An Interactive Space That Learns to Influence Human Behavior

Kynan Eng, *Member, IEEE*, Rodney J. Douglas, and Paul F. M. J. Verschure, *Associate Member, IEEE*

***Abstract*—A key question in the design of intelligent environments is how a space can influence the actions of its users, and how such behavior can be learned. In this paper, we present the results of experiments conducted as part of the Ada project, an interactive entertainment exhibit deployed at the Swiss national exhibition Expo.02. We used a learning model called distributed adaptive control (DAC) that is based on the animal learning paradigms of classical and operant conditioning. DAC has been developed using mobile robots in foraging tasks. Here, it was applied to the learning of effective cues for guiding visitors in a given direction. Our results show that, by using this learning mechanism, Ada was able to influence the behavior of visitors by learning to deploy particular types of cues. Many visitors could be induced to move toward a region of the space that they normally avoided visiting—an effect that can be seen as a spatial classification of visitors into interactive and noninteractive categories. In our analysis, we also introduce a measure of human activity that combines different types of data to capture key aspects of human behavior in interactive spaces.**

***Index Terms*—Behavior, human–machine interaction, interactive space, learning.**

I. Introduction

A COMMONLY stated goal of ambient intelligence is the seamless support of human activities through the use of pervasive sensor and effector technologies. Many approaches to the creation of such environments focus on natural input, using techniques such as speech and gesture recognition. However, the resulting behavior of the system often follows a simple predefined master–slave control mechanism, perhaps with some context-dependent processing to resolve ambiguous inputs. We argue that such approaches do not address the key issues concerning intelligent environments, because they ignore the symbiotic nature of the human–environment relationship. In true symbiosis, control is a two-way street—environments must be able to actively affect human behavior as well as vice versa. In this context, the key questions for intelligent environments can be phrased as follows.

1) Language: how can an environment and its inhabitants communicate, and what limits should (and can) be placed on this interaction?
2) Learning: given an agreed upon language, how can environments and their inhabitants adaptively influence each other's behavior in order to achieve common or complementary goals?

A large body of knowledge already exists to address the first question, i.e., in the context-sensitive speech and gesture recognition system [3] that is part of the Massachusetts Institute of Technology intelligent room project [4]. Another noteworthy feature of this project is the use of gaze estimation to activate voice recognition, where a user looks at a projection of a face on a screen in order to signal that the system should prepare to receive voice commands [5]. The broadly similar Interactive Workspaces project at Stanford University [6] focuses on the use of multiple high-resolution displays in multiperson collaboration, and ways of interacting with various computing devices (e.g., context-sensitive pointing and handwriting recognition). More passive communication methods are being explored at the University of Tokyo, which has developed a system for accumulating human behavior in a small prototypical apartment [7], using mainly tactile sensors. In this project, a human goes about his/her normal daily routine while being monitored by devices, such as a pressure-sensitive bed [8], a pressure-sensitive chair [9], and a high-resolution pressure-sensitive floor [10]. One envisioned application of such a room is in background patient monitoring and abnormal patient behavior detection in the invalid care industry. Microsoft Research has also entered the field with its EasyLiving project [11], [12], using visual tracking of humans and devices such as mice and keyboards to enable automatic context identification and preference retrieval. So far, both explicit voice or gesture-based languages and implicit movement-based ones have been explored. The Ada project has explored an alternative approach by developing a nonverbal language that uses the media of sound and light, combined with tactile movement tracking.

The second question is more problematic, and consists of two parts: 1) inhabitants learning to influence environments and 2) environments learning to influence inhabitants. Given that the inhabitants in our case are humans, we can treat the human learning problem as being solved by our own genetic inheritance. The key issue here is to understand the scope and limits of human perceptual and behavioral learning capabilities in order to allow optimal interfacing to and interaction with artefacts. We made the assumption that humans are active explorers and learners as long as they are sufficiently motivated. The visitor appreciation of the exhibition suggests that this assumption was justified. However, what about environments learning to influence humans? In investigating this question, we set ourselves the following boundary conditions and assumptions.

Manuscript received October 15, 2003; revised March 30, 2004 and June 1, 2004. This work was supported in part by the Swiss Federal Institute of Technology, Zurich, in part by the University of Zurich, in part by the Swiss Expo02, in part by Manor AG, the Velux Foundation, and in part by the Gebert Rüf Foundation. This paper was recommended by Guest Editor G. L. Foresti.

The authors are with the Institute of Neuroinformatics, Swiss Federal Institute of Technology (ETH), Zurich CH-8057, Switzerland (e-mail: kynan@ini.phys.ethz.ch; rjd@ini.phys.ethz.ch; pfmjv@ini.phys.ethz.ch).

Digital Object Identifier 10.1109/TSMCA.2004.838467

1) Language: the environment can only influence its inhabitants using the nonmechanical means of light and sound. The practical motivation for the exclusion of mechanical means of interaction was safety. The light and sound design should be generally pleasant, as it is preferable to influence behavior by coaxing rather than by coercion (e.g., via the use of very unpleasant sounds) as suggested by the work of Skinner and others [13]. Furthermore, the lights and sounds should not include natural language-dependent features such as written or spoken words in order to, among other things, avoid an anthropomorphization of the artefact. (We believe that although the facilitation of anthropomorphization may make sense for some humanoid robots, it can hamper long-term development by denying that most embodied artefacts are not humans.) The environment can receive its input from its users either via sounds or movement.

2) Learning: we assume that, in essence, an environment is an inside-out robot. This means that existing algorithms for mobile robots should be able to be translated into a suitable form for use in an interactive space. The key distinction is that, in the former case, a large part of the dynamics of system environment interaction is self-generated, while in the latter case, it mostly originates in the movements of other entities, i.e., the inhabitants. This implies that although an intelligent space does not usually have to solve the self-localization problem, all other aspects of perceptual and behavioral learning are surprisingly similar. Moreover, algorithms that can be successfully adapted to this different task domain could be said to be closer to meeting Newell's requirement for general intelligence, where anything can be a task [14]. Hence, applying algorithms tested on mobile robots to an interactive space serves both a practical goal of generalization and a basic science goal in the study of intelligence.

The above boundary conditions and assumptions were behind our concept of Ada, an interactive entertainment space that was built and exhibited at the Swiss national exhibition Expo.02. The 160-m^2 main space of Ada contains visual, audio, and tactile inputs, and noncontact light and sound effectors. Visitors to the space are immersed in an environment where their only significant sensory stimulation comes from Ada (and other visitors). Like an organizm, Ada's output is designed to have a certain level of coherence and convey an impression of a basic unitary sentience to Ada's visitors. Ada 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. Development of Ada commenced in late 1998 and involved over 30 person-years of effort (including ten public demonstrations and tests) in its technical components. The exhibit ran continuously for up to 12 hours a day over five months from May 15, 2002 to October 20, 2002. During this period, over 550 000 visitors entered the space.

The following sections of this paper contain a brief overview of the main components of Ada, followed by a description of the learning model that we tested for influencing the behavior of visitors. The experimental procedures are then presented, followed by the results of the experiments and suggestions for future extensions.

Fig. 1. Typical live user-interaction scene within Ada during *explore* mode. Visible are floor tiles, a visitor being highlighted by a light finger (center left: the light finger itself is in the ceiling frame), a dynamic 3-D visualization (top) and a live gazer video on the screens (top left) provided by BigScreen (from [2]).

II. About Ada

This section is intended to give only a brief overview of Ada. More complete information about Ada can be found in [15]–[17] and [2], as well as the official Ada web page [18].

In total (including auxiliary exhibition areas), Ada has 15 video inputs, 367×3 tactile inputs, 9 audio input channels, 46 mechanical degrees of freedom, 17 output audio channels, 367×3 floor tile lights, 30 ambient lights, and 20 full-screen video outputs (Fig. 1). All of these inputs and outputs can be addressed independently, giving a rich array of sensory modalities and output possibilities. Ada has the following sensory capabilities.

Vision: Pan-tilt cameras called *gazers* 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, with which Ada is able to localize sound sources by triangulation. Some basic forms of sound and word recognition and pitch extraction are available.

Touch: Ada has a "skin" of 0.66-m-wide hexagonal pressure-sensitive floor tiles [19] that can detect the presence of visitors by their weight. Each contains a microcontroller and sits on a serial bus running an industrial automation protocol called interbus.

As well as sensing, Ada can also express itself and act upon its environment in the following ways.

Visual: Ada uses a 360° ring of 12 LCD projectors called BigScreen to express its internal states and visitor interaction dynamics. These projectors collectively show a single, unified display of three-dimensional (3-D) objects covering multiple screens in real time, as well as 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.

Audio: Ada is able to generate a wide range of sound effects. These sounds can be distributed across the entire space or localized using a matrix mixer. Ada expresses herself using sound and music composed in real time on the basis of Ada's internal states and sensory input. Ada can also change the pitch of Ada's output depending on what Ada hears from Ada's visitors. The composition is generated using a system called Roboser [20].

Touch: Ada has 20 16-bit pan-tilt *light fingers* 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.

A *tracking system* uses data from the floor-tile pressure sensors to determine the location, speed, direction, and weight of visitors. 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. To obtain more information about individual visitors, a *vision system* deploys gazers to collect images of people who have been localized on the floor. The *audio system* localizes and recognizes basic sounds (e.g., the word Ada, pitch, note, and key) to help identify salient individuals. On the output side, the *Roboser* system composes real-time music and sound effects, a *video server* allows the visualization of saved and live images, video streams, and trajectory plots of identified visitors, and a *DMX server* controls the light fingers, gazers, and ambient lights. The software implementing all of these functions is a mix of procedural/object-oriented C/C++, Java-based software agents, and models of large-scale neuronal circuits built using a real-time simulation package called IQR [21].

During normal operation, the main space received about 25 visitors at a time and entertained them for about 4–6 min, depending on the interactions that occurred. Due to the extremely large number of visitors that wanted to see Ada, it was necessary to control visitor flow very rigidly to avoid problems with overcrowding. This was desirable both for safety reasons and to ensure that each visitor had a reasonable amount of space in which to interact with Ada. Before entering the main space, all visitors were given a brief introduction to Ada by passing through a *conditioning tunnel* containing examples of Ada's components. While watching the group in front of them, visitors also received brief verbal explanations (in German, French, or English) of how to interact with Ada.

To provide for a natural progression in visitor interaction, Ada incorporates six basic behavioral modes: *sleep, wake, explore, group, play*, and *end*. In *sleep*, visitors enter Ada, filled with soft blue floor and light finger effects, simple reactive floor behaviors, and soft music. They can then induce Ada to *wake* by running around and making noise. As a result, Ada will display a sudden change in the ambient illumination conditions to bright yellow and a different music style. This transient phase settles down to *explore* mode, the longest phase of Ada's activity cycle. During this phase, visitors are tracked as they wander around the space and their compliance is tested by periodically assessing their responsiveness to visual cues consisting of single flashing white tiles next to them. Visitors who respond to the cues by stepping on the flashing tiles build up compliance, until they are rewarded with light fingers directed at them and live video of themselves projected on the BigScreen. This sequence of interactions expresses the principle of resource allocation based on behavioral salience, i.e., Ada only provides device attention to visitors who are responsive to interactive cues and are, therefore, interesting to Ada. After a while, the space switches to *group* mode, which contains similar interactions to *explore* mode but with one key addition: all tracked visitors may be given cues to induce them to move to a particular target location. Upon reaching the target location, the space switches to *play* mode, where a simple game is played where visitors try to step on an animated bouncing tile. Finally, it is time for the visitors to leave when Ada enters *End* mode, characterized by sad music, dark red color schemes and tile cues moving toward the exit of the space.

In this paper we concentrate on how Ada was able to learn the most effective cues to deploy to influence people to move to a particular location in the space during *group* mode. The following sections describe the learning architecture behind this aspect of Ada's learning capability and the results of various experiments designed to measure its effectiveness.

III. Visitor Grouping: Neural Adaptive Action Selection Using Distributed Adaptive Control (DAC)

The learning module operating in *group* mode is based on a neural model of classical and operant conditioning called DAC [22]. DAC is a model of the basic paradigms of animal learning; classical and operant conditioning. Here we focus solely on the classical conditioning part. Classical conditioning goes back to the work of Pavlov in the late 19th and early 20th century [23]. He showed that when a motivational stimulus, e.g., food, or unconditioned stimulus (US) was paired with a neutral stimulus, e.g., a sound, or conditioned stimulus (CS), the latter would over trials trigger a conditioned response (CR), e.g., salivation, similar to the unconditioned response (UR) triggered by the US alone. This form of learning is now seen as one of the fundamental mechanisms underlying adaptive behavior. Given a set of appetitive and aversive $(\mathrm{US}^{+}/\mathrm{US}^{-})$ and CS, DAC selects UR based on a set of predefined ($\mathrm{US}^{\pm}$, UR) reflex mappings. If a CS is consistently paired with a US, DAC will acquire a CR. In this way, DAC directly implements the notion of stimulus substitution, a standard interpretation of the classical conditioning paradigm.

DAC consists of three tightly coupled layers of behavioral control: reactive, adaptive, and contextual control (Fig. 2). The reactive control layer provides a behaving system with a prewired repertoire of reflexes, which enable it to interact with its environment and accomplish simple automatic behaviors. 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. 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- and long-term memory. These representations are used to control ongoing behavior in

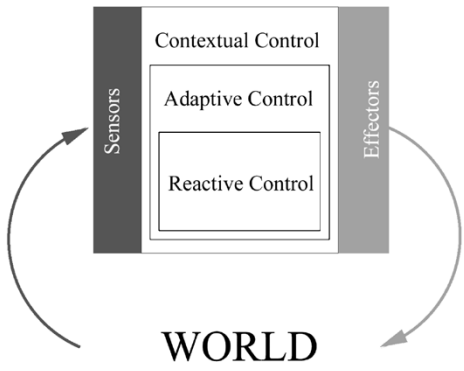

Fig. 2. Overview of the DAC architecture that consists of three layers: reactive, adaptive, and contextual control. (See text for a detailed explanation.)

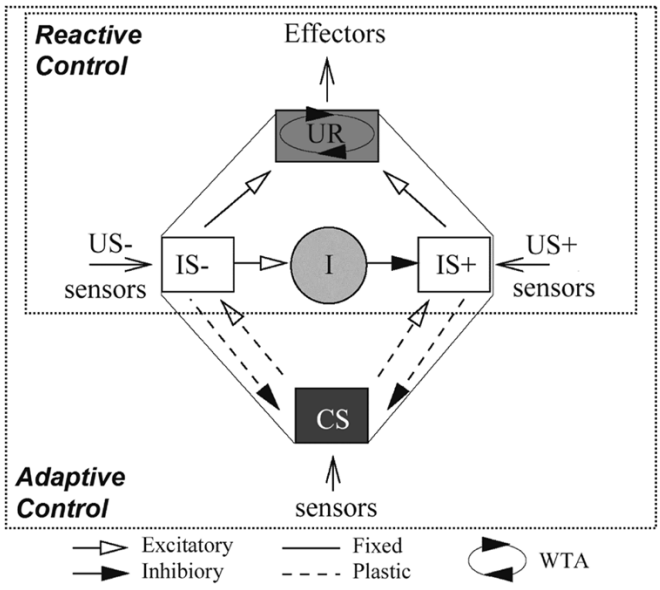

Fig. 3. DAC reactive and adaptive control layers. $(-)$ = aversive, $(+)$ = appetitive. (See text for an in-depth explanation; adapted from [1].)

the context of behavioral plans. DAC has been tested using different simulated and real mobile robots and has been shown, among other things, to approximate an optimal, in a Bayesian sense, solution to foraging [1], [24], [25]. For Ada, only the reactive and adaptive control levels of DAC are used.

A. Reactive Control

The reactive control structure (Fig. 3) is implemented by four populations of simulated neurons that are interconnected by prewired fixed connections. The sensor readings of the appetitive US$^+$, and the aversive US$^-$, are projected onto two populations of internal state (IS) neurons, IS$^+$ and IS$^-$. The input v_i^k of cell i in IS population k is defined by

$$v_i^k = p_i^k - \gamma^k I \tag{1}$$

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. Note that all variables are time varying.

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

$$o_i^k = H(v_i^k - \theta^k) \tag{2}$$

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

Conflicts in action selection are resolved through competition between IS$^+$ and IS$^-$. This competition is expressed via the 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$^-$

$$v_i^I(t+1) = \alpha^I v_i^I(t) + \gamma^I \frac{1}{M^{\mathrm{IS}^-}} \sum_{j=1}^{M^{\mathrm{IS}^-}} o_j^{\mathrm{IS}^-}(t) \tag{3}$$

where α^I is a decay factor $(\alpha^I < 1), M^{\mathrm{IS}^-}$ is the size of the IS$^-$ population and o^{IS^-} is the activity of the IS$^-$ population. The activity of population I is determined by thresholding v^I with θ^I (3).

Both IS populations will depolarize specific neurons in population UR, which represents particular actions. The input, r_l, of unit l in the UR population is defined as

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

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 using θ^{UR}. If its activity is above threshold, it will induce a particular motor action, or a CR or UR. If none of the neurons of the internal state or motor populations are active, the reactive control structure will resort to its default action of *exploration*.

B. Adaptive Control

The adaptive control layer is defined on the basis of the reactive control layer. It adds the components dealing with the processing of CS events, and their association with internal states and overt responses. The activity u_j of unit j in population CS is derived from the state s_j of element j of the input sensor

$$u_j = f(s_j) \tag{5}$$

where f is a transduction function.

CS, in turn, excites populations IS$^+$ and IS$^-$, in a modified form of (1). In this new form, the input, v_i^k, of cell i in IS population k is defined by

$$v_i^k = \sum_{j=1}^{N} w_{ij}^k u_j + p_i^k - \gamma^k I \tag{1'}$$

where N is the size of the CS population, w_{ij}^k is the efficacy of the connection between CS cell j and cell i of IS population k. 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.

The synapses forming the connections between these populations are modifiable using a predictive Hebbian learning rule [25]. This method embeds a local learning rule in a recurrent circuit. After the inputs v^k of the IS populations are updated (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

$$u'_j = u_j - \gamma^r e_j \tag{6}$$

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

$$e_j = \sum_{k=1}^{K} \sum_{i=1}^{M^k} \frac{w_{ij}^k v_i^k}{M^k} \tag{7}$$

where K denotes the number of IS populations, M^k is the size of IS population k, w_{ij}^k is the strength of the connection between cell i of IS population k and cell j of the CS population, and v_i^k 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:

$$\Delta w_{ij}^k = \eta^k v_i^k u'_j \tag{8}$$

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 to particular IS states. This solution allows the use of local learning rules, while preventing problems such as overgeneralization, primacy, and saturation [25].

DAC has been investigated using formal approaches [26] and robots [22], [25], [27], [28], and has been shown to be compatible with formal Bayesian models of human decision-making [1]. In one task, a mobile robot associated colored patches on the floor of an arena with actions in order to minimize 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 [28]. The DAC architecture has established itself as a standard in the field of artificial intelligence and behavior based robotics [29]–[34]. 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 [35], [36] and the adaptive timing of responses by the cerebellum [37]–[39].

A. Newell defined general intelligence as the ability to make anything a task [14]. This implies that any architecture underlying such an ability must be shown to generalize easily to different task domains. Although DAC was originally developed as a model of classical and operant conditioning as applied to mobile robots, here, we assessed whether it generalizes to other learning situations. In Ada, DAC was applied to the learning of cues for guiding visitors to a particular location. For both the mobile robots and Ada, the DAC architecture was employed as a model of adaptive action selection where the generic learning problem consists of two elements: 1) stimulus identification, i.e., deciding which events are behaviorally relevant 2) and action selection, i.e., deciding which action allows the system to achieve its goals given these salient stimuli. In this respect, DAC deviates from popular machine-learning approaches, such as reinforcement learning that aim at solving the latter problem assuming *a priori* definition of an input space [40].

To learn how to guide visitors, Ada must choose a behavior (a cue), deploy it (show the cue to a visitor), and evaluate the effect on the environment (the visitor's subsequent action). It is not known *a priori* what types of cues will be effective. If the visitor follows the cue and moves toward the goal location, then it was effective; if the visitor moves away (or does not move at all), then it was not. This provides a mapping of visitor movement toward the goal position as appetitive (US^+) and movement away as aversive (US^-). Ada was given a library of four different cue colors (red, green, blue, and white) and two types of cue (single flashing tile and travelling bullet toward goal), giving a total of eight different cues. Thus, there are eight different types of $\mathrm{US}^+/\mathrm{UR}$ and eight corresponding $\mathrm{US}^-/\mathrm{UR}$ 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 easily be placed in dynamically changing positions.

In principle, the initially neutral CS could be composed of many multimodal components combining tactile, audio, and visual information. However, for simplicity the CS was set to reflect the general level of crowdedness of the space, with four neurons coding increasing levels of visitor crowdedness. The rationale behind this choice was twofold. First, 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. Second, 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.

Fig. 4 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 compliance testing procedure described in Section II) is tracked. An error signal calculated by the *ranger* module, indicating motion toward or away from the goal position, is fed into the *cue* module. Once activated by *behavior_server*, the *cue* module converts and gates the error signal into DAC as a US^+ or US^- coded with 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. The density coding consists of four cells which are active as follows: 0–10 visitors $\rightarrow$ cell 0; 11–20 visitors $\rightarrow$ cell 1; 21–30 visitors $\rightarrow$ cell 2; >31 visitors $\rightarrow$ cell 3.

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 $(\mathrm{UR} = 0)$

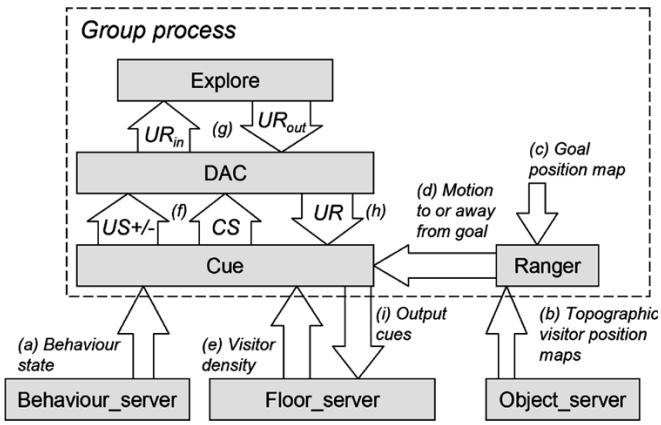

Fig. 4. Overview of neural pathways within the *group* process. Boxes inside the dashed rectangle provide for the core functionality of this behavioral mode. Behavior_server, Floor_server, and Object_server provide the key interface processes to the Ada infrastructure. (a) The process is activated when signalled by the Behavior_server. (b) The topographic location of the most salient visitor provided by the Object_server is compared with (c) the goal position and (d) a motion error signal is generated is generated by the Ranger module. (e) The motion error signal and (e) the current visitor density form the (f) $US^{\pm}$ and CS signals, respectively. (h) The UR coming from DAC can then be output to produce (i) the selected cue. If no US or strong enough CS is available, (g) a default explore behavior is triggered, which maps a random action onto the output.

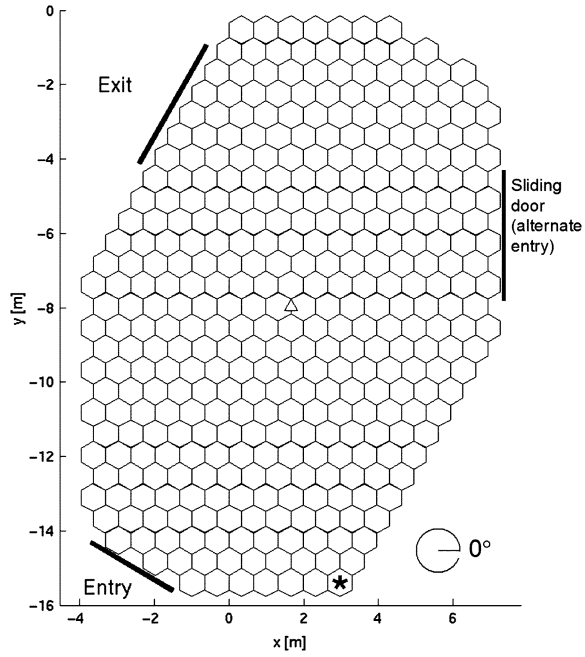

Fig. 5. Tile layout of Ada main space (360 tiles) and location of target tile for *group* mode experiments (marked with an asterisk). Each tile is about 0.66 m across. The entire space was surrounded by half-mirrored walls, except for the entry and exit, which were left open. The sliding door was not used.

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 within DAC 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 in DAC provides for the selection 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 initialize these rules.

IV. Adaptive Behavior Selection Tests

Groups of approximately 25–30 visitors were exposed to the normal exhibit behavior cycle of about 5-min duration. Before entering the space, the visitors were given the normal explanation of the interactions in Ada, but were not told anything about *group* mode. During *group* mode, which was fixed to about 30-s duration, all tracked visitors were exposed to cues generated by DAC. The cues were all directed to one corner of the space (Fig. 5), which previous measurements had shown that visitors were very unlikely to visit during a typical stay in the space. Three different cases were tested:

1) no grouping cues (18 visitor cycles);
2) DAC learning of grouping cues (19 visitor cycles);
3) fixed-grouping cues (white bullet of tiles toward target, length four tiles, cycle time ∼1 Hz) (11 visitor cycles).

Visitor tracking and tile-occupancy data were recorded simultaneously, along with a digital video record of the experiments. At the start of every trial started, the synapse weights between the CS and IS populations were set to 0. During each trial with a visitor group in the DAC learning condition, the synapse weights evolved by the DAC module were recorded.

V. Results

Two different measures were considered to evaluate the effects of DAC on visitor behavior: their global CoG, and their tendency to visit the area around the target tile. These measures are discussed in the following Sections V-A and V-B, respectively.

A. Visitor Distribution: CoG

One of the most simple measures of overall group distribution is their CoG. The CoG is defined as the mean x and y coordinates of all loaded tiles at each point in time. Fig. 6 shows a comparison of the progression of the visitor CoG for the DAC learning and no cues cases. When the group process is switched on, there is a relative CoG shift during *group* mode of up to $(-1.0, +0.35)\ m$ in the (x, y) direction. The effect of the shift tended to persist during the following (*game*) mode, and lasted until about halfway through *end* mode. This represents a change of visitor behavior lasting for over a minute. However, the shift was not found to be significant (two-tailed t test, $p > 0.05$) in either the x or y direction due to the large variability of the visitor CoG at any timestep. The deviations between the curves in *sleep* and *end* modes are due to normal fluctuations in the entry/exit patterns of groups of visitors.

These results could mean that there was no significant underlying change in visitor behavior, but this interpretation seems unlikely since visual observation indicated that a large proportion of the visitors did follow the learned cues. Rather, it is possible that visitor behavior may have changed but maintained the

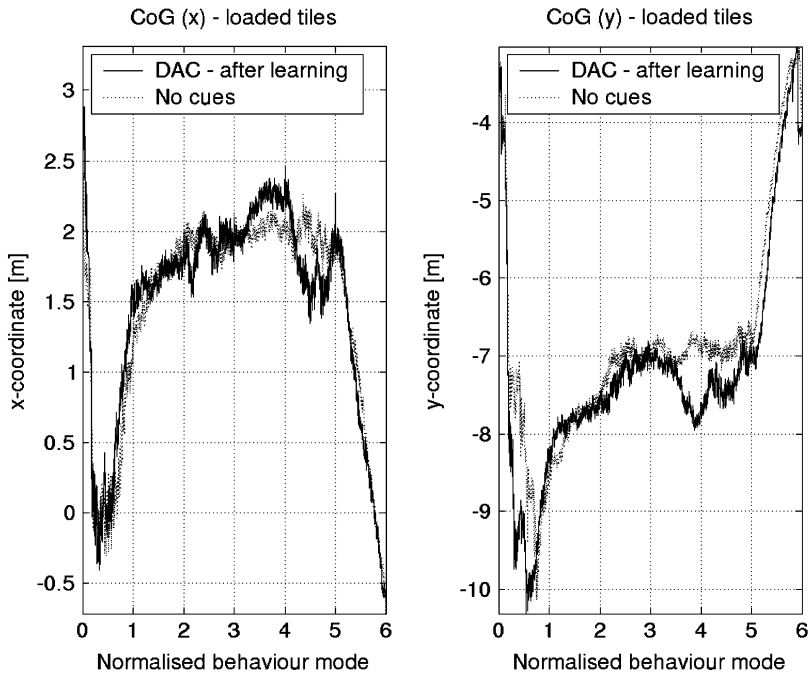

Fig. 6. Effect of *group* mode on visitor CoG. Shown are the x and y coordinates of the visitor CoG during a behavioral cycle with DAC learning enabled (solid lines, averaged over 13 cycles) and disabled (dotted lines, averaged over 18 cycles). The learning-enabled data are shown after the synaptic weights have stabilized (after six cycles). The length of each cycle is normalized on the horizontal axis as follows: sleep 0–1; wake 1–2; explore 2–3; group 3–4; game 4–5; and end 5–6.

same overall average, e.g., if half of the visitors follow the cues and the other half remain stationary, then their overall CoG will not change much. As will be seen in the following section, this is indeed what happened in the grouping experiments.

B. Visitor Distribution: Tile Occupancy and Tile Event Rate

Two measures of visitor activity will be referred to in this section: tile occupancy and tile event rate. Tile occupancy is defined as the fraction of the total length of the behavior mode that a particular tile is loaded (someone is standing on it). This is a static measure of visitor behavior, since a person could be standing on a single tile without moving—this would lead to high occupancy for that tile, even though the visitor is probably not doing very much. In contrast, the tile event rate is defined as the frequency of step-on or step-off events for a particular tile. This is a dynamic measure of visitor behavior, as a tile which is constantly being jumped on will have a high event rate, without necessarily having a high average occupancy. Both measures are required to gain an overall understanding of the movements of visitors in the space.

A comparison of visitor behavior during *group* mode, with DAC switched on or off, is shown in Fig. 7. These plots show the difference between the DAC on and off cases; i.e., identical visitor behavior would show up as a difference of 0. The difference in occupancy shows a bias toward the lower righthand corner of the space (middle panel). There are also some dark patches with negative relative occupancy, which is to be expected since the overall visitor occupancy, for the whole floor should remain roughly the same if nothing else changes in the behavior mode. Each plot considered on its own appears to be highly noisy, even though it represents the average of many trials. The main reason for this is that there are many more tiles than there are visitors, and *group* mode is relatively short, meaning that the probability of any tile being loaded during a given behavior mode is low. As a result, there are many isolated peaks in the plots, representing places where a single visitor lingered on a tile (in the case of tile occupancy) or was particularly active (in the case of tile event rate).

While both the tile occupancy and the tile event rates are increased for *group* mode with DAC learning switched on, the separate plots are not visually convincing. This is largely due to the high noise levels, and the fact that we are considering two partly complementary aspects of visitor behavior: occupancy and event rate. What is needed is a combination of the two measures to quantify areas with high activity, i.e., those with both high tile occupancy and high tile event rates. A straightforward way of doing this is to define the visitor activity difference ΔA as the weighted sum of the changes in the normalized visitor occupancy distribution O and the tile event distribution E

$$\begin{aligned}\Delta A_i &= \frac{1}{2}\left(\frac{\Delta O_i^d - \min(\Delta \mathbf{O}^d)}{\max(\Delta \mathbf{O}^d) - \min(\Delta \mathbf{O}_d)}\right) \\ &\quad + \frac{1}{2}\left(\frac{\Delta E_i^d - \min(\Delta \mathbf{E}^d)}{\max(\Delta \mathbf{E}^d) - \min(\Delta \mathbf{E}^d)}\right) \\ &= \text{difference in activity at tile } i \text{ between group} \\ &\quad \text{mode learning on and off} \\ \Delta O_i^d &= O_i^{\text{group\_on}} - O_i^{\text{group\_off}} \\ &= \text{difference in averaged occupancy at tile } i \\ &\quad \text{between group mode learning on and off} \\ \Delta E_i^d &= E_i^{\text{group\_on}} - E_i^{\text{group\_off}} \\ &= \text{difference in averaged tile event rate at tile } i \\ &\quad \text{between group mode learning on and off.}\end{aligned}$$

(Here, the weighting factors have been set to 0.5 in both terms, but this need not necessarily be the case, depending on

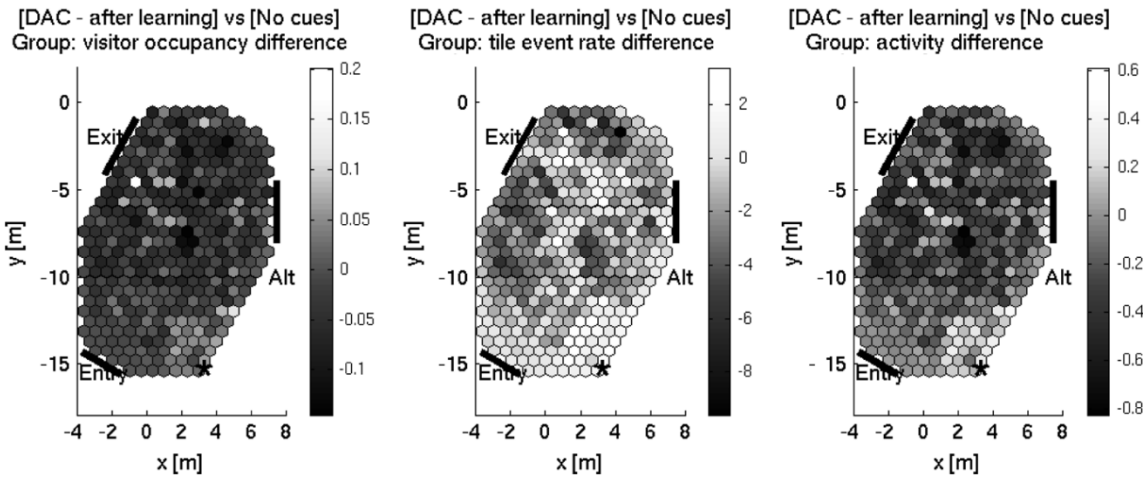

Fig. 7. Effect of *group* mode on (a) tile occupancy, (b) tile event rate, and (c) activity distributions (averaged data). Positive values on the contour scale indicate areas of positive changes in occupancy, tile event rates, or activity as caused by DAC. The target tile is indicated by a black asterisk in the lower-right corner of the space.

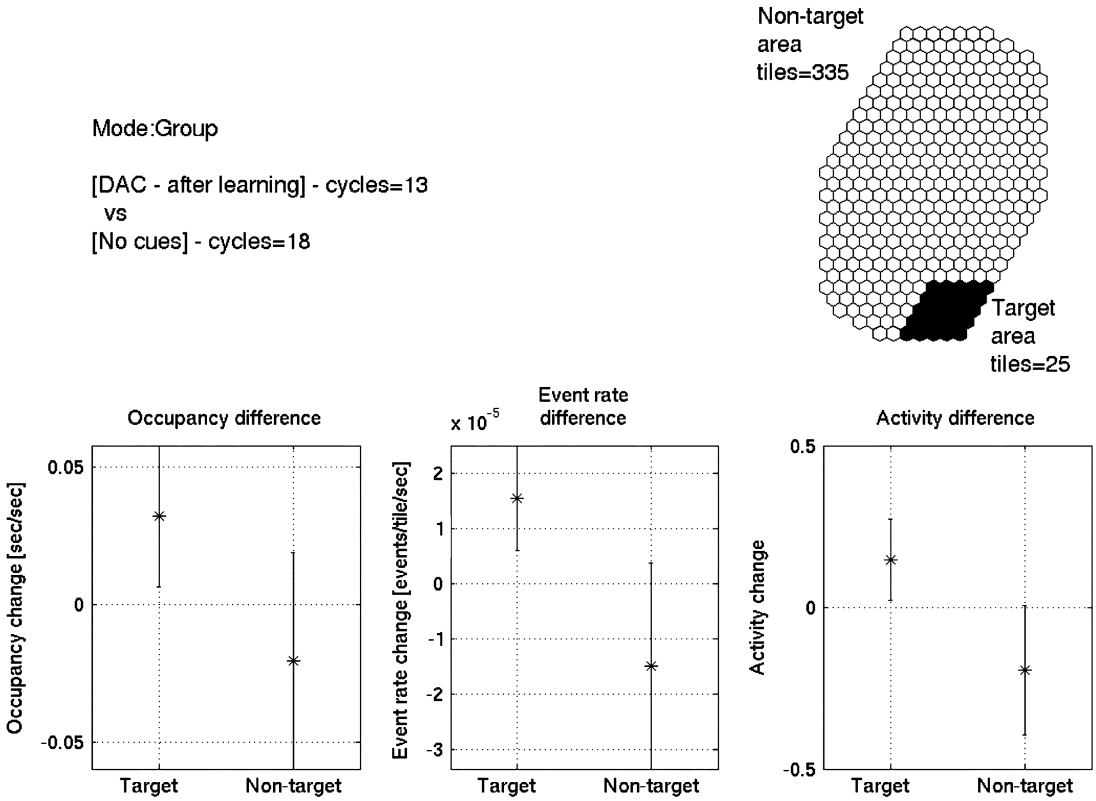

Fig. 8. Comparison of average changes in target and nontarget areas of the floor during *group* mode for DAC-learned cues and no cues. Shown are differences in tile occupancy, tile event rate, and tile activity. Error bars indicate one standard deviation.

the relative importance given to tile occupancy and tile event rates.) The resulting visitor activity plot (Fig. 7, right panel) shows a much clearer shift of visitor activity during *group* mode in the direction of the target location induced by DAC.

To check whether the observed changes are significant, we can compare the visitor activity measures for two different regions: the target area and the rest of the floor. We define the target area as being all tiles within a five-tile radius of the target (Fig. 8). Fig. 8 was chosen to allow for the size of the different cues which could be up to four tiles long. As can be seen, the mean changes in the tile occupancy, tile event rate, and activity level were all higher for the target area than in the rest of the floor. The differences are significant for all measures (two-tailed t test, $p \ll 0.001$), with the most significant change for the combined activity measure. No similar change was seen for *explore* mode. This significant, mode-dependent change in activity shows that the cues learned by DAC during *group* mode are effective in influencing visitor behavior.

One objection that could be raised to this result is that any cue might have worked to influence visitor behavior, i.e., DAC

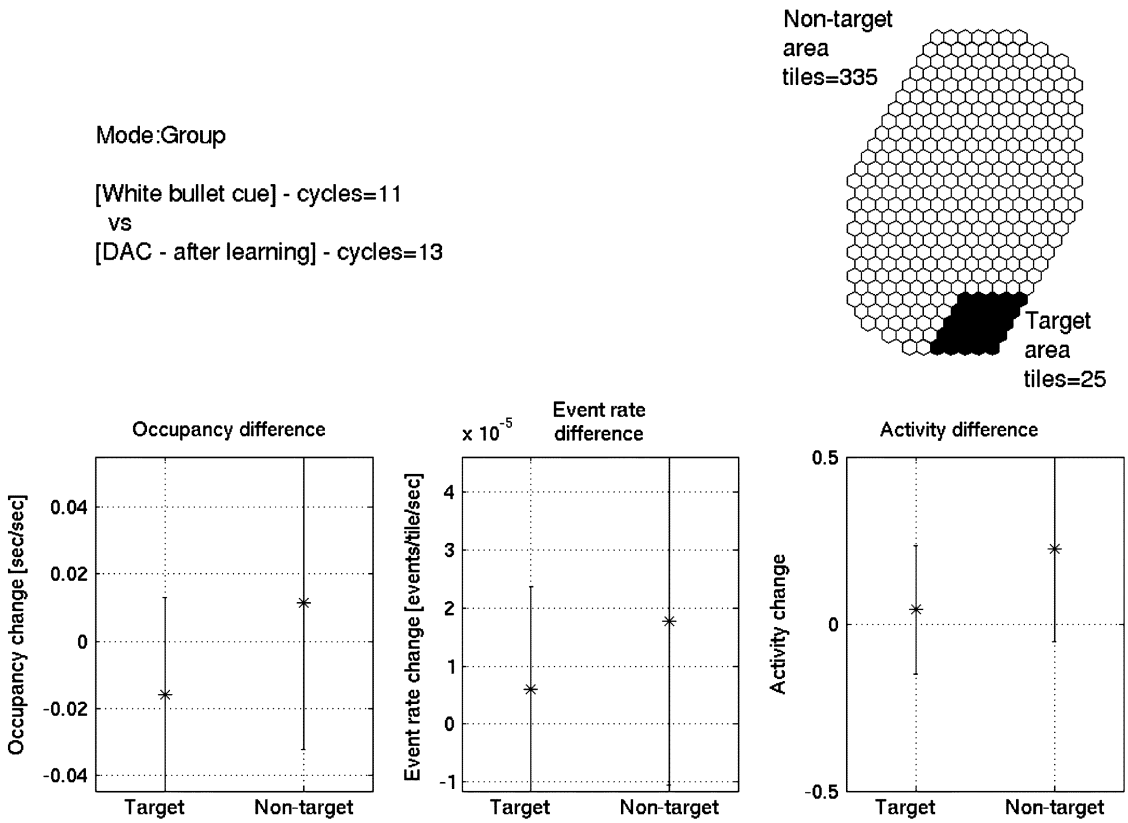

Fig. 9. Averaged comparison of changes in target and nontarget areas of the floor for fixed white bullet cues and DAC-learned cues for *group* mode. Shown are differences in tile occupancy, tile event rate, and tile activity. Error bars indicate one standard deviation.

did not learn anything useful. This can be addressed by using a control case, where visitors were always exposed to a constant cue during *group* mode. This cue—an animated white bullet shooting four tiles in the cue direction at a rate of 1 Hz—was chosen as a very visually salient cue. Moreover, visitors were already attuned to the color white as a cue color, since the compliance testing during *explore* mode made use of cues consisting of single flashing white tiles. A comparison of the change in visitor behavior in *group* mode (Fig. 9) shows that the changes in the target area were significantly smaller for the white bullet cue than for DAC (occupancy: $p < 0.01$, event rate: $p < 0.05$, activity: $p < 0.01$). Thus, we can conclude that DAC was more effective at influencing visitor behavior than the selected control case cue.

Overall, we can conclude that enough of the visitors followed the cues enough of the time to cause a measurable increase in their activity in the area near the cue target. However, their overall CoG was not significantly changed, suggesting that their distribution relative to each other changed during *group* mode.

C. Synapse Weights

In the previous sections, we saw that the cues acquired by DAC were able to effectively influence visitor behavior. In order to verify that the observed difference in visitor behavior is due to the internal processing in DAC, its learning performance must be analyzed and the selected cues identified. If learning was effective one would expect to see the association of one particular action (cue type CR) with a particular value of crowdedness (CS). The evolution of the synaptic weights of the DAC learning system for a typical run is shown in Fig. 10. There are eight different cues (US) and four visitor density categories (CS), giving a total of 32 (US^+, CS) and 32 (US^-, CS) possible associations. Most of the synapse weights remain close to zero; however, for both US^+ and US^- a few weights emerge that dominate 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 the case of IS^+, the synaptic weights converge after nine *group* mode cycles (trials); in this example, a cue consisting of a single flashing blue tile was selected. A second action was also reinforced (single flashing red tile), but it did not translate into overt actions due to the winner-take-all mechanism in DAC. For IS^-, learning progresses more rapidly, and after five trials a cue is identified that is not effective (single flashing blue tile). The fact that the single flashing blue tile appears in both and IS^+ and IS^- is evidence of the inconsistent input provided by different most compliant visitors; however, this does not cause DAC to fail as the weight in IS^+ remains stable and much higher than the IS^- weight. The weights continue to change over time, suggesting a competition among several cues.

In addition to the learning phase, recall tests were conducted in which learning was allowed to proceed for about 10–12 cycles (about 1 h), after which the synaptic weights were frozen and the $\mathrm{US}^{\pm}$ signals cut off from DAC. Thus, the input that DAC would receive was the CS signal, which should be enough to elicit a consistent UR on its own. Visual observation of the recall tests conducted in this way indicated that the CS-driven recall

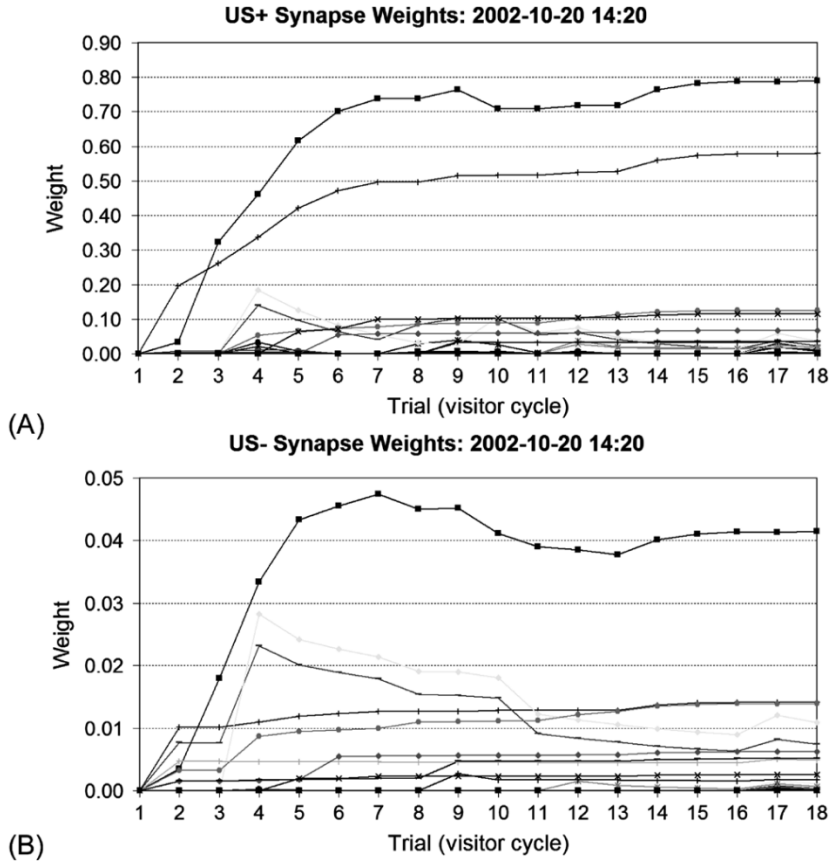

Fig. 10. Typical evolution of DAC weights in *group* mode. The weights shown are at the end of each visitor cycle (1 cycle = 1 trial). Shown are typical developments of learned (A) appetitive and (B) aversive DAC synapse weights for cue selection during one experiment. The different traces (line colors/symbols) indicate the development of synapse weights for different cues.

did indeed occur as expected, demonstrating that DAC had been able to store its learned knowledge correctly.

VI. Discussion

This study assessed the ability of the DAC architecture for classical conditioning to generalize from mobile robot based paradigms to an interactive space. In parallel our experiments evaluated whether the behavior of the visitors to this space could be influenced in an adaptive and dynamic fashion. Our results show that DAC is able to learn to select cues that are effective in influencing the positions of visitors despite high variability in the visitor reactions to the cues. Visitors were not aware of the DAC learning system or instructed to react in any particular way to the cues it generated. They were told about the compliance test in *explore* mode, and it is possible that they may have extrapolated this information to the multicolored, multitile cues used by DAC. However, the control test we ran showed that DAC was still able to learn a cue that was more effective than one similar to that used in the compliance test (which visitors already knew how to respond to). DAC was able to consistently develop and rapidly adapt stable weights, and it did not show catastrophic forgetting due to exceptions despite the highly variable user input, a known problem of other learning models developed in the field of reinforcement learning [40]. Thus, the DAC architecture, operating as designed, was able to learn to perform the task of adaptively influencing human behavior by extracting an effective relationship between the crowdedness of the space and the cues that should be deployed to guide the visitors.

The change in visitor behavior resulted in a significant local increase in activity around the target tile, without a corresponding significant change in the overall CoG of visitors. This probably reflects the observation that some visitors followed the cues, while others did not. Thus the system was acting as a sort of spatial visitor filter, i.e., at the end of *group* mode, the more interactive visitors were probably those nearest to the target tile. This effect of grouping visitors with similar behavioral properties could be used to allow future interactive spaces to customize their interactions according to broad similarities between spatially separated groups of visitors. Such a scheme could be particularly useful in situations where collecting data from single visitors over extended time periods is difficult or impractical, since the space can group similar visitors rather than trying to keep track of each visitor's individual characteristics. Moreover, this learning mechanism can allow an interactive space to shape its cues to its inhabitants.

Our analysis of two components of visitor behavior as measured by a tactile floor—tile occupancy and tile event rate—led to the definition of visitor activity as a normalized sum of the two components. This measure was able to provide convincing evidence of the effect that DAC had on the visitors. Based on this

result, we suggest that it is possible that the concept of visitor activity as defined here may reflect some key aspects of visitor behavior in interactive spaces. Such a multicomponent measure could be extended in the future to include other sensory modalities to provide richer visualizations and more accurate quantification of human behavior in these spaces.

The successful generalization of DAC to the task of influencing human motion suggests that it at least partly satisfies Newell's requirement for a generally intelligent system in which anything can be a task [14]. In general, DAC could be applied to any classical and operant conditioning learning task where a set of appetitive and aversive stimuli need to be matched to appropriate behaviors. The most significant challenge in doing this would be taking timing into account, which is important for achieving behavior switching on appropriate timescales for visitors. Timing issues in the cue-learning application discussed here were handled in an ad-hoc manner, but a more systematic approach would be needed to handle different tasks. The implementation of DAC used here (DAC2) does not include any timing mechanisms; however, later versions such as DAC3 and DAC5 do contain notions of action timing and sequencing [24]. Incorporating the notion of time (such as that found in later versions of DAC) into environmental control learning tasks will be important for supporting more advanced, finer-grained communication between environments and their inhabitants.

ACKNOWLEDGMENT

The authors gratefully acknowledge the entire Ada development team for their work on the project that made this series of experiments possible.

REFERENCES

[1] P. F. M. J. Verschure and P. Althaus, "A real-world rational agent: Unifying old and new AI," *Cogn. Sci.*, vol. 27, pp. 561–590, 2003.

[2] K. Eng, A. Baebler, U. Bernardet, M. Blanchard, M. Costa, T. Delbruck, R. J. Douglas, K. Hepp, D. Klein, J. Manzolli, M. Mintz, F. Roth, U. Rutishauser, K. Wassermann, A. M. Whatley, A. Wittmann, R. Wyss, and P. F. M. J. Verschure, "Ada—Intelligent space: An artificial creature for the Swiss Expo.02," presented at the IEEE/RSJ Int. Conf. Robotics Automation, Taipei, Taiwan, 2003.

[3] K. Gajos, L. Weisman, and H. Shrobe, "Design principles for resource management for intelligent spaces," *Lecture Notes in Comput. Sci.*, vol. 2614, pp. 198–215, 2003.

[4] MIT Intelligent Room Project Web Page (2002). [Online]. Available: www.ai.mit.edu/projects/iroom

[5] A. Oh, H. Fox, M. Van Kleek, A. Adler, K. Gajos, and M. L. T. Darrell, "Evaluating look-to-talk: A gaze-aware interface in a collaborative environment," presented at the Conf. Human Factors Comput. Syst., Minneapolis, MN, 2002.

[6] B. Johanson, A. Fox, and T. Winograd, "The interactive workspaces project: Experiences with ubiquitous computing rooms," *IEEE Pervasive Comput.*, vol. 1, no. 2, pp. 67–74, Apr./Jun. 2002.

[7] H. Noguchi, T. Mori, and T. Sato, "Construction of accumulation system for human behavior information in room," presented at the IEEE/RSJ Int. Conf. Intell. Robots Syst., Lausanne, Switzerland, 2002.

[8] T. Harada, T. Sato, and T. Mori, "Estimation of Bed-Ridden human's gross and slight movement based on pressure sensors distribution bed," presented at the IEEE Int. Conf. Robotics Automation, Washington, DC, 2002.

[9] T. Mori, Y. Okazaki, and T. Sato, "Measuring pain using pressure sensored chair," *Adv. Robotics*, vol. 14, pp. 413–414, 2000.

[10] H. Morishita, R. Fukui, and T. Sato, "High resolution pressure sensor distributed floor for future human–robot symbiosis environments," presented at the IEEE/RSJ Int. Conf. Intell. Robots Systems, Lausanne, Switzerland, 2002.

[11] Microsoft Research EasyLiving Web Page (2003, Mar. 26). [Online]. Available: http://research.microsoft.com/easyliving

[12] S. Shafer, B. Brumitt, and J. Cadiz, Interaction issues in context-aware intelligent environments, in Human–Comput. Interaction, vol. 16, no. 2–4, pp. 363–378, 2001.

[13] G. H. Bower and E. R. Hilgard, *Theories of Learning*, 5th ed. Englewood Cliffs, NJ: Prentice-Hall, 1981.

[14] A. Newell, "Physical symbol systems," *Cogn. Sci.*, vol. 4, pp. 135–183, 1980.

[15] K. Eng, D. Klein, A. Baebler, U. Bernardet, M. Blanchard, M. Costa, T. Delbruck, R. J. Douglas, K. Hepp, J. Manzolli, M. Mintz, F. Roth, U. Rutishauser, K. Wassermann, A. M. Whatley, A. Wittmann, R. Wyss, and P. F. M. J. Verschure, "Design for a brain revisited: The neuromorphic design and functionality of the interactive space Ada," *Rev. Neurosci.*, vol. 14, pp. 145–180, 2003.

[16] K. C. Wassermann, J. Manzolli, K. Eng, and P. F. M. J. Verschure, "Live soundscape composition based on synthetic emotions," *IEEE Multimedia*, vol. 10, no. 4, pp. 82–90, Oct. 2003.

[17] K. Eng, A. Baebler, U. Bernardet, M. Blanchard, A. Briska, M. Costa, T. Delbruck, R. Douglas, K. Hepp, D. Klein, J. Manzolli, M. Mintz, T. Netter, F. Roth, K. Wassermann, A. Whatley, A. Wittmann, and P. Verschure, *Ada: Buildings as Organisms*. Delft, Holland: Game, Set, Match, Faculty Architec., Tech. Univ. Delft., 2003, pp. 33–43.

[18] Ada Exhibit Press Kit (2004, Mar. 18). [Online]. Available: http://ada.ini.ethz.ch/

[19] T. Delbrück, R. J. Douglas, P. Marchal, P. F. M. J. Verschure, and A. M. Whatley, "A device for controlling a physical system," U.S. Pat. 6 603 082; EU Pat. Applicat. 99 120 136.9-2215, 1999.

[20] K. C. Wassermann, M. Blanchard, U. Bernardet, J. M. Manzolli, and P. F. M. J. Verschure, "Roboser: An autonomous interactive composition system," presented at the Int. Comput. Music Conf., San Francisco, CA, 2000.

[21] U. Bernardet, M. Blanchard, and P. F. M. J. Verschure, "IQR: A distributed system for real-time real-world neuronal simulation," *Neurocomputing*, vol. 44–46, pp. 1043–1048, 2002.

[22] P. F. M. J. Verschure, B. Kröse, and R. Pfeifer, "Distributed adaptive control: The self-organization of structured behavior," *Robotics Autonomous Syst.*, vol. 9, pp. 181–196, 1992.

[23] I. P. Pavlov, *Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex*. London, UK: Oxford Univ. Press, 1927.

[24] P. F. M. J. Verschure, T. Voegtlin, and R. J. Douglas, "Environmentally mediated synergy between perception and behavior in mobile robots," *Nature*, vol. 425, pp. 620–624, 2003.

[25] T. Voegtlin and P. Verschure, "What can robots tell us about brains? A synthetic approach toward the study of learning and problem solving," *Rev. Neurosci.*, vol. 10, pp. 291–310, 1998.

[26] P. F. M. J. Verschure and A. C. C. Coolen, "Adaptive fields: Distributed representations of classically conditioned associations," *Network*, vol. 2, pp. 189–206, 1991.

[27] P. F. M. J. Verschure, J. Wray, O. Sporns, G. Tononi, and G. Edelman, "Multilevel analysis of classical conditioning in a behaving real world artifact," *Robotics Autonomous Syst.*, vol. 16, pp. 247–265, 1995.

[28] P. F. M. J. Verschure and T. Voegtlin, "A bottom-up approach toward the acquisition, retention, and expression of sequential representations: Distributed adaptive control III," *Neural Netw.*, vol. 11, pp. 1531–1549, 1999.

[29] R. Arkin, *Behavior-Based Robotics*. Cambridge, MA: MIT Press, 1998.

[30] W. J. Clancey, *Situated Cognition: On Human Knowledge and Computer Representations*. Cambridge, MA: Cambridge Univ. Press, 1996.

[31] R. Cordeschi, *The Discovery of the Artificial. Behavior, Mind and Machines Before and Beyond Cybernetics*. Norwell, MA: Kluwer, 2002.

[32] H. Hendriks-Jansen, *Catching Ourselves in the Act*. Cambridge, MA: MIT press, 1996.

[33] D. McFarland and T. Bosser, *Intelligent Behavior in Animals and Robots*. Cambridge, MA: MIT Press, 1993.

[34] R. Pfeifer and C. Scheier, *Introduction to New Artificial Intelligence*. Cambridge, MA: MIT Press, 1999.

[35] M. P. Kilgard and M. M. Merzenich, "Plasticity of temporal information processing in the primary auditory cortex," *Nature Neurosci.*, vol. 1, pp. 727–731, 1998.
[36] M. A. Sanchez-Montanes, P. König, and P. F. M. J. Verschure, "Learning sensory maps with real-world stimuli in real time using a biophysically realistic learning rule," *IEEE Trans. Neural Netw.*, vol. 13, no. 3, pp. 619–632, May 2002.
[37] C. Hofstoetter, M. Mintz, and P. F. M. J. Verschure, "The cerebellum in action: A simulation and robotics study," *Eur. J. Neurosci.*, vol. 16, pp. 1361–1376, 2002.
[38] J. J. Kim, D. J. Krupa, and R. F. Thompson, "Inhibitory cerebello-olivary projections and blocking effect in classical conditioning," *Science*, vol. 279, pp. 570–573, 1998.
[39] P. F. M. J. Verschure and M. Mintz, "A real-time model of the cerebellar circuitry underlying classical conditioning: A combined simulation and robotics study," *Neurocomputing*, vol. 38–40, pp. 1019–1024, 2001.
[40] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

Kynan Eng (M'04) received the B.Sc. degree in applied mathematics/computer science and the B.E. degree in mechanical engineering from Monash University, Melbourne, Australia, in, 1994 and 1996, respectively, and the Ph.D. degree from the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, in 2004.

After a number of years in industry, working on telecommunications fault-management software systems and gas turbine design for power generation, he joined the Institute of Neuroinformatics, ETH, in 2000, to work on the Ada project. His main responsibilities within the project were the overall technical system integration, development of the behavioral control strategies, and the on-site operation of the exhibit. His research interests include interactive spaces, human–machine interaction, and robotics.

Dr. Eng is a member of the Swiss Society for Neuroscience.

Rodney J. Douglas received the M.B. and Ch.B. degrees in science and medicine and the Ph.D. degree in neuroscience from the University of Cape Town, Cape Town, South Africa, in 1974 and 1985, respectively.

He is a Professor of neuroinformatics and Co-Director at the Institute of Neuroinformatics, Swiss Federal Institute of Technology and the University of Zürich (ETH), Zurich, Switzerland. He was with the Anatomical Neuropharmacology Unit in Oxford, where he continued his research on the anatomy and biophysics of the microcircuitry of cerebral cortex together with K. Martin. As a Visiting Associate and then Visiting Professor at the California Institute of Technology, Pasadena, he extended his research interests in neuronal computation to the modeling of cortical circuits using digital methods (together with C. Koch), and also by the fabrication of analog very large scale integration circuits with M. Mahowald. In 1996, he and K. Martin moved to Zurich to establish the Institute of Neuroinformatics.

Prof. Douglas was awarded the Körber Foundation Prize for European Science in 2000.

Paul F. M. J. Verschure (A'92) received the M.A. degree from the University of Amsterdam, Amsterdam, Holland, in 1990 and the Ph.D. degree in psychology from the University of Zurich, Zurich, Switzerland, in 1992.

He is currently a Group Leader at the Institute of Neuroinformatics, Swiss Institute of Technology (ETH) and the University of Zurich, Zurich, Switzerland, as well as the Project Leader of *Ada: Intelligent Space*. He works on biologically realistic models of perception, learning, and problem-solving applied to real-world artefacts, ranging from flying robots to large-scale interactive spaces. He has also pursued research at institutions in the US and Europe.

Dr. Verschure is an Associate of Behavioral and Brain Sciences and a Member of the Association for Computing Machinery, the International Neural Network Society, and the Society for Neuroscience.