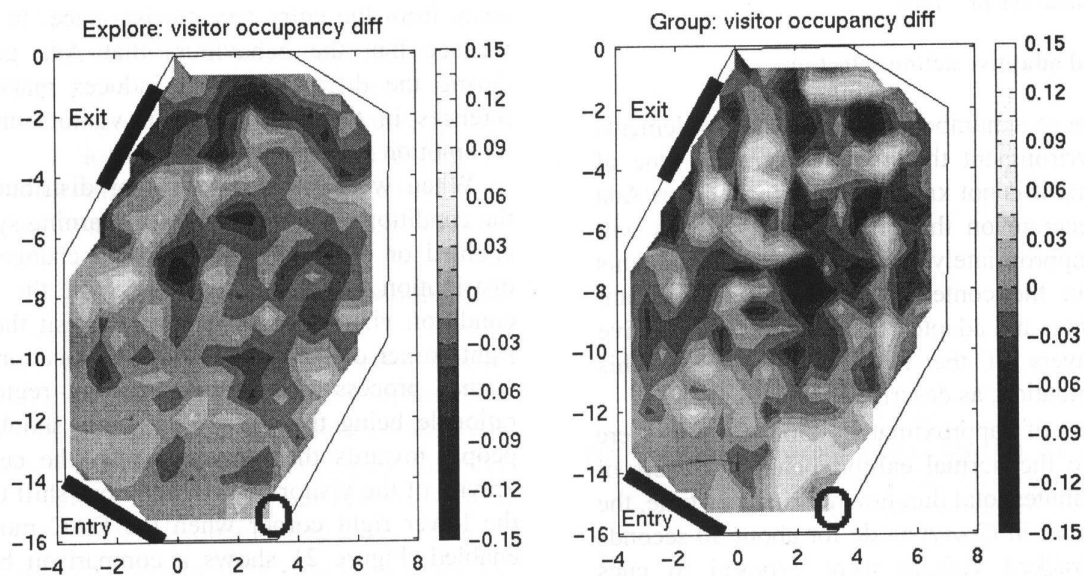

Fig. 21: *Group* process affects visitor distribution. Shown is the difference between the visitor distribution contours with the cue learning system enabled or disabled, averaged over approx. 18 cycles. The target location of the *Group* process is indicated by a black circle. (Left) Difference plot for *Explore* mode. (Right) Difference plot for *Group* mode.

on or off. As expected, in *Explore* mode the difference is virtually zero. However, when the *Group* process is switched on, the visitor concentration around the target location increases dramatically when the cue learning system is enabled. The other peaks in the distribution are due to visitors standing still, possibly indicating visitors who are unsure about how to respond to the cues.

Figure 22 highlights the difference in the movement of the visitor centre of gravity. The overall centre of gravity of visitors shifts by about 1.5 m towards the goal area, while no similar effect can be seen in a control case with the *Group*

process switched off. The effect of the shift also tended to produce an offset that persisted into the following (*Game*) mode. Hence, the grouping cues learned by the space through trial and error are effective in guiding the visitors.

In order to understand whether this difference in visitor behaviour is due to DAC we need to analyse its learning performance. If learning were effective one would expect to see the association of one particular action (cue CR) with a particular value of crowdedness (CS). The evolution of the synaptic weights of the DAC learning system is shown in Figure 23. There are eight different cues (US) and
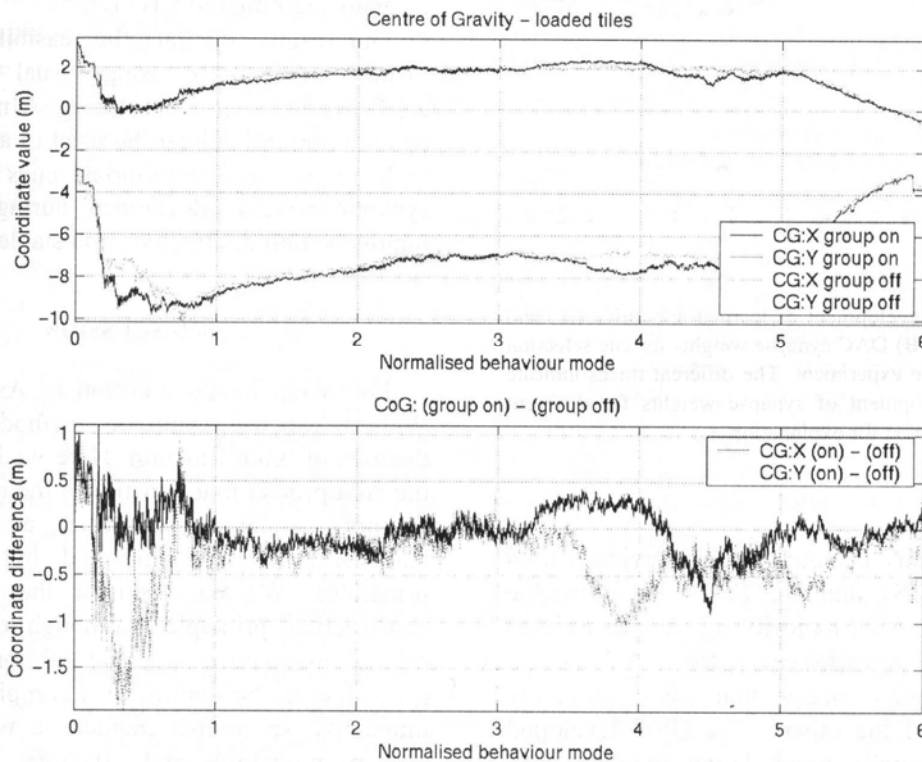

Fig. 22: Group process affects visitor distribution. (**Top**) Movements of the visitor centre of gravity (CoG) during the behavioural cycle with *Group* mode learning enabled (solid lines) and disabled (dotted lines). All curves are averages over ~18 visitor cycles. The length of each cycle has been normalised on the horizontal axis as follows: *Sleep* 0-1, *Wake* 1-2, *Explore* 2-3, *Group* 3-4, *Game* 4-5, *End* 5-6. (**Bottom**) Difference between x and y coordinates for *Group* mode on and off. At the start of *Group* mode (3-4), both cases have their CoG at the same position. For the case in which the *Group* process is switched on, there is a measurable CoG shift during the *Group* mode of (-1.0, +0.35) m in the (x, y) direction, relative to the case in which it is switched off. At the start of *Game* mode, the visitors then move rapidly in the (-x, +y) direction. The deviations between the curves in the *Sleep* and *End* modes are due to normal fluctuations in the entry/exit patterns of groups of visitors.
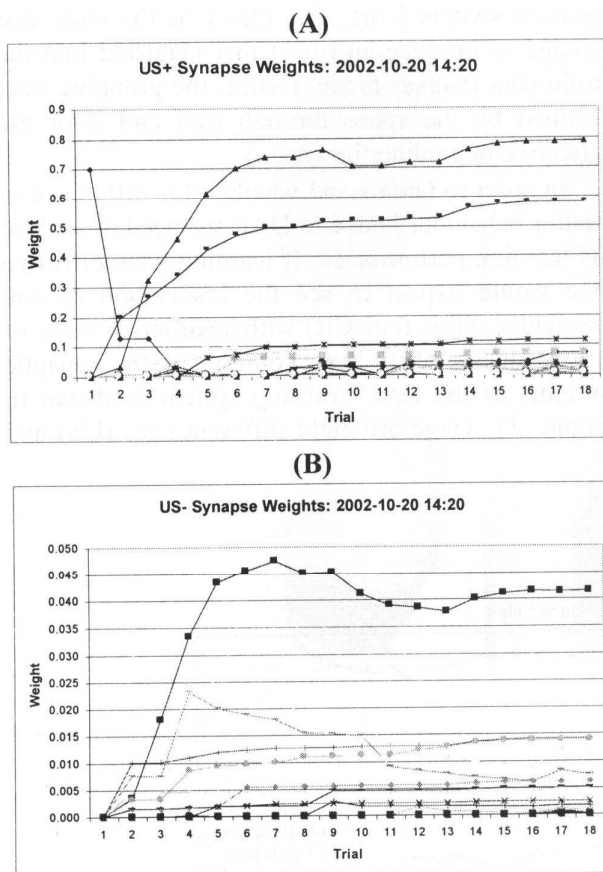
(A)

(B)

Fig. 23: Typical development of learned appetitive (**A**) and aversive (**B**) DAC synapse weights for cue selection during one experiment. The different traces indicate the development of synapse weights for different cues. See text for explanation.

four visitor density categories (CS), giving a total of 32 (US+, CS) and 32 (US-, CS) possible associations. The vast majority of synapse weights remains close to zero; however, for both US+ and US- a few weights emerge that are significantly larger than all of the others. The US+ developed weights are generally much larger than the US- weights. The evolution of the synaptic strength between the CS and IS populations, representing the learned cue, shows that learning occurs very rapidly. In case of IS^+ the synaptic weights converge after ten trials, i.e. ten *Group* mode cycles. There is an initially high weight of 0.7 that turns out to be a spurious response; this cue turns out to be unsuitable and is automatically discarded by the

learning process over time. In this example a cue consisting of a flashing blue tile was selected. A second action was also reinforced (flashing red tile) but it did not translate into overt actions due to the winner-take-all mechanism in DAC. For IS^-, learning progressed more rapidly, and after five trials a cue was identified that was not effective. The learning dynamics seems to suggest a competition among several cues. However, this could be due to variable visitor responses. It is important to note that visitors were not aware about this cue learning system or instructed to react in any particular way to the cues it generated. Hence, the reinforcement received by DAC was highly variable. Our model, however, shows robust learning performance and does not show catastrophic forgetting due to exceptions, a well-known problem of many learning models /47/.

Our results illustrate the feasibility of guiding visitors in a space using visual cues, without explicit prior visitor prompting. A neural classical conditioning model can be used to allow the space to learn the most appropriate cues to apply. The synaptic weights developed during learning are rapidly acquired, effective, and stable over time.

6. DISCUSSION

Following in the tradition of Ashby and Grey Walter, we use synthetic methods to evaluate theories of brain function. Here we have presented the Ada project that constitutes the most ambitious attempt so far to construct a real-time and real-world artefact controlled by neuromorphic principles. We have shown that neuromorphic construction principles, although originally conceived as models of natural organisms, can generalise to the control of a complex artefact, an interactive space that includes a wide variety of sensory modalities and effectors. Moreover, we have shown that neuronal models of auditory localisation and behavioural learning generalise well to the tasks Ada had to solve. The modular organisation of the Ada architecture also allowed for a clear separation between algorithmic solutions to specific problems and neuromorphic ones. Running experiments using the space has allowed us to simultaneously test theories of behavioural

control and modulation, artificial emotional expression, auditory processing and learned action selection. We were able to do this using multiple high bandwidth data recording techniques that would simply not be possible using any known animal-based paradigm. By also running this space as a live public exhibit, we have demonstrated that the capability now exists for neuromorphic technologies to be more widely deployed in society.

Ada's closest relative from a project perspective was also her direct physical neighbour - the EPFL robotics exhibit at Expo.02 /49/. This project dealt with different technical content to Ada (autonomous cooperating museum guide robots), but both projects were of similar size and operated under similar conditions.

There is a developing trend to use robots for the investigation of models of brain function (see /3/ and /40/ for partial reviews). It is beyond the scope of this article to review all these approaches. We therefore limit ourselves to a more restricted comparison with other projects. Several research projects deal with issues related to home automation and 'intelligent rooms', and many companies offer commercial home automation systems, such as the GE Smart series from GE Industrial Systems /23/. This system offers a substrate for connecting electrical devices and home network services with a common software interface. The control system software is based on rule sets or driven directly by end users, either within the building or via remote links. In this sort of system, the design emphasis is on ease of end-user installation, operation and customisation, rather than advanced behavioural functionality.

Other control systems exist in projects such as the Intelligent Room at MIT /35/. The Intelligent Room project aims to develop systems that support human activities in a seamless, flexible way. To date, work has been done on context-aware speech and gesture recognition, flexible resource allocation /22/ and an agent-based extension to Java called MetaGlue. Ada has a similar set of functionalities, but with three main differences. Firstly, Ada is a product for general public use and is much larger than the Intelligent Room, in terms of physical size, number of components and degree of behavioural integration. Secondly, the design of the user inter-

action with the space is immersive rather than invisible - the building does not serve its users' needs in the background, but is an active participant in their experiences. Thirdly, Ada achieves her own goals by actively engaging her users.

A similar project, also named the Intelligent Space, is being pursued at the University of Tokyo /2,31/. The general approach is to design a platform to facilitate communication between the entities that inhabit it - whether they be humans, robots, or components of the space itself. The concept of a Distributed Intelligent Network Device (DIND) is proposed for connecting devices in the space. Each DIND has sensors, processing and communications components. In this way the space is seen not as an explicit entity like Ada, but as a common networking medium in a physical area. Another group at the University of Tokyo has developed a system for accumulating human behaviour in a small prototypical apartment /39/ using mainly tactile sensors. Two noteworthy developments are a pressure-sensitive bed and a high-resolution pressure-sensitive floor /36/ for use in the invalid care industry in which 24-hour monitoring of patients is desirable.

An animal-like analogue to Ada is the *Mutant* dog robot /20/ and its commercially available successor *Aibo* from Sony. Ada and Aibo are both complete systems designed to interact with the general public, and both integrate visual, audio and tactile information to produce behaviour. They both have an internal emotional model and layered system architectures: Aibo's architecture is agent-based, while Ada has a hybrid of simulated neural networks and agent-based software components. Sony has formalised its system architecture in the OPENR model for building robots /19/. The main differences between Aibo and Ada are the obvious ones of appearance and size. By looking like a dog, Aibo has an inherent advantage over Ada for human interactions. A decision made in designing Ada was to explore the limits of human interactions that could be supported without the use of pre-existing metaphors, and to discourage visitors from anthropomorphising the system. In this way, visitors to Ada had to be convinced of her 'intelligence' by Ada's behaviour, rather than by any projections of attributed cognitive or emotional

abilities that could occur with an animal-shaped robot. The designers of Aibo were also able to draw on the rich available literature on canine behaviour when designing the behavioural models used in the robot /3/. On the engineering front, Aibo has the dual challenges of miniaturisation and minimising power consumption, whereas Ada faces power consumption constraints on a much larger scale.

There are also robots that, like Ada, seek to emulate the functions of organisms and interact with humans. Two notable humanoid robot examples are the SDR-4X from Sony /3/ and Asimo/P3 from Honda /44/, which are both now reaching commercial application. Asimo has a system architecture with horizontal and vertical integration rather similar to Ada's, but with a different set of behaviours for the different problem set applicable to humanoid robots. In particular, SDR-4X and Asimo must cope with navigation and legged locomotion problems that are not applicable to Ada. Instead, Ada deals with the complementary problem of visitor tracking and identification. Another well-known example is the humanoid torso Cog /10/. The Cog project has so far dealt more with individual competencies rather than overall behaviours, such as visual-motor processing, human interaction with robot facial expressions based on an emotional model, and neural models of arm motor control. The emotional model in Cog runs on a head-only subsystem of Cog called Kismet /9/, and has some similarities to Ada's emotional model - both contain a set of drives (goal functions) and a set of emotional states. While individual components of Cog have achieved coordinated functionality, the individual behaviours have not yet been integrated into a cohesive whole, and it has not made the move from the laboratory into the real world in the same way as Ada, Asimo or Aibo.

As of this writing, more in-depth data analysis of Ada's performance is also ongoing in the following areas:

- automatic calibration of gazers and multi-modal visual/tactile tracking;
- assessment of visitor reactions to the exhibit based on demographic measures, and an investigation of the effects of various manipulations of the functionality of the space on visitor perception;
- the ability of the space to actively affect the speed and position of visitors, and the effect of boundary conditions (entry/exit placement, pre-conditioning sequences) on their distribution in the space;
- detailed floor tracking characteristics.

Further development of the system lies in the direction of the incorporation of more biologically realistic models for different components, in particular the behaviour modulation and learning processes. We expect that it should be possible for a modified version of the DAC system to be used for the overall behavioural modulation of Ada. Individual components, such as the auditory system, floor tiles and visual processing, will also be further enhanced in future. In particular, the vision capabilities of Ada (during the exhibition visual information did not directly affect Ada's behaviour) will be elaborated.

It is clear that we are seeing a trend towards the creation of artificial organisms that is gathering in momentum, in terms of the sophistication of the techniques used and the diffusion of these creatures from the laboratory into everyday life. The design of the computational abilities of these systems is, in many cases, inspired by what has been learned from observing natural organisms. Analysis of artificial organisms will, in turn, lead to a deepening of the understanding of how their biological counterparts work, and open up new fields of application beyond the reach of natural organisms. The technology used in Ada is the first of these new applications. Ada is one of the largest-scale artificial organisms yet created, in terms of size, computational power and number of sensors and effectors. Her construction is based on explicitly declared neuromorphic design principles, and was the result of a large-scale multi-disciplinary research and development effort of approximately 50 man-years. During 5 months of active operation, Ada successfully entertained over half a million visitors, while also serving as a platform for research into several different topics. We have demonstrated that large coherent neuromorphic systems containing multiple agent-based and neural models can be successfully deployed and evaluated. While definite plans were not available at the time of writing,

it is expected that Ada will be reassembled in a new location to allow an ongoing mix of research, development and public performance.

APPENDIX 1

Summary of devices and drivers in Ada

Device	Product, Maufacturer	Interface	Server
Floor tile network	Custom. Mechanical construction by Westiform; Niederwengen, Switzerland. Interface card by Hilscher; Hattersheim, Germany	Interbus	Floor_server
Gazer	Mechanism: custom modified Martin MAC250; Århus, Denmark Sony EVI-400 zoom camera block; Tokyo, Japan	Mechanism: DMX on PCI card Camera block: VISCA RS-232	DMX_server Visca_server
Ambient light	IHGL-CCU MK II DMX-controlled neon-tube box; Despar, Mainz, Germany	DMX on PCI card	DMX_server
Light finger	Martin MAC250; Århus, Denmark	DMX on PCI card	DMX_server
Video camera	Hauppauge frame grabber	PCI card	Vision_server
Video output	Sharp XGA video projector; Osaka, Japan Matrox XGA dual-head graphics card; Dorval, Canada	PCI card	BigScreen
Sound input	Audio-Technica Pro45 unidirectional cardoid condenser; Stow, OH, USA M-Audio Delta44 sound card	PCI card	Audio
Sound output	Akai S5000 sampler; Tokyo, Japan	MIDI	Roboser

APPENDIX 2

Operations performed by behavioural processes on each server
Each process (column) performs operations on multiple servers (rows)

	Track @ floor_server	Identify @ object_server	Group @ cue	Play @ game_server
Floor_server	• Identify persons • Weigh persons • Display colour labels	• Display compliance test cues • Display compliance reward effects • Display effects for localised sounds	• Display grouping cues	• Display game background • Display game objects
DMX_server	• Assign labels to persons and groups	• Direct light fingers at person • Direct gazer at person		
Visca_server		• Set camera zoom to get an optimal image		
Vision_server		• Collect images and videos of salient persons		
BigScreen		• Display saved image, live video and recorded trajectory of person on screen		
Audio		• Supply localised sound location and type		
Roboser				• Play game-specific sound effects
Object_server	• Store person trajectories and weights	• Store compliance statistics	• Retrieve visitor locations	• Retrieve visitor locations
Game_server				• Game algorithm and state control
Cue			• Generate grouping cues	
Behaviour_server	• Set operation mode	• Set operation mode	• Set operation mode	• Set operation mode

REFERENCES

1. Aertsen AM, Johannesma PI. The spectro-temporal receptive field. A functional characteristic of auditory neurons. Biol Cybern 1981; 42: 133-143.
2. Appenzeller G, Lee J-H, Hashimoto H. Building topological maps by looking at people: an example of cooperation between intelligent spaces and robots. In: International Conference on Intelligent Robots and Systems (IROS 1997), 1997.
3. Arkin R. Behavior-Based Robotics. Cambridge, MA: MIT Press, 1998.
4. Arkin RC, Fujita M, Tagaki T, Hasegawa R. An ethological and emotional basis for human-robot interaction. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002), Lausanne, Switzerland, 2002.
5. Ashby WR. Design for a Brain. London: Chapman & Hall, 1952.
6. Bernardet U, Blanchard M, Verschure PFMJ. IQR: a

distributed system for real-time real-world neuronal simulation. Neurocomputing 2002; 44-46: 1043-1048.

7. Bizzi E, Gizterb SF, Loeba E, Mussa-Ivaldic FA, Saltiela P. Modular organization of motor behavior in the frog's spinal cord. TINS 1995; 18: 442-446.

8. Borbely AA, Achermann PJ. Sleep homeostasis and models of sleep regulation. J Biol Rhythms 1999; 14: 557-568.

9. Breazeal C, Scassellati B. A context-dependent attention system for a social robot. In: Joint Conference on Artificial Intelligence (IJCAI99), Stockholm, Sweden, 1999.

10. Brooks RA, Breazeal C, Marjanovic M, Scassellati B, Williamson MM. The Cog Project: building a humanoid robot. Lecture Notes in Computer Science 1999; 1562: 52-87.

11. Clancey WJ. Situated Cognition: On Human Knowledge and Computer Representations. Cambridge: Cambridge University Press, 1996.

12. Cohen L. Time Frequency Analysis. Englewood Cliffs, NJ: Prentice Hall, 1995.

13. Cordeschi R. The Discovery of the Artificial. Behavior, Mind and Machines Before and Beyond Cybernetics. Dordrecht: Kluwer, 2002.

14. Dallos P, Popper AN, Fay RR, eds. The Cochlea. New York: Springer-Verlag, 1996.

15. Delbrück T, Douglas RJ, Marchal P, Verschure P, Wahtley AM. A device for controlling a physical system. EU Patent Application no. 99120136.9-2215, 1999.

16. Depireux DA, Simon JZ, Klein DJ, Shamma SA. Spectro-temporal response field characterization with dynamic ripples in ferret primary auditory cortex. J Neurophysiol 2001; 85: 1220-1234.

17. Edwards MA, Adams DB. Role of midbrain central gray in pain-induced defensive boxing of rats. Physiol Behav 1974; 13: 113-121.

18. Ekman P. Moods, emotions, traits. In: The Nature of Emotion. Fundamental Questions. Oxford: Oxford University Press, 1994; 56-58.

19. Fujita M, Kageyama K. An open architecture for robot entertainment. In: International Conference on Autonomous Agents. ACM Press; 1997.

20. Fujita M, Kitano H. Development of an autonomous quadruped robot for robot entertainment. Autonomous Robots 1998; 5: 7-20.

21. Gabrielsson A, Juslin PN. Emotional expression in music performance: between the performer's intention and the listener's experience. Psychol Music 1996; 24: 68-91.

22. Gajos K, Weisman L, Shrobe H. Design principles for resource management for intelligent spaces. In: International Workshop on Self-Adaptive Software (IWSAS '01), 2001.

23. General Electric Industrial Systems. GE Smart web page (www.ge-smart.com) 2002.

24. Graziano MSA, Taylor CSR, Moore T, Cooke DF. The cortical control of movement revisited. Neuron 2002; 36: 349-362.

25. Hendriks-Jansen H. Catching Ourselves in the Act. Cambridge, MA: MIT Press, 1996.

26. Hofstoetter C, Mintz M, Verschure PFMJ. The cerebellum in action: a simulation and robotics study. Eur J Neurosci 2002; 16: 1361-1376.

27. Infoweek, Infoweek online poll: Welche IT-Ausstellung an der Expo 02 hat Sie am meisten beeindruckt? 2002.

28. Kilgard MP, Merzenich MM. Cortical map reorganization enabled by nucleus basalis activity. Science 1998; 279: 1714-1718.

29. Kim JJ, Krupa DJ, Thompson RF. Inhibitory cerebello-olivary projections and blocking effect in classical conditioning. Science 1998; 279: 570-573.

30. Kowalski N, Depireux DA, Shamma SA. Analysis of dynamic spectra in ferret primary auditory cortex. I. Characteristics of single-unit responses to moving ripple spectra. J Neurophysiol 1996; 76: 3503-3523.

31. Lee J-H, Hashimoto H. Intelligent space - its concept and contents. Advanced Robotics 2002; 16: 265-280.

32. Logothetis NK, Sheinberg DL. Visual object recognition. Annu Rev Neurosci 1996; 19: 577-621.

33. Maslow A. Toward a Psychology of Being, 3rd Ed. New York: Wiley, 1998.

34. McFarland D, Bosser T. Intelligent Behavior in Animals and Robots. Cambridge, MA: MIT Press, 1993.

35. MIT Intelligent Room Project web page (www.ai.mit.edu/projects/iroom), 2002.

36. Morishota H, Fukui R, Sato T. High resolution pressure sensor distributed floor for future human-robot symbiosis environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002), Lausanne, Switzerland, 2002.

37. Mussa-Ivaldi FA, Bizzi E. Motor learning through the combination of primitives. Phil Trans R Soc Lond B 2000; 355: 1755-1769.

38. Newell A. Physical symbol systems. Cognitive Sci 1980; 4: 135-183.

39. Noguchi H, Mori T, Sato T. Construction of accumulation system for human behavior information in room. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002), Lausanne, Switzerland, 2002.

40. Pfeifer R, Scheier C. Introduction to New Artificial Intelligence. Cambridge, MA: MIT Press, 1999.

41. Popper AN, Fay RR, eds. The Mammalian Auditory Pathway: Neurophysiology. New York: Springer Verlag, 1992.

42. Redish AD. Beyond the Cognitive Map: From Place Cells to Episodic Memory. Cambridge, MA: MIT Press, 1999.

43. Russell JA. Affective space is bipolar. J Soc Psychol 1979; 37: 345-356.

44. Sakagami Y, Watanabe R, Aoyama C, Matsunaga S, Higaki N, Fujimura K. The intelligent ASIMO: system overview and integration. In: IEEE/RSJ International

Conference on Intelligent Robots and Systems (IROS 2002), Lausanne, Switzerland, 2002.

45. Sanchez-Montanes MA, Verschure P, König P. Local and global gating of synaptic plasticity. Neural Comput 2000; 12: 519-529.

46. Saper CB, Chou TC, Elmquist JK. The need to feed: homeostatic and hedonic control of eating. Neuron 2002; 36: 199-211.

47. Sutton RS, Barto AG. Reinforcement Learning: An Introduction. Cambridge, MA: MIT Press, 1998.

48. Thorndike EL. Animal Intelligence. 1911.

49. Tomatis N, Terrien G, Piguet R, Burnier D, Bouabdallah S, Siegwart R. Design and system integration for the Expo.02 robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 2002.

50. Varela FJ, Maturana HR, Uribe R. Autopoiesis: the organization of living systems. BioSystems 1974; 5: 187-196.

51. Verschure PFMJ, Althaus P. A real-world rational agent: unifying old and new AI. Cognitive Sci 2003; in press.

52. Verschure PFMJ, Coolen ACC. Adaptive fields: distributed representations of classically conditioned associations. Network 1991; 2: 189-206.

53. Verschure PFMJ, Kröse B, Pfeifer R. Distributed adaptive control: the self-organization of structured behavior. Robot Auton Syst 1992; 9: 181-196.

54. Verschure PFMJ, Mintz M. A real-time model of the cerebellar circuitry underlying classical conditioning:

a combined simulation and robotics study. Neuro-computing, 2001; 38-40: 1019-1024.

55. Verschure PFMJ, Voegtlin T. A bottom-up approach towards the acquisition, retention, and expression of sequential representations: Distributed Adaptive Control III. Neural Networks 1998; 11: 1531-1549.

56. Verschure PFMJ, Wray J, Sporns O, Tononi G, Edelman G. Multilevel analysis of classical conditioning in a behaving real world artifact. Robot Auton Syst 1995; 16: 247-265.

57. Voegtlin T, Verschure P. What can robots tell us about brains? A synthetic approach towards the study of learning and problem solving. Rev Neurosci 1998; 10: 291-310.

58. Walter WG. A machine that learns. Sci Am 1951; 184: 60-63.

59. Wang K, Shamma SA. Self-normalization and noise robustness in early auditory processing. IEEE Trans Audio Speech 1994; 2: 421-435.

60. Wassermann KC, Blanchard M, Bernardet U, Manzolli JM, Verschure PFMJ. Roboser: an autonomous interactive composition system. In: International Computer Music Conference (ICMC), San Francisco, CA, USA, 2000. The International Computer Music Association.

61. Wassermann KC, Manzolli J, Eng K, Verschure PFMJ. Live soundscape composition based on synthetic emotions. IEEE Multimedia (submitted).

62. Webb B. Robots in invertebrate neuroscience. Nature 2002; 417: 359-363.