The Origin of Epistemic Structures and Proto-representations

Sanjay Chandrasekharan

Terrence C. Stewart

Institute of Cognitive Science,

Carleton University,

1125 Colonel By Drive,

Ottawa, Canada, K1S 5B6

schandra@sce.carleton.ca, sanjayen@gmail.com

tcstewar@connect.carleton.ca, terry.stewart@gmail.com

(e-mail contact preferred)

Contact Phone Number for First Author: 91-532-9935827275

Contact Fax Number for First Author: 91-532-2460738

Running Head: **The origin of epistemic structures**

Abstract: Organisms across species use the strategy of generating structures in their environment to lower cognitive complexity. Examples include pheromones, markers, colour codes, etc. Distributed Cognition theory has argued that studying such 'epistemic structures' can provide insights into the development and nature of internal representations, and cognition itself. We develop this claim by providing a model of the origin of such structures, and present a simulation where organisms with only reactive behaviour learn, within their lifetime, to add such structures systematically to their world to lower cognitive load. This implementation is then extended to show that the same underlying process could generate traces of the world in an 'internal environment' to lower cognitive load. We then examine two implications of this internal trace model. First, it provides a novel account of the origin of internal representations. Further, as both external and internal traces lower cognitive load and are generated using the same mechanism, the location of the structure becomes opportunistic, and a matter of utility. This supports the 'extended mind' hypothesis. Second, the stored internal traces develop entirely out of actions. They thus encapsulate action components and could activate actions. This feature explains the origin of enactable and action-oriented mental content.

Key Words: Distributed Cognition, Epistemic Structure, Extended Mind, Representation, Simulation Theory, Situated Cognition

Introduction

Many organisms add stable structures to their environments to reduce cognitive complexity (minimize search, inference, memory load etc.), for themselves, for others, or both. Wood mice (*Apodemus sylvaticus*) distribute small objects, such as leaves or twigs, as points of reference while foraging. Such 'way-marking' is exhibited even under laboratory conditions, using plastic discs, and has been shown to diminish the likelihood of losing interesting locations during foraging (Stopka & MacDonald, 2003). Red foxes (*Vulpes vulpes*) use urine to mark food caches they have emptied. This marking acts as a memory aid and helps them avoid unnecessary search (Henry, 1977, reported in Stopka & MacDonald, 2003). The male bower bird builds colorful bowers (nest-like structures), which are used by females to make mating decisions (Zahavi & Zahavi, 1997). Ants drop pheromones to trace a path to a food source. Many mammals mark their territories (Bradbury & Vehrencamp, 1998). Bacterial colonies use a strategy called 'quorum sensing' to know that they have reached critical mass (to attack, to emit light, etc.). This strategy involves individual bacteria secreting molecules known as auto-inducers into the environment. The auto-inducers accumulate in the environment, and when they reach a threshold, the colony moves into action (Silberman, 2003).

Such 'doping' of the world is commonly seen in smaller animals (such as rodents and insects). In general, large animals (such as horses, elephants, monkeys etc.) are not known to exploit this strategy. Humans are an exception to this general trend. The list of epistemic structures used by humans is almost endless -- markers, color-codes, page

numbers, credit-ratings, stains, traces, badges, shelf-talkers, speed bugs, road signs, post-it notes etc.

The pervasiveness of such epistemic structures across species indicates that adding structures to the world is a fundamental cognitive strategy (Kirsh, 1996). From here onwards, we will term such stable external structures that provide "cognitive congeniality" (Kirsh, 1996), *epistemic structures* (ES). The term is derived from the distinction between epistemic and pragmatic action, developed by Kirsh & Maglio (1994).

A significant chunk of the cognitive science literature on ES is from the field of Distributed Cognition (Kirsh, 1995; 1996; Hutchins 1995a; 1995b). Kirsh's work explores the structural and computational properties of such structures, how they function, and how organisms *interact* with such structures. Such run-time interaction with external structures has been used to argue the case for situated and distributed cognition, i.e. the use of the environment as a cognitive resource by the organism. In the extreme, this dependence has been used to argue against the existence of representation-based (i.e. symbolic) cognition (Brooks, 1997).

We are interested in how organisms *generate* such structures – which is the other half of the ES problem. Generation is about *internal mechanisms* that enable organisms to generate structures in the environment and use them. Currently there are no models of specific internal mechanisms that lead to the emergence of external structures. This paper

presents the first effort in this direction. Besides this focus on internal mechanisms that could drive the generation of external structures, the paper presents three novel approaches:

1) Traditionally, the ability to modify the environment to lower cognitive complexity has been considered a capacity exclusive to humans (Kirsh, 1995). The modifications animals make to their cognitive environments have been examined extensively under the rubric of signalling, but the research focus in this area has been on evolutionary models and game theory models of signals (Bradbury & Vehrencamp, 1998), and not cognitive complexity. In contrast, we consider the human and animal cases to be of a kind, and our analysis is based on the cognitive advantage provided by these structures. We seek to develop an integrated and evolutionarily plausible cognitive model of how epistemic structures arise, where the underlying mechanism is similar in both human and non-human cases. In this view, the distinction between generation by humans and generation by non-human organisms is one of complexity, and not of mechanism (for details, see Chandrasekharan, 2005).

2) The exploitation of external structures has been used to argue that the mind "extends" into the environment (Clark & Chalmers, 1998). In a similar vein, but more conservatively, Hutchins (1995a, 1995b) has argued that the study of external structures can provide insight into the development and nature of internal representations, and cognition itself. In the second half of the paper, we develop this claim beyond the descriptive level of interaction used by distributed cognition, and present an implemented

model of how *internal* traces of the world could originate in reactive agents (agents who can only sense and act, they do no internal processing) *within lifetime*, using the same underlying process that allow organisms to generate external structures to reduce cognitive complexity. This model integrates generation of external and internal structures under a common mechanism, and thus provides a clearer picture of how cognition could extend out into the world. Since this implementation is based on reactive agents interacting with an environment, and they develop the ability to store useful internal traces of the environment, this model also integrates the situated cognition position with the symbolic cognition position, by showing how useful internal races of the world could arise out of situated activity. The model also presents a number of interesting characteristics of such internally stored traces of the world.

3) The agents in our model develop internal traces of the world entirely out of actions, and any representational content the traces possess consists of action information. Such action-oriented content offers the possibility of "enaction" or "Simulation", which is not a possibility in traditional conceptions of internal traces of the world. This view of the origin of internal content from actions provides support and evolutionary plausibility for the Simulation theory of cognition, which argues that cognition involves a form of 'virtual enaction' (Metzinger & Gallese, 2003; Svenson & Ziemke, 2004). Further, our implementation offers a rudimentary model of the character of such 'simulatable' internal traces of the world and how they could originate, thus integrating Simulation models with situated cognition models.

The paper is organised into three sections. Section 1 considers the generation of epistemic structures. It provides a model and implementation of how such external structures are generated by non-human organisms. Section 2 extends this model to the generation of internal traces of the world. It develops an account and implementation of how organisms could generate such internal structures to lower cognitive load. Section 3 addresses the theoretical implications of this second (extended) model for two wider issues in cognition (representation, and the Simulation/enaction model of the mind).

To make the paper accessible to a wide audience, we have simplified the description of our implementations in the main text. Most of the implementation details are provided in the endnotes. For those interested in further details, the code for the two implementations (in Python) is publicly available. Please see endnote 7 for the URL for the files.

1. The Origin of Epistemic Structures

Epistemic structures can be classified into three types, based on whom they are generated for (examples of each in brackets).

1. Structures generated for oneself (Cache marking, bookmarks)
2. Structures generated for oneself and others (Pheromones, color codes)
3. Structures generated exclusively for others (Warning smells, badges)

There are other ways to classify epistemic structures (for instance by function: structures for mating, foraging etc.), but the above classification is more suited to the objective of

this paper, which is to understand the mechanisms that lead up to the generation of such structures. The above classification provides a good framework to develop progressive models of epistemic structure generation – moving from structures generated for oneself to structures generated exclusively for others. The classification also captures the entire space of epistemic structures generated.

A central feature of such structures is their task-specificity (more broadly, function/goal-orientedness). To illustrate this concept, consider the following example. Think of a major soccer match in a large city, and thousands of fans arriving in the city to watch (the example is based loosely on the Paris World Cup Series). The organizers put up large soccer balls on the streets and junctions leading up to the venue. Fans would then simply follow the balls to the game venue. Obviously, the ball reduces the fans' cognitive load, but how? To see how, we have to examine the condition where big soccer balls don't exist to guide the fans.

Imagine a soccer fan walking from her hotel to the game venue. She makes iterated queries to the world to find out her world state (What street is this? Which direction am I going?), and then does some internal processing on the information gained through the queries. After every few set of iterated queries and internal processing, she updates her world state (I'm at point X) and internal state (now searching for point Y), and this process continues until she reaches her destination.

What changes when the ball is put up? The existence of the big soccer ball cuts out the iterated queries and internal processing. These are replaced by a single query for the ball, and its confirmation. The agent just queries for the ball, and once a confirmation of its presence comes in, she updates her internal state to look for the next ball. The ball allows the agent to perform in a reactive, or almost-reactive mode, i.e., move from perception to action directly. The key advantage is that almost no (or significantly less) inference or search is required, compared to the case where the ball does not exist.

This happens because the ball is a *task-specific* structure; it exists to direct soccer fans to the game venue. Other structures, like street names and landmarks in a city, are function-neutral or task-neutral structures. The fans have to access these task-neutral structures and synthesize them to get the task-specific output they want. Once the widely visible ball, a task-specific structure, exists in the world, they can use this structure directly, and cut out all the synthesizing. (How the soccer fans manage to discover the ball's task-specificity is a separate and relevant issue, see Chandrasekharan, 2005 for an account). In graph theoretic models (see Kirsh, 1996) such task-specific structures work by "collapsing" longer paths in a task-environment. Task-specificity is a common property of all epistemic structures found in nature, including pheromones and markers.

1.1 The Tiredness Model of ES Generation

How are such task-specific structures that lower cognitive complexity generated? In this paper we consider the case of non-human organisms like ants, wood mice and red foxes.

(See Chandrasekharan, 2005 for an account of the human case). We will first describe our model in high-level terms, and then develop the computational model.

We make two assumptions:

1. Organisms sometimes generate random structures in the environment (pheromones, urine, leaf piles) as part of their everyday activity.
2. Organisms can track their physical or cognitive effort (i.e., they get 'tired'), and they have a bias to reduce physical or cognitive effort. We will use the terms 'cognitive load' and 'energy load' interchangeably to indicate this effort (see endnotes 1 and 2 for an explanation).

Now, some of the randomly generated structures are encountered while executing tasks like foraging and cache retrieval. In some random cases, actions executed during these encounters make the task easier for the organisms (following pheromones reduces travel time, avoiding urine makes cache retrieval faster, avoiding leaf-piles reduce foraging effort). That is, these random structures shorten paths in the task environment in some random cases. Given the postulated bias to avoid tiredness[1], these paths get preference, and they are reinforced. Since more structure generation leads to more of these paths, structure generation behavior is also reinforced.

This high-level model gives us the outline for building a computational model, where artificial agents display the ability to learn to systematically generate such cognitively[2] 'congenial' (Kirsh, 1996) structures in their environment.

1.2 The Implementation

To test and investigate the above model of epistemic structure generation, a multi-agent simulation was implemented. Multi-agent simulations typically consist of a number of agents (usually reactive agents) that have the ability to move around in an environment. The agents can sense some events and objects in the environment, and execute some actions that change the state of the environment. Such simulations are an effective way of understanding complex and dynamic agent-environment relationships, and have been used extensively to understand a diverse range of phenomena, including honey-bee nest architectures (Camazine, 1991), ant foraging (Bonabeau et al, 1999), evolution of language (Kirby, 2002), human mate-choice (Todd & Miller, 1999), and the development of markets (Tesfatsion, 2002).

The task we chose is analogous to foraging behavior, i.e. navigating from a home location to a target location and back again. Our environment consisted of a 30x30 toroidal (doughnut-shaped) grid-world, with one 3x3 square patch representing the agent's home, and another representing the target. This 'target' can be thought of as a food source, to fit with our analogy to foraging behavior.

1.2.1 Agent Actions

At any given time, an agent can do one of five possible actions. The first and most basic of these is 'moving randomly'. This consists of going straight forward, or turning to the left or right by 45 degrees and then going forward. The agent does not pick which of these three possibilities occurs (there is a 1/3 chance of each).

In deciding the actions available to the agent, we needed to postulate some basic facilities within each agent. For our case, we felt it was reasonable to assume that the agents could distinguish between their home and their target, as we were interested in the origins of structure-generation behaviour and not landmark-identification behaviour. However, this ability to distinguish target and home was provided in a limited fashion, using two more actions. These were exactly like the first action, but instead of moving randomly, the agent could move towards whichever square was sensed to be the most 'home-like' (or the most 'target-like'). Initially, the only things in the environment that are 'home-like' or 'target-like' are the home and the target themselves.

One way to think about these actions is to consider the pheromone-following ability of ants. Common models of ant foraging (e.g. Bonabeau et al, 1999) postulate the automatic release of two pheromones: a 'home' pheromone and a 'food' pheromone. The ants go towards the 'home' pheromone when they are searching for their home, and they go towards the 'food' pheromone when foraging for food. This exactly matches these two actions in our agents. The 'home' pheromone would be an example of a 'home-like' structure in the ant environment.

The fourth and fifth possible actions provide for the ability to generate these 'home-like' and 'target-like' structures. In the standard ant models, this could be thought of as the releasing of pheromones. However, our simulation has an important and very key distinction. Here, this ability to modify the environment is something the agents can do *instead* of moving around. That is, this generation process requires time and effort. The best way to envisage this is to think of an action that a creature might do which inadvertently modifies its environment in some way. Examples include standing in one spot and perspiring, or urinating, or rubbing up against a tree. These are all actions which modify the environment in ways that might have some future effect, but do not provide any sort of immediate reward for the agent. Kirsh (1996) terms these 'task-external actions'.

It must be stressed here that this implementation does not presume any sort of long-term planning on the part of the agents. We simply specified a collection of actions available to them, and they will choose these actions in a purely reactive manner (i.e., based entirely on their current sensory state). They do not initially have any sort of association between the action of making 'home-like' structures and the action of moving towards 'home-like' things. Any such association must be learned (either via evolution, or via some other learning rule). It may also be noted that our 'actions' are considered at a slightly higher level than is common in agent models. Our agents are not reacting by 'turning left' or 'going forward'; they are reacting by 'following target-like things' or 'moving randomly'.

Also, our agents are not designed to form structures automatically as they wander around (as is the case in standard ant models). In our simulation, a creature must expend extra effort to generate these structures in the world. An agent that does this will be efficient only if the effort spent in generating these structures is more than compensated for by the effort saved in having them in the world. Moreover, these are not permanent structures. The agents' world is dynamic and the structures do not persist forever. The 'home-likeness' or 'target-likeness' of the grid squares decrease exponentially over time. Furthermore, these structures also spread out over time. A 'home-like' square will make its neighboring squares slightly more 'home-like'. This can be considered similar to ant pheromones dispersing and evaporating, or leaf/twig piles being knocked over and blown around by wind or other passing creatures.

1.2.2 Agent Sensing

Since our agents are reactive creatures and thus do no long-term planning, they require a reasonably rich set of sensors. Our agents had four sensors, two external and two internal, to detect their current situation. The two external sensors sense how 'home-like' and how 'target-like' the current location is (digitized to 4 different levels). The internal sensors are two simple bits of memory. One indicates whether the agent has been to the target yet, and the other indicates how long it has been since the agent generated a structure in its environment (up to a maximum of 5 time units). This is all that the agents can use to determine which action to perform. This configuration gives each agent 192 (4 x 4 x 6 x 2) possible different sensory states.

1.3 The Learning Rules

For a purely reactive agent, we needed some way of determining which action the agent will perform in each of these 192 states. We investigated two different methods for matching sensory states to actions: a Genetic Algorithm, and Q-Learning.

1.3.1 Stage A: The Genetic Algorithm

Before determining whether the agents could learn to drop 'pheromones' to decrease their tiredness within their lifetimes, we first decided to check that it was possible to learn this task across generations, i.e. on an evolutionary time scale. For this, we used a genetic algorithm to evolve foraging behaviour in the agents (see endnote 3 for details of the genetic algorithm). A genetic algorithm is a general-purpose, but usually very slow, method of finding good solutions to a problem. In this case, no learning at all would occur during the lifetime of one agent; each agent would be locked into a particular sense-response pattern. The agents would thus always perform the same task for a particular state. For example, the agents might be defined to always drop one kind of pheromone whenever they are on a very 'home-like', but not 'target-like' square, if they are searching for food and if it has been three time steps since they dropped any pheromone.

The agents start out with completely random settings for what to do in each sensory state. Given this fact, the agents will start off performing very poorly. To improve their behaviour, the genetic algorithm makes slight modifications (random 'mutations') to the

set of rules. These 'mutations' change the behaviour in unpredictable ways. The changed agents are then simulated to discover how well they do. Over time, the agents 'evolve' to become better and better at their foraging task.

Our definition of tiredness was simply the amount of time required to travel from the home to the target and back. 10 agents tried to forage at the same time. Initially, the agents behaved randomly. Starting at the 'home', they would wander about and might, by chance, find the target and then, if they were very lucky, their home. Indeed, most agents did not find the target and make it back within the 1000 time steps. On average, we found that each agent was completing 0.07 foraging trips every 100 time steps. After a few hundred generations, the agents were soon completing an average of 1.9 trips in that same period of time. In other words, the agents were able to learn to systematically make use of their ability to sense and generate structures in the world, on an evolutionary time scale. Furthermore, this systematic adding of structures to the world provided a very large tiredness advantage over completely random behavior.

This result confirmed that it is possible for agents to learn to systematically generate and use structures in the world on an evolutionary time scale. It also showed that we had not chosen an impossible task for the agents to learn. However, for our purposes, we were much more interested in an individual agent learning to generate epistemic structures within that agent's lifetime. To investigate this, we turned to the Q-Learning algorithm.

1.3.2 Stage B: Q-Learning

The heart of our investigation was to determine whether a simple, general learning algorithm would allow our agents to discover and make use of the strategy of systematically adding structures to the world within their lifetimes. The only feedback to the learning mechanism (postulated by the 'tiredness' model of ES generation) is an indication of exertion or effort. Our analysis indicated that the delayed-reinforcement learning rule known as Q-Learning (Watkins, 1989) would be the simplest automatic method that was likely to be able to perform this task.

The Q-Learning algorithm is a probabilistic learning rule that maps states in the world [s] to possible actions [a], using feedback from rewards and punishments. That is, it learns what actions in any given situation are likely to lead to the maximum long-term reward (or minimum long-term punishment). Given our assumption that the only feedback is 'tiredness', we give our agents a punishment (a reward of –1) whenever they perform an action, and a reward of 0 whenever they successfully complete a trip. Thus, to minimize their long-term punishment, they would need to travel from their home to the target and back as quickly as possible. Ideally, the agents would learn that generating epistemic structures can make their trips faster (by allowing them to find the target and the home more easily), and so would learn to do so.

The method utilized by the Q-Learning algorithm to achieve this result is structurally simple, but complex in practice. The idea is to estimate future rewards based on past

experience. As an initial (approximate) example, consider an agent in state S_1. It may have learned from previous situations that doing action A_1 in state S_1 tends to lead to a reward of R_1. It may also have learned that doing action A_2 in state S_1 leads to reward R_2. The system could then compare R_1 and R_2 to choose which action it should perform. This is the basic idea behind Q-Learning, with the vital exception that instead of R_1 and R_2, it uses Q_1 and Q_2, which are the *predicted long-term rewards*, not the simple one-moment-from-now rewards.

In other words, the agent chooses an action based on this Q value, which is an approximated 'projection' of future reward, based on previous values from experience. Importantly, this projection is not calculated by explicitly running possible action chains for every possible sequence of actions into the future and compiling their rewards. The projection is calculated using a function (the Q function) learned in real-time, derived from previously executed actions, *where every action in the world is considered a 'test' action*. Once derived, the use of this Q function can be considered as *implicitly running* possible future actions, across time. This is because every use of Q involves an implicit projection into the future (For details of this projection, see section 3.2.2, which provides a more graphical description of Q-Learning. See also Stewart & Chandrasekharan, 2005).

Using the Q-Learning algorithm, we ran 10 agents for 1000 time steps (for implementation details of Q-Learning, see endnote 4). To indicate 'tiredness', we gave them a reinforcement value of -1 while 'foraging' (indicating a constant 'punishment' for expending any effort). When they returned home after finding the target, they were given

a reinforcement of 0, and they were then sent back out again for another trip. Each agent independently used the Q-Learning algorithm, and there was no communication among the agents. The following figure presents an outline of the learning model's architecture.

Figure 1 about here

Result: Figure 2 shows the model at different stages of learning. The dark line in figure 3 below shows the results averaged over 100 separate trials. We can clearly see that the agents are improving over time (i.e., they are making more trips, i.e., spending less time to perform their foraging task).

Figure 2 about here

1.3.3 Confirming the Role of ES

Although we have observed improvement over time, we still need to show that it is the agents' ability to systematically add structures to the world that is causing this effect. To prove this, we re-ran the experiment, this time removing the agents' ability to generate structures in the world. No other changes were made.

Result: We found that when the agents were unable to generate structures in the world, Q-Learning did not provide as much improvement. This result is shown in the lighter line in Figure 3. There is still a small improvement given by Q-Learning, but we are able to conclude that the significant improvement seen in the previous experiment is due to the

agents' ability to modify their environment. Q-Learning also did not provide significant improvement if the agents were only able to generate one type of structure, or if any of the agent's sensors were removed.

We can also see from Figure 3 that having these extra actions available does incur some cost in the early stages. Initially, the agents perform slightly worse. However, the advantage of being able to form epistemic structures quickly improves the agents' performance. By the end of the simulation, agents require only around 150 time steps to make a complete trip (a foraging rate of 0.66 trips in 100 time steps). This is twice as quick as agents without the structure-forming ability.

When we analyzed the actions of the agents, we found that they actually spent 58% of their time generating structures. This is striking, since time spent generating these structures means less time for wandering around trying to find the target or their home. Table 1 gives the breakdown of how time was allocated to different actions. The data indicates that epistemic structure generation allowed the agents to go from spending 300 time steps down to 150 time steps to complete their foraging task, *even though over half of those 150 time steps are spent standing still*. This happens because the Q-Learning algorithm learns that the long-term punishment (tiredness) resulting from generating these structures is lower than the tiredness resulting from not generating these structures. There is clearly a very large efficiency advantage in generating and making use of these markers in the world, and the Q-Learning algorithm is able to discover this *without explicit long-term planning*.

Figure 3 about here

Table 1 about here

There are many Reinforcement Learning algorithms available other than Q-Learning, and any one of them could be used in this type of model. All these algorithms learn in a similar way, but with rather different details. So the resulting high-level behavior may be different. Our ongoing research explores the capabilities of these various methods.

1.4 Discussion

The Q-Learning system is a concrete proof-of-concept implementation of our model: a simple learning mechanism that allows agents with purely reactive behavior to systematically add structures to the world to lower search.

The 'tiredness'-based learning model implemented in this simulation can explain the generation of task-specific structure in cases 1 and 2 (structures for oneself, structures for oneself & others). Case 2 (structures generated for oneself & others) is explained by appealing to the similarity of systems – if a structure provides congeniality for me, it will provide congeniality for other systems like me. The agents in our simulation ended up forming structures that were useful for everyone, even though they were just concerned about reducing their own tiredness. This was possible only because the agents were similar to one another. This is comparable to how paths are formed in fields: one person

cuts across the field to reduce his physical effort, others, sharing the same system and wanting to reduce their effort, find the route optimal. As more people follow the route, a stable path is formed. The evolution of such case 2 structures have been explored by work in stigmergy (Susi & Ziemke, 2001).

For case 3, (structures generated exclusively for others), the 'tiredness' model explains only some cases. For instance, it could explain the generation of warning smells and colors exclusively for others, because the effect of such structures could be formulated in terms of tiredness (the release of some random chemical cautions some random predators; this reduces the number of fleeing responses the organism makes, reducing tiredness; this advantage, when fed back, reinforces the initial random generation).

However, the model, as it stands, cannot explain the generation of structures such as the male bower bird's bower (a mating signal that help female birds make better mating decisions), as the bowers do not seem to provide any tiredness benefit for the generator. However, a similar learning system, using another reinforcement factor (say dopamine) along with tiredness, could explain this case. In this scenario, the reward for the generator is from dopamine (i.e. a mating opportunity), while the reward for the user would be a reduction in tiredness. A closely related study (Schultz, 1992) is reported by Braver & Cohen (2000), where dopamine responds initially to a rewarding event, and with training this response "migrates" to predictive cues. This behaviour, where learning can chain backwards in time to identify successively earlier predicators of reward, has been modeled using a temporal difference (TD) learning algorithm (similar to Q Learning) by

Montague, Dayan and Sejnowski (1996). Such a dopamine-based model may help explain the mechanism underlying the generation of structures exclusively for others, particularly by humans.

It is worth noting that our model also presents a novel simulation of ant behaviour. The closest existing models are those in Bonabeau et al (1999), which use the 'home-pheromone' and the 'food-pheromone'. This is in contrast to such models as (Nakamura & Kurumatani, 1996), where a land-based and an airborne pheromone are used, or any models of the Cataglyphis species of ant, which uses a complex landmark-navigation scheme which allows it to return directly to the nest (Miller & Wehner, 1988). That said, all of these other models assume both that pheromones are continually being released while the ant forages, and that there is no learning happening during the foraging behaviour. Our Q-Learning model does not make either of these assumptions.

We were unable to find references indicating that real ants (or other creatures) might, in fact, learn to use pheromones (or other epistemic structures, but see the pigeon example below) within their lifetime, or any research that indicates that the effort required to produce these pheromones might interfere with foraging behavior. So while our model may not be ideal for specifically understanding ants, it does illustrate a mechanism that could lead to the evolution of epistemic structures within the lifetime of an individual. Our simulation thus provides a very novel result, as current biological models assume (based on experimental evidence) that such ES structure-generation behavior is mostly innate, and is based on evolutionary learning. Interestingly, recent research shows that

homing pigeons learn within their lifetimes to use human-generated environment structure in a similar fashion to reduce cognitive load. They follow highways and railways systematically to reach their destination, even following exits (Guilford et al., 2004). A similar landmark-based navigation system has also been reported in bees (Gould, 1990).

An interesting aspect of the within lifetime learning model is that it scales well to human situations (see Chandrasekharan, 2005), and could be used to explain the developmental origins of structure generation behaviour in humans. It can also be extended to account for the generation of internal structures that lower cognitive load. The following section presents this model.

2. The Generation of Internal Structures

Section 1 established that a simple, plausible, and efficient learning mechanism could enable reactive agents to add structures to their environment within their lifetime. The agents could perform actions that changed aspects of their environment in ways that could be sensed by that agent later, and the effects of reinforcement caused some structures to be systematically created in certain situations. We have established that this works with a simple foraging task and the basic Q-Learning algorithm.

The success in implementing this within-lifetime learning model led us to ponder a new question: What about actions that change the *agent*, rather than the *environment*? That is, can the same sort of learning mechanism and actions lead to the systematic generation

of structures in the agent's *mind*, rather than in the agent's environment? This seems to be both a natural extension of our work on external structures, and, more importantly, this seems to be a novel way to model *the origin of internal representations* in rudimentary agents within their lifetime. If an agent can learn this strategy of generating internal structures to lower tiredness, then it can *choose* to remember particular things in particular ways to benefit it in the long term, just as our earlier experiments showed that it was possible to choose to drop pheromones in useful ways.

The goal was to develop a task and a set of actions that change *internal states,* so that an agent using the same Q-Learning approach can learn to remember some states of the world and use this information to better execute a task. For consistency, we decided to use the same foraging-style task used in the previous experiment. To do this, an agent needs two things: a way of remembering where it has come from (or, equivalently, is going next), and a way of knowing how to get there. In our previous experiment, we were giving the agent the first capability (the *internal state sensors,* indicating whether they were looking for the target or looking for home), and getting it to learn the second capability. We now wanted to change the task so that we will give it the second capability and have it learn the first. That is, the agent will not be given the knowledge of whether it is supposed to be looking for its home or looking for the target. It must learn to keep track of that information on its own, via actions that change *internal states*.

2.1 Initial Failures

All our initial attempts to get agents to learn this behaviour failed, and it is worth noting why. We began with a very simple kind of memory: a single value that the agent could set to be either zero or one. In theory, this would be sufficient: the agent could learn to set it to a zero whenever it had found it home, and set it to a one when it had found the target, and whenever it was wandering around in between it could look at the value stored there to let it know which way it was going. We tried a large number of variations on this idea, but not one was successful. The conclusion from an analysis of these failures was that this sort of memory was too fragile: the agent could make one random mistake (changing the value at the wrong time, for example), and the system would become useless.

2.2 Mapping Internal and External Structures

We then took a careful look at the differences between the successful *external* actions/sensors in the previous experiments, and the unsuccessful *internal* actions/sensors. After all, the learning mechanism is the same in both experiments, and the internal/external distinction is not one that should affect Q-Learning. Whether the structures are generated in the world or in the head, they should still be able to be used. So what is different about these two forms of structure generation?

Two key differences were apparent, and they form our hypothesis about what is needed for this type of learning to work:

1. The structure generation process should be *gradual*. Dropping pheromones makes a small change to the pheromone level at a particular location in space. This allows for the smoothing out of errors. This also allows the learning process to *converge* to a solution, instead of learning from discrete bits and pieces of information.

2. Structure generation should be *context-specific*. When an agent is dropping pheromones, it does not have the option of dropping pheromones anywhere in its universe. It can only drop them (and sense them) where it is. The action is thus not *drop-pheromones*, but rather is *drop-pheromones-at-my-location.* Similarly, the sensors sense *pheromone-at-my-location*.

Note that these two (gradualness, context-specificity) are not canonical features of internally stored structures in classical models of internal structures (Fodor & Pylyshyn, 1997). In contrast, connectionist models do argue for internal structures with these features (Smolenksy, 1997). However, such models consider these design features as enhancing robustness, for instance through graceful degradation or multiple encoding. We find it interesting that we arrived at gradual and context-specific internal structures via a different pathway: by trying to approximate the previous (ES-generating) agent's interactions with the world. These design features arise out of the environment in our case.

2.3 A Model of Internal Structures

Given the above mapping between external and internal structures, we needed a gradual and context-specific style of memory, rather than the discrete and context-neutral mechanisms used in our initial (failed) experiments. This mechanism would be an internal equivalent to the pheromone dropping, spreading, and sensing mechanisms in our epistemic structure experiment. This means it would act as an 'internal environment' that the agent can alter, just as it altered the external environment in the first experiment.

Such a mechanism needs to have the following three capabilities:

1. It needs to be able to *store data associated with a particular context*. That is, we need to be able to give it a particular sensory state and a particular piece of data (say a 1 or a 0), and it should be able to remember this pairing. This is functionally similar to dropping pheromones of different types at a particular point in the world – the data being remembered can be thought of as being at a particular 'point' in the creature's memory.

2. It needs to be able to *recall data when in a particular context*. That is, when the agent is in a particular sensory state, it should have a sensor that indicates what value was stored in the past in this state. This is functionally similar to the sensor that indicates the level of pheromones at a particular point in the world – the value being given by this internal sensor is, in some sense, the value being stored at a particular location in the agent's memory.

3. The information needs to *spread and change gradually*. That is, data stored in one context should be available in similar contexts, and any new data being stored should cause only small changes. This is functionally similar to the spreading of pheromones in space, and the fact that dropping new pheromones makes only a small change to the amount of pheromones at that location.

This would allow a creature in state X to perform an internal action that associates a particular number with state X. Then, in the future, when it is in state X (or in another, similar state), it will be able to remember that number. Furthermore, if it later chooses to associate a different number with state X, its memory will gradually change.

How could such a memory system be implemented? Interestingly, there is a simple and well-studied mechanism that has exactly these characteristics: the standard feed-forward neural network trained by back-propagation of error. We chose such a network as our internal memory mechanism. Just as our first simulation had agents with mechanisms for dropping and sensing pheromones, in our new experiment we give the agent a mechanism for storing data into this sort of network, and a mechanism for sensing the current output of the network. This network thus *plays the same role as the external environment* in the first experiment, and provides us the basic architecture for an internal mechanism allowing an agent to form internal structures to reduce the effort required to perform tasks.

Note again that our reason for using this neural network is quite different from the traditional reasons for using a neural network (such as graceful degradation and neural plausibility). We are using a neural network because *it is the system which is most similar to the gradually changing world the agent lives in*. Note also that the neural network is not being used to represent the world. It is being used to allow the agent to represent those useful parts of the world that it cannot directly sense.

2.4 Defining the Experiment

Given the above characterisation of an internal memory store (or internal environment), we can now precisely define the new version of the foraging problem. We again have a simple grid-world, with a 'Home' and a 'Target'. Since the structures agents would generate are internal and not shared by others in an external world, we have just one agent in the simulation. At any moment, the agent can perform one of the following actions:

1. Move Randomly
2. Move Towards the Target
3. Move Towards the Home
4. Train the Internal Neural Network to associate the value 1 with the current Sensory State
5. Train the Internal Neural Network to associate the value 0 with the current Sensory State

As discussed previously, we are for the moment giving the agent the ability to simply move directly towards the target, since we are focusing on its ability to learn to remember which way it is currently supposed to be going.

The agent makes its decision on which action to perform based on its sensory information. Here, we have three sensors, which define the agent's current sensory state:

1. 'Home' Detector (1 if the agent is at its home, 0 otherwise)
2. 'Target' Detector (1 if the agent is at the target, 0 otherwise)
3. Current Memory (the output of the neural network for the current sensory state – i.e. the data currently being remembered)

The agent then uses Q-Learning to learn to perform different actions based on its current sensory state. Whenever the Q-Learning system chooses actions 4 or 5, the system uses back-propagation learning to train the internal neural network to associate a value (0 or 1) with the current sensory state. Figure 4 provides an outline of the learning system's architecture. It should be noted that there is a subtle recursion happening in this model. One of the components of the agent's sensory state is the output of the neural network (the internal environment), but that output is itself *dependent on the current sensory state*. This means that *what the agent remembers is dependent on what it is currently remembering*. This architecture is similar to the one used by Tani and Nolfi (1997). This new memory system can be seen as a sort of *Internal Environment*, as it is functionally

connected to the rest of the agent in exactly the same manner as the actual *External Environment*. (For more information on the neural network, see endnote 5.)

As before, the agent's only reward is based on the total amount of effort required to complete a trip to the target and back home. The agent's performance can then be compared to that of the same agent without the ability to perform these internal actions. This is the same approach taken in the previous experiment.

Figure4 about here

2.5 Experimental Results

Figure 5 compares the foraging performance of the agent with the ability to generate internal structures to that of an agent without this internal mechanism. As in the previous experiment, we can see that having the ability to generate internal structures results in behaviour that is initially worse, but that then improves to be consistently better than the agent without this ability. These data are an average of 3,000 runs at each setting.

Figure 5 about here

Table 2 about here

As can be seen, the agent spends only 22% of its time generating internal structures, compared to the 58% in the external case. Comparing the two cases is not entirely justified, as the internal traces and external structures serve significantly different purposes in the foraging task -- the first orients the agent, the second marks the route. But since we consider the generation of external and internal traces as equivalent *strategies*, the two cases could be compared at the strategy level. In such a comparison, the low level of structure generation in the internal case could be interpreted in two ways. One, the internal structures do not provide much advantage, so they are generated less. This option can be safely rejected, as the agent's performance in the task is significantly lower when it does not generate internal traces (see figure 5). The second, and more plausible, interpretation would be that these internal traces are more stable structures than the external structures generated in the first experiment, as they do not disperse and evaporate. So they do not need to be generated as often as the external ones to be useful, and are therefore generated less. Since generation requires energy in our model, the internal traces provide more "bang for the buck", so to speak. This could be one reason why the internal trace strategy is used more widely in nature than the external one.

The results above show that the agents were in fact able to benefit from having the ability to choose to remember particular values in particular contexts. In other words, the simple reinforcement-learning approach that worked for learning to generate external epistemic structures is also able to systematically generate and make use of internal structures. Importantly, it seems to work only when the internal structures generated are context-specific and gradually distributed.

3. Theoretical Implications

The above two simulations present an integrated proof-of-concept model of how both external structures and internal memory structures come to be used as task-specific structures, and how such structures could systematically be generated within lifetime, based just on the feedback of cognitive load via a steady 'tiredness' punishment. The two models have wide theoretical implications, but we will focus on the model of internal structures, and its implications for the following two areas of cognition.

1) Representation
2) The Simulation Model of the mind

3.1 Internal Traces as Proto-representations

The defining feature of mental representations is that they 'stand-in' for other things – they are 'about' other things (Dennett & Haugeland, 1987). Epistemic structures, as we have defined them, do not meet this criterion, as they do not stand-in for anything, and could be viewed as being used directly by organisms. However, the stored internal traces of the world in our second simulation could be considered *proto-representations*, because once the structures stabilize, they 'stand-in' for something specific in the world, namely the home location and target location. The traces are 'about' something in the world*, and they are useful because of this aboutness.* By themselves the traces are just two values (0,1). The agents store and use the traces to exploit the aboutness, as this feature helps them choose the best action in a context.

However, these internal traces are not entirely representations, because our agents cannot use the internal traces as surrogates when the actual structures *do not* exist in the world (as in the case of being able to mentally rotate an object when the object is not in the visual field, see Beer, 2003). This is one reason why we consider our internal traces *proto-representations*. Another reason why we consider the stored internal structures as proto-representations is our agents' "selective representation" of the world (Mandik & Clark, 2002), where an organism is considered to perceive and cognize a "relevant-to-my-lifestyle world, as opposed to a world-with all-its-perceptual-properties". In this view, the mental representations of organisms are highly constrained by the biological niches within which the organisms evolved.

The criticism has been raised that the model assumes and builds in some basic internal structures such as those involved in sensing, acting and learning. We consider this justified because strictly speaking, there are no reactive agents in the world – all agents have some basic internal structures. We show that given this basic ability for sensing, acting and learning, agents could develop a secondary form of representation, a structure that "stands-in" for something in the world. Moreover, our model explains what such 'primitive' representations are: they are the internal traces of the world that allow the agent to shorten paths in a task environment. Roughly, they are computation-reducing structures (and equivalently, energy-saving structures). Metaphorically, they are internal 'stepping stones' that allow organisms to efficiently negotiate the ocean of stimuli they encounter. By extension, 'aboutness', or the standing-in property of internal traces, is an

energy-saving mechanism in our model. This view is very different from traditional conceptions of representation and aboutness.

The model provides a unified account of the generation of external as well as internal structures, as the internal structures are stored using the same process as the external structures, and the structure of the internal traces is similar to the structure of the external traces. Given this same underlying mechanism, the agent can transform the world or itself, depending on task and resource conditions. The two manipulations – internal and external – are equivalent at the mechanism level. Internal changes lead to internal representations, external changes lead to the world being used directly. This integrated generation mechanism illustrates a way of storing task-specific traces inside and outside in an opportunistic fashion. The organism could exploit both together, thus 'extending' cognition out into the world (Clark & Chalmers, 1998). In this proposal, the notions of storing representations and using the world directly (the symbolic and situated views) are not at odds with each other, they are just two ways of solving the adaptation problem.

3.1.1 The Nature of Proto-representations

Our failed initial attempts and the second working model of how such structures could be generated provide some insights into the character of proto-representations. The classical notion of internal traces of the world assumes the storing of static structures, with a one-to-one relation with structures in the world (Fodor & Pylyshyn, 1997). This is the traditional symbolic representation view. Our experience shows that it is difficult for reactive agents to learn to store and make use of such static structures. Such structures do

not support incremental and contextual learning, which is the kind of learning commonly postulated in low-level organisms. If we assume that the storing of internal structures originated to support tasks, and this storing behaviour was learned by organisms, the notion of storing static structures with one-to-one relations with entities in the world would need to be revised.

To support the learning of the internal trace strategy, we will have to postulate a "process" view of representation, in place of the traditional "product" view (see Clark, 1997 for a closely related position). In the process view, elements are initially randomly stored in an internal network (which acts as an equivalent of the external environment), the agents sense these internally stored elements and act. Through an incremental process based on feedback of cognitive load, these elements then gradually get systematically stored and acquire a representational nature. Such an internal representation is not a single well-defined structure that reflects the world mirror-like, but a systematic coagulation of contexts and associated actions, spread over a network. The structure itself is just a common thread of elements running through contexts and associated actions that lower cognitive load. It is this common status that leads to the thread getting stabilized as an internal trace. Metaphorically, such an internal representation resembles the core of an active bee swarm, rather than static symbolic entities like words or pictures.

This action-driven model of representation, and the recursive and dynamic relation between the internal traces of the world and the agent's actions-in-the-world, make our implementation more than a standard neural network model. A central difference is this:

in our model a reference relation *develops* between the neural network and elements in the agent's environment. This reference relation is usually hard-coded or assumed in standard neural network models. Secondly, though we use a neural network, it is used to create an internal equivalent of the agent's *task environment*, and the agent learns to store *task-specific* internal traces of some aspects of that environment. As far as we are aware, there are no neural network models that show the origin of such *task-specific* internal traces. Most neural network models assume that a correspondence relation, albeit a distributed one, exists between the network and the world.

We would also like to emphasize here that the internal traces developed by our agent are radically different from the categorical structures evolved in recent work using evolutionary learning (See Beer, 2003; Steels, 2005). The central differences are: 1) Our agent executes a task similar to a real-life task; 2) it learns to store *task-specific* internal traces of the world 3) It learns to do this within its lifetime.

Our agent exists in a "representation-hungry" task environment (i.e. one that requires some form of representation to do the task well, see Clark and Toribio, 1994). The agent learns to represent an aspect of this environment, because such representations lower cognitive load and help the agent in executing the task with less effort. So the internal traces our agent develops are *task-specific* traces of the world, not categories. Our model considers internal traces as useful and action-driven structures, and we show that they arise because of these features. In contrast, evolutionary models of category learning presume categories that mirror the world, and it is not clear why they arise. A related

difference is that storing traces requires effort in our model, and the agent chooses to store traces because the effort involved in storing a trace is compensated for by the advantage such storing provides. Category learning models assume that learning categories are useful, and therefore category learning is not something the agent chooses to do out of many possible actions.

An interesting aspect of our model is that some elements in the environment acquired salience as the model learned to store and use internal structures. However, this is an unintended effect, as we developed our model to investigate how internal structures originate. In this modeling approach, salience of elements arises as a by-product of the strategy of generating and using internal traces of the environment. By extension, this means focusing on salient elements is not a primary process, it derives from learning to store representations. Most current models consider salience to be a primitive property, existing prior to representation. Our model provides a way of deriving salience as a sub-process of representation.

3.2 Internal Traces and the Simulation/Enaction Model

This section describes the highly debated Simulation/enaction model of cognition, and how our model of internal structures supports one form of Simulation/enaction and explains its origins. To avoid confusion with the simulation we implemented, we will use Simulation with a capital S when discussing this proposed cognitive mechanism.

In general, Simulation models of cognition propose that neural structures responsible for action and/or perception are recruited in the performance of cognitive tasks (such as language processing or observing another agent execute an action). Such a recruitment process is indicated by experiments (see Svenson & Ziemke, 2004 and Brass & Heyes, 2005 for reviews). This evidence is used to make the argument that different aspects of cognition involve a 'virtual running' of actions (Metzinger & Gallese, 2003). Of particular significance is the claim that such Simulation/enaction 'grounds' symbols and other representations, i.e. provide their content (see for instance Barsalou, 1999; 2003). The Simulation view is a rapidly developing theoretical framework in cognitive science, and is used to explain cognitive processes ranging from perception to language, reasoning and theory of mind phenomena (Metzinger & Gallese, 2003; Svenson & Ziemke, 2004).

The cognitive mechanism of Simulation is considered to involve "re-enactments of states in modality-specific systems" (Barsalou et al., 2003). As against non-Simulation models, which involve "redescriptions of states in amodal representational languages" (Barsalou, et al., 2003). The central distinction is between 'reenactment' of actions and 'redescription' using symbols. Simulation is considered to involve enactment or 'acting out' an experience or action to cognize a state, while non-Simulation is considered to involve (just) retrieval and manipulation of descriptive symbols, as in doing logic or arithmetic.

A crude example to illustrate the two processes would be two ways of remembering an accident. In the first case of remembering, the event is "acted out" in the mind, and

results in bodily states associated with the event, like shaking and crying. The other way to remember the event would be as images, without any acting out of the event, and therefore without the associated body states. Since the former involves acting out, it leads to changes in the perception and action modules of the brain associated with the actual experience of the event, so it is modality-specific (*modal approach*, in Barsalou's terminology). The latter does not involve acting out of the memory, just a retrieval (and/or manipulation) of stored images. This mechanism thus represents an *amodal approach*, in Barsalou's terminology. The following figure captures the distinction using traditional modules used in cognitive psychology.

Figure 6 about here

There are two major kinds of Simulation/enaction identified in the literature. The two are closely connected, but we will treat them as separate for the purposes of our discussion. The first involves enacting or 'running' actions 'virtually' while performing cognitive tasks like processing verbs. That is, brain areas that are involved when actually doing the action associated with the verb (say chewing) are implicitly activated while processing a representation associated with that action (the verb *chew*). Such implicit activation of action areas also happens when an agent observes another agent performing an action (Brass & Heyes, 2005). We will term this kind of virtual enaction Simulation-R, for Simulation linked to representations.

Evidence in support of this Simulation mechanism comes from recent work in

neuroscience, which shows that action areas are activated while observing (i.e. representing) an action, and also during linguistic processing. Gallese et al (2002) reports that when we *observe* goal-related behaviors executed by others (with effectors as different as the mouth, the hand, or the foot) the same cortical sectors are activated as when we *perform* the same actions. Whenever we look at someone performing an action, in addition to the activation of various visual areas, there is a concurrent activation of the motor circuits that are recruited when we ourselves perform that action. We do not overtly reproduce the observed action, but our motor system acts as if we were executing the same action we are observing. This imitation effect exists in monkeys as well, and has been replicated across a series of studies (see Metzinger & Gallese, 2003; Heyes et al, 2005).

A similar process of Simulation of actions linked to representations has recently been demonstrated in language understanding. Bergen et al (2004), reports an imaging study where subjects performed a lexical decision task with verbs referring to actions involving the mouth (like *chew*), leg (like *kick*) or hand (like *grab*), areas of motor cortex responsible for mouth/leg/hand motion displayed more activation, respectively. It has also been shown that passive listening to sentences describing mouth/leg/hand motions activates different parts of pre-motor cortex.

The second kind of Simulation discussed in the literature is more complex, and involves predicting another agent's behaviour by virtually enacting the other agent's system states using one's own system as a proxy. This notion of Simulation is mostly found in the

theory of mind literature (see Nichols, Stich, Leslie & Klein, 1996). A closely related notion of Simulation is the virtual enaction of another agent's actions across time using one's own system, and then 'mutating' these actions to generate alternatives to reality. This notion of Simulation is found in counterfactual thinking literature (see Kahneman and Tversky, 1982). At a rudimentary level, such Simulation of system states could also be used to test alternatives to *one's own* current state (for instance, what would be my state at time T if I do action X?). We term this type of virtual enaction to predict future system states (others' or one's own) Simulation-S, for Simulation of system states.

There is only indicative evidence that this type of system-level enaction can allow agents to judge other agents', or one's own, system states. Svenson & Ziemke (2004) review three sources of evidence supporting an equivalence between an action and its Simulation: mental chronometry, autonomic responses and neuroimaging experiments. Mental chronometry experiments show that the time to mentally execute actions closely corresponds to the time it takes to actually perform them. Autonomous response experiments show that responses beyond voluntary control (like heart and respiratory rates) are activated by motor imagery, to an extent proportional to that of actually performing the action, and as a function of mental and actual effort. Neuroimaging experiments show that similar brain areas are activated during action and motor imagery. Beside this evidence that supports action-Simulation equivalence at the system level, there is a whole host of theoretical arguments that support simulating of other agents' system states to predict their behaviour (see Nichols, Stich, Leslie & Klein, 1996 for a

review). The literature on motor imitation also indicates that action observation could lead to a judgement of another agent's system state (Brass & Heyes, 2005).

While these two Simulation mechanisms (Simulation-R, Simulation-S) are used to explain a range of cognitive phenomena, two aspects of the Simulation idea remain unclear.

1 What is the nature of internal structures that support such virtual enaction?
2 How do such enactable internal structures originate?

We argue below that our model of stored internal traces provides tentative answers to both these questions, and thus provides an evolutionary basis to the Simulation model.

3.2.1 Simulatable Content

We will begin our discussion with Simulation-R, the idea that while processing a given internal representation (like the verb *chew*), brain systems associated with performing actions related to that representation (like the action of chewing) are also activated.

Our agent in the second experiment develops internal traces of the world using a feedback system based on actions and the cognitive load associated with actions. Values are initially randomly stored in an internal neural network environment, and then they are learned to be systematically stored, based on the feedback of cognitive load. As observed in the section on representation, the systematically stored values are tightly coupled to

actions – they are nothing but a thread that links actions that lower cognitive load. The systematically stored elements thus contain action information. That is, if zero is stored at target always, that means storing zero, sensing zero, and executing the action associated with zero lowered cognitive load. The sensing of the zero is thus not just a sensing of the zero, but a sensing of the actions and cognitive load associated with zero. And these action components are implicitly activated when zero is sensed.

At a high level, it could be argued that such a proto-representation emerging out of actions encapsulates information about actions and cognitive load, because there is nothing else contributing to such an internal trace. This means such a representation *supports* the Simulation of actions related to the representation, because the representation is a task-specific structure that emerges out of actions and feedback based on actions. This possibility for Simulation does not exist if representations are considered as static structures learned in a mirror-like fashion, with no link to actions or system states like cognitive load. For instance, if the verb *chew* is just 'captured' and stored (in either word-like or image-like fashion), it is hard to see how (and why) processing *chew* leads to Simulation of chewing. The same applies to a stand-alone neural network that learns to categorise inputs about an action as 'chew'. In contrast, task-specific internal structures are stored by agents acting in an environment. They arise out of actions and system states, and they therefore naturally support virtual enaction of those actions.

The above is a high-level view. To get a more detailed sense of the link between Simulation and the internal trace generation process, we have to examine the nature of

the Q-learning algorithm. One way to think of Q-Learning is to think of 'pretend play' by chess players, where they 'try out' potential moves. The organism 'tests' the environment with different potential actions to see what 'reward' that particular environment provides for that particular action. But, for both the chess player and the Q-Learning system, it is *not* the immediate reward for that action which is important. Instead, it is the *long-term* reward (Q) that is important. Using the Q function is equivalent to a chess player who tries out the move of taking a knight with his queen, and then looks at the new board position and gets the feeling of 'that looks dangerous – I better not do that'. Importantly, this Q function is constantly being updated by the results of all of an agent's actions in the world. Indeed, Q-Learning systems tend to 'try out' exploratory actions to gather information about what rewards will be in unknown situations[6].

Furthermore, like the chess player, the Q-learning algorithm only 'enacts' actions one step ahead, but through the use of the Q-function, its evaluation of how good that step is includes the whole future set of actions, because the Q-function approximates the possible outcome of an entire range of state-action combinations. Instead of developing an estimate of rewards for a single action, the Q function can be thought of as 'perturbing' the agent-environment system, and then developing an estimate of the reward structure of these 'perturbations' as they propagate. This means it can look ahead (i.e. test run) only one step, but the output of that test-run provides an estimate of how the system as a whole would perform many steps into the future, and the reward structure after that time. Once the Q function is developed, the agent still technically looks ahead only one step, but it can be considered to implicitly run many states ahead.

It can be seen from the above description that the Q-learning algorithm is doing a rudimentary form of Simulation-S – it is evaluating possible alternative system states by enacting them using the agent's own system. While learning, the agent is *simulating itself and its own interactions with the world*. In our second experiment, this means the agent is able to simulate itself, its interactions with the world, *and its own modifications of its own memory*. It is this Simulation-S that allows the system to learn to generate internal structures.

This means we assume Simulation-S, and a simple version of it is built into our model. The proto-representations are a product of this basic Simulation-S process. However, this process illustrates something important: not only do the proto-representations in our model implicitly contain action information (as they arise out of actions that lower cognitive load), but the Q-learning system also virtually 'enacts' these actions to judge cognitive load. This is because the internal structures are 'test-run' for their reward structure. So the proto-representations in our model not only *support* enaction linked to representations, *but also provide a working model of this enaction process*.

However, note that the above explicit enaction process is executed primarily when the system learns, and not when the proto-representations are used by the agent (to choose between target and home). Any enaction that happens post-learning is only implicit, i.e. only in the sense that the proto-representations are tightly coupled to actions and action-related information, and these actions and information are activated when the

representations are sensed. This implicit enaction may be similar to Simulation, in which case the Simulation mechanism arises out of learning. A similar position is put forward by the Associative Sequence Learning model of imitation (see Brass & Heyes, 2005).

Given this relation between learning and simulation, one possible way of interpreting the role of Simulation-R in 'grounding' content could be as follows: when Simulation-R (say enacting chewing while processing the word *chew*) happens in a system, it may not be grounding the content of *chew* directly. The enaction could be part of the learning process (i.e. the process that led to the storing and use of the word *chew*), *which is what grounds the content of the word*, by activating the contexts, actions and environmental conditions linked to that trace. In this interpretation, the learning process is always running in the background, as agents in dynamic environments cannot afford to stop learning. The Simulation mechanism is a way of 'linking' to this learning process.

The constant learning process is particularly true in our model, where the proto-representations, actions and cognitive load are tightly coupled. For instance, some changes in the environment could make the stored structures useless for our agent, and the agent will then have to reconfigure the link between actions and the stored structures in the internal network. If the learning system is activated when the trace is accessed, it will allow the agent to use the most current relation(s) between the trace, the world and the agent's system. In this view, if Simulation plays any role in the 'grounding' of representations, that role comes from the learning mechanism, which links internally

stored structures with 1) entities in the world, 2) environmental conditions, and 3) the agent's task and biological needs.

A related implication of this model of enaction of traces is that Simulation-S is more basic than Simulation-R, as the former type of Simulation leads to internal structures that support the latter type of Simulation. This could mean that at least part of the enaction that happens during Simulation-R is related to system states that led to the learning of that representation. In this interpretation, if Simulation is considered to 'ground' representations (i.e. provide their content), at least part of that content relates to system states, particularly cognitive load. (This ties in very well with our claim in section 3.1 that aboutness of internal traces is partly a mechanism to reduce cognitive/energy load.) The task-specific nature of internal traces plays a central role in this view, as the Simulation capability of traces arises out of action-driven learning linked to tasks. This means task-specificity and action-driven learning of internal traces would need to be central components of any project that seeks to use Simulation to ground representational content.

Extensions

We have presented an integrated model of the origin of epistemic structures and proto-representations, and examined some of the major theoretical implications of our model of the origin of proto-representations. Currently we are implementing the two simulations using other learning algorithms such as Sarsa and Actor-Critic methods. One planned extension is to develop an agent that has the ability to add both internal and external

structures, and then to vary the environment and task to see which structures (internal, external, combinations) originate in which conditions. An implementation of the internal structure model using Lego robots is being considered as well. We are also exploring the connections between the internal trace model and teleological approaches to semantics (see Millikan, 1994).

The model of the generation of epistemic structures presents equally interesting possibilities, particularly in studying the role of signs in higher-level cognition. One area we are investigating is the role played by signs in the computation of trust (see Bacharach & Gambetta, 2003), and whether such signs could be modeled as task-specific structures. In particular, we are interested in modeling how such structures could lose their epistemic value over time, i.e. lose their signaling character. This could be modeled as the gradual loss of task-specificity. A related project in humans is to study the origin of epistemic structures using a virtual reality game similar to the one recently used by Galantucci (2005) to examine the emergence of signing systems. The notion of epistemic structures could also be used to model the emergence of "epistemic things" in research laboratories (see Rheinberger, 1997), particularly new stains and traces in cellular and molecular biology. We are also exploring the connections between our model of epistemic structure generation and innovation in animals, in particular how organisms can use the tiredness mechanism to choose between innovation and imitation depending on environment conditions.

Endnotes

1. The term 'Tiredness' in the high-level model indicates the "felt" quality of the feedback in organisms, which allows tracking of cost using affect, i.e. without using a separate computational module that tracks cost.
2. The distinction between physical and cognitive congeniality is quite thin at the level of lower-level organisms. Avoiding cognitive effort usually means avoidance of search, at best this can be viewed as indirect physical congeniality. Avoiding physical effort is more direct, as in the case of pulling a grain from the side, instead of the front.
3. The genetic algorithm used involved a look-up table genome, indicating which action to perform for each of the 192 possible sensory states. Mutations consisted of randomly changing exactly one item in the genome, and crossover was uniform. Extrema Selection (Stewart, 2001) with a threshold of 90% was used to increase evolution speed along neutral networks. Population size was 50 and the system ran for 300 generations.
4. Q-Learning (for both the external and internal cases) was performed using a standard look-up table memory. The exploration rate (ε: the chance of performing an action at random instead of the system's best guess) was 0.1, the learning rate (α: the amount by which to change the internal look-up table values) was 0.2, and the discounting rate (λ: the geometric reduction of importance of future rewards) was 0.95. Q-Learning maintains a table of estimated reward values for taking each possible action in each possible sensory state, and is represented by $Q_{S,A}$. These

values are all initially zero. As experience in the world occurs, and rewards/punishments are received, the values are changed. To update, we first calculate $r+\lambda \bullet maxQ_{S2}$ (where r is the received reward, λ is the fixed discounting rate, and $maxQ_{S2}$ is the largest value founding looking at all the actions that could be taken from the sensory state the agent finds itself in after performing the action. This value is then combined with the old prediction ($Q_{S1,A1}$) using the learning rate (α), resulting in the following formula:

$$Q_{S1,A1} \leftarrow (1-\alpha)Q_{S1,A1}+\alpha(r+\lambda \bullet maxQ_{S2})$$

To choose an action to perform, the system simply takes the current state and looks at each possible action that could be performed. The action with the highest Q value is chosen $1-\varepsilon$ of the time. In the remaining ε times, a random action is chosen.

5. The neural network was a feed-forward multi-layer perceptron, trained using back-propagation of error (Rumelhart, Hinton, and Williams, 1986). It had three input nodes (for the three values of the sensory state, scaled to be between -1 and 1), one output node, and three hidden nodes. The activation function for all nodes was the hyperbolic tangent, and the learning rate (α) was 0.2. To handle the feedback between the output value and the input state, the network was run 100 times.
6. Interestingly, such tests are known to exist in the animal world. Curio (1976) reports that most animals that predate on herds make a “test attack” to identify

animals whose ability to run away is insufficient to protect them. In such cases, the actions in the world are not ‘real’, but ‘tests’, or ‘simulated’ actions. And the organism uses itself and the environment as a ‘test-bed’ or ‘Simulation environment’ to judge the quality of its own actions.

7. The source code for both the simulations (written in Python) can be downloaded from this website: *http://www.carleton.ca/ics/ccmlab/epistemic.html*

Acknowledgements

We thank Dr. Andrew Brook for clarifying and sharpening the philosophical ideas presented here. Thanks also to Dr. Narayanan Srinivasan and Jennifer Schellinck for critical review and feedback. Any mistakes are of course our own.

Figure Captions

Figure 1: The architecture of the Q-learning model for external structures.

Figure 2: The computer model at 10, 100, and 300 time steps. Black dots are the agents. The shading is darker the more 'home-like' or 'target-like' a particular square is.

Figure 3: The effect of ES generation. The above figure is an average over 1000 runs of the simulation.

Figure 4: The architecture of the learning system for internal structures. Note that the only difference between this and the previous learning system is the neural network memory in the agent (upper segment).

Figure 5: The foraging performance of the agent, with and without internal structures. An average over 1000 runs of the simulation.

Figure 6: In the Simulation mechanism (left), the central executive is considered to pass processing of cognitive tasks onto the different component neural units, including the motor one, resulting in a process that is almost equivalent to the embodied agent acting in the world. In non-Simulation processing, the central executive is considered to process stored representations of the world by itself, with minimal or no input from the component neural units. This results in a disembodied process that is detached from the world. These two ways of processing (modal and amodal) need not be mutually exclusive and could be considered two ends of a continuum.

Table 1: Time spent performing various actions (ES generation).

Table 2: Time spent performing various actions (Internal Trace generation).

References

1. Bacharach, M. and Gambetta, D. (2003). Trust in Signs. In Cook, S.K., (Ed.) *Trust in Society*, Russell Sage Foundation, New York, NY.
2. Barsalou, L.W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences*, 22, 577–6604
3. Barsalou, L.W. (2003) Situated Simulation in the human conceptual system. *Language and Cognitive Processes*, 18, 513–562
4. Barsalou, L.W., Simmons, W.K., Barbey, A.K., & Wilson, C.D. (2003). Grounding conceptual knowledge in modality-specific systems. *Trends in Cognitive Sciences*, 7, 84-91.
5. Beer, R.D. (2003). The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, 11(4):209-243.
6. Bergen, B., Chang, N., Narayan, S. (2004). *Simulated Action in an Embodied Construction Grammar*. In K. D. Forbus, D. Gentner & T. Regier (Eds.), Proceedings of the 26th Annual Meeting of the Cognitive Science Society, Chicago. Hillsdale, NJ: Lawrence Erlbaum.
7. Bonabeau E., Dorigo M. and Theraulaz G. (1999). *Swarm intelligence: From natural to artificial systems*. Santa Fe Institute studies in the sciences of complexity. New York: Oxford University Press.
8. Bradbury, J.W. & Vehrencamp, S.L. (1998). *Principles of Animal Communication.* Sunderland, Mass: Sinauer Associates.
9. Brass, M. & Heyes, C. (2005). Imitation: is cognitive neuroscience solving the correspondence problem?, *Trends in Cognitive Sciences*, 9 (10)

10. Braver, T. S., & Cohen, J. D. (2000). On the control of control: The role of dopamine in regulating prefrontal function and working memory. In S. Monsell & J. Driver (Eds.), *Control of cognitive processes: Attention and Performance XVIII*. Cambridge, MA: MIT Press.
11. Brooks, R. (1997). Intelligence without Representation. In *Mind Design II: Philosophy, Psychology, Artificial Intelligence*, Haugeland, J. (Ed.), MIT Press, Cambridge, Mass.
12. Camazine, S. (1991) Self-organizing pattern formation on the combs of honey bee colonies. *Behavioral Ecology & Sociobiology*, 28: 61-76
13. Chandrasekharan, S. & Stewart, T. (2004) *Reactive agents learn to add epistemic structures to the world*. In Forbus, K.D., Gentner D., & Regier, T. (Eds.), Proceedings of the 26th Annual Meeting of the Cognitive Science Society, Chicago. Hillsdale, NJ: Lawrence Erlbaum.
14. Chandrasekharan, S. (2005). *Epistemic Structure: An Inquiry into How Agents Change the World for Cognitive Congeniality*. Ph.D. Thesis, Carleton University, Ottawa, Canada. Available as a Carleton University Cognitive Science Technical Report at: *http://www.carleton.ca/iis/TechReports/files/2005-02.pdf*
15. Clark, A., Toribio, J. (1994). Doing without Representing? *Synthese*, 101, 401-431.
16. Clark, A. (1997). *Being There: putting brain, body, and world together again*, Cambridge, Mass., MIT Press.

17. Clark, A. (1997). The Presence of a Symbol. In *Mind Design II: Philosophy, Psychology, Artificial Intelligence*, Haugeland, J. (Ed.), MIT Press, Cambridge, Mass.

18. Clark, A. and Chalmers, D. (1998). The Extended Mind, *Analysis,* 58: 1: p.7-19

19. Curio, E. 1976. *The ethology of predation*. Springer Verlag, New York.

20. Dennett, D. and Haugeland, J. (1987). Intentionality. In R. L. Gregory, (Ed.), *The Oxford Companion to the Mind*, Oxford: Oxford University Press.

21. Fodor, J. & Pylyshyn, Z. (1997). Connectionism and Cognitive Architecture: A critical Analysis. In *Mind Design II: Philosophy, Psychology, Artificial Intelligence*, Haugeland, J. (Ed.), MIT Press, Cambridge, MA.

22. Galantucci, B. (2005). An experimental study of the emergence of human communication systems. *Cognitive Science*, (29) 737-767.

23. Gallese, V., Ferrari P.F., Kohler E., Fogassi L. (2002). The eyes, the hand and the mind: behavioral and neurophysiological aspects of social cognition. In: *The Cognitive Animal*. Bekoff, M., Allen, C., Burghardt, M. (Eds.), MIT Press, pp. 451-462.

24. Gould, J.L. (1990) Honey bee cognition. *Cognition*, 37, 83-103.

25. Guilford, T., Roberts, S. & Biro, D. (2004) Positional entropy during pigeon homing II: navigational interpretation of Bayesian latent state models. *Journal of Theoretical Biology*, 227 (1), 25-38.

26. Henry, J.D. (1977). The use of urine marking in the scavenging behaviour of the red fox (Vulpes vulpes). *Behaviour*, 62:82-105.

27. Heyes, C., Bird, G., Johnson, H., Haggard, P. (2005). Experience modulates automatic imitation. *Cognitive Brain Research*, 22 (2005) 233-240

28. Hutchins E. (1995a). *Cognition in the Wild*. MIT Press, Cambridge, Mass.

29. Hutchins, E. (1995b). How a cockpit remembers its speeds. *Cognitive Science*, 19, 265-288.

30. Kahneman, D., & Tversky, A. (1982). The Simulation heuristic. In D. Kahneman, P. Slovic, & A. Tversky, (Eds.), *Judgment under uncertainty: Heuristics and biases* (pp. 201-208). New York: Cambridge University Press.

31. Kirby, S. (2002) Natural Language from Artificial Life. *Artificial Life*, 8(2):185--215.

32. Kirsh, D., & Maglio, P. (1994). On distinguishing epistemic from pragmatic action. *Cognitive Science*, 18, 513-549.

33. Kirsh, D. (1995). The Intelligent Use of Space. *Artificial Intelligence*. 73: 31-68

34. Kirsh, D. (1996). Adapting the environment instead of oneself. *Adaptive Behavior*, Vol 4, No. 3/4, 415-452.

35. Mandik, P. and Clark, A. (2002). Selective Representing and World Making. *Minds and Machines*, 12: 383-395.

36. Metzinger, T & Gallese, V. (2003) The emergence of a shared action ontology: Building blocks for a theory. *Consciousness and Cognition*, 12, 549 - 571. Special Issue on Grounding the Self in Action.

37. Miller, M., & R. Wehner (1988). Path integration in desert ants, Cataglyphis fortis. *Proceedings of the National Academy of Sciences* USA 85: 5287-5290.

38. Millikan, R. G. (1994). Biosemantics. In Stich, S. and Warfield, T. A. (Eds.), *Mental representation, a reader*. Oxford: Blackwell.

39. Montague, P.R., Dayan, P. & Sejnowski, T.K. (1996). A framework for mesencephalic dopamine systems based on predictive Hebbian learning. *Journal of Neuroscience*, 16, 1936-1947.

40. Nakamura, M., & Kurumatani, K. (1996). *Formation mechanism of pheromone pattern and control of foraging behavior in an ant colony model*. Proceedings of the Fifth International Conference on Artificial Life, 67-74.

41. Nichols, S., Stich, S., Leslie, A., and Klein, D. (1996) Varieties of Off-Line Simulation. In *Theories of Theories of Mind*, eds. P. Carruthers and P. Smith. Cambridge: Cambridge University Press, 39-74.

42. Rheinberger, H.-J. (1997). *Toward a History of Epistemic Things: Synthesizing Proteins in the Test Tube*, Stanford University Press.

43. Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986) Learning representations by back-propagating errors. *Nature*, 323, 533--536.

44. Schultz, W. (1992). Activity of dopamine neurons in the behaving primate. *Seminars in Neurosciences*, 4, 129-138.

45. Silberman, S. (2003), The Bacteria Whisperer. *Wired*, Issue 11.04, April 2003.

46. Smolensky, P (1997). Connectionist Modeling: Neural Computation/ Mental Connections. In *Mind Design II: Philosophy, Psychology, Artificial Intelligence*, Haugeland, J. (Ed.), MIT Press, Cambridge, MA.

47. Steels, L. and Belpaeme, T. (2005). Coordinating Perceptually Grounded Categories Through Language: A Case Study For Colour. *Behavioral and Brain Sciences*, 28, pp. 469-529.

48. Stewart, T.C. (2001) *Extrema Selection: Accelerated Evolution on Neutral Networks*. Proceedings of the IEEE Congress on Evolutionary Computation.

49. Stewart, T. & Chandrasekharan, S. (2005). *Two Cognitive Descriptions of Q-Learning*. Carleton University Cognitive Science Technical Report. Available at: *http://www.carleton.ca/iis/TechReports/files/2005-03.pdf*

50. Stopka, P. & Macdonald, D. W. (2003) Way-marking behavior: an aid to spatial navigation in the wood mouse (Apodemus sylvaticus). *BMC Ecology*, published online, *http://www.biomedcentral.com/1472-6785/3/3*

51. Susi, T. & Ziemke, T (2001). Social Cognition, Artifacts, and Stigmergy: A Comparative Analysis of Theoretical Frameworks for the Understanding of Artifact-mediated Collaborative Activity. *Cognitive Systems Research*, 2(4), 273-290.

52. Svenson, H. and Ziemke, T. (2004). *Making Sense of Embodiment: Simulation Theories and the Sharing of Neural Circuitry Between Sensorimotor and Cognitive Processes*. In K. D. Forbus, D. Gentner & T. Regier (Eds.), Proceedings of the 26th Annual Meeting of the Cognitive Science Society, Chicago. Hillsdale, NJ: Lawrence Erlbaum.

53. Tani J., Nolfi S. (1997). *Self-organization of modules and their hierarchy in robot learning problems: A dynamical ststems approach*. Technical report, SCSL-RL-97-008, Sony CSL, Tokyo, Japan.

54. Tesfatsion, L. (2002). Agent-Based Computational Economics: Growing Economies from the Bottom Up, *Artificial Life*, Vol. 8 (1), 2002, pp. 55-82

55. Todd, P.M., and Miller, G.F. (1999). From pride and prejudice to persuasion: Realistic heuristics for mate search. In Gigerenzer, G., Todd, P.M., and the ABC Research Group, *Simple heuristics that make us smart*. New York: Oxford University Press.

56. Watkins, C. (1989). *Learning From Delayed Rewards*, Doctoral dissertation, Department of Psychology, University of Cambridge, Cambridge, UK.

57. Zahavi, A., & Zahavi, A. (1997). *The Handicap Principle: A missing piece of Darwin's puzzle*. Oxford: Oxford University Press.

Figures and Tables

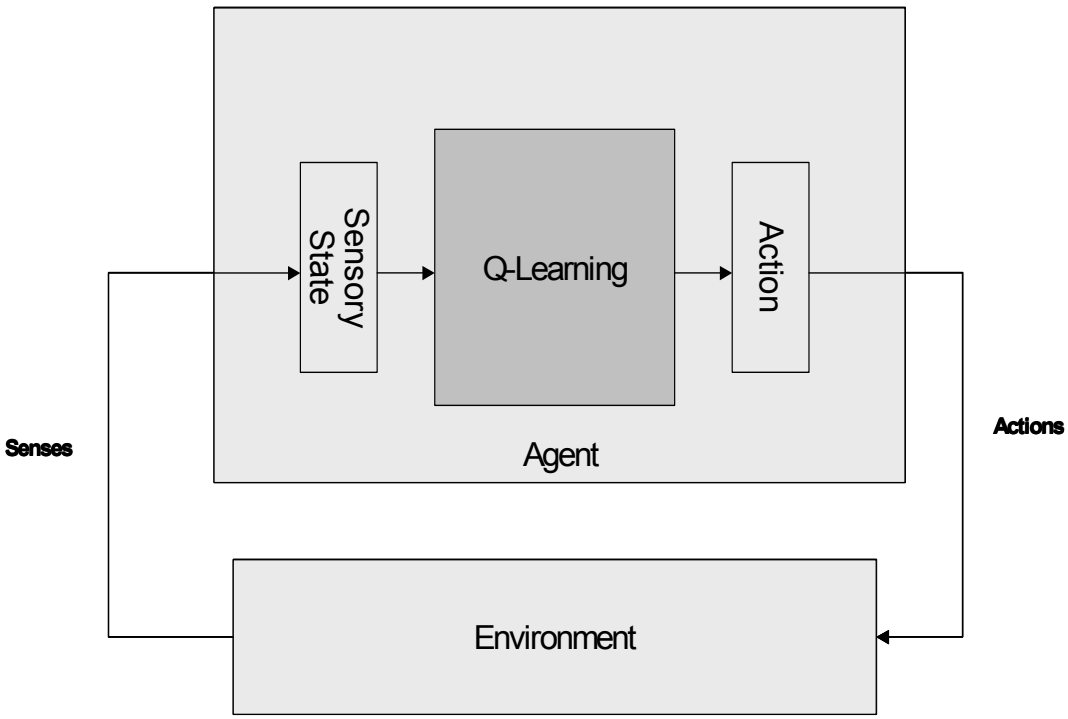

Figure 1: The architecture of the Q-learning model for external structures.

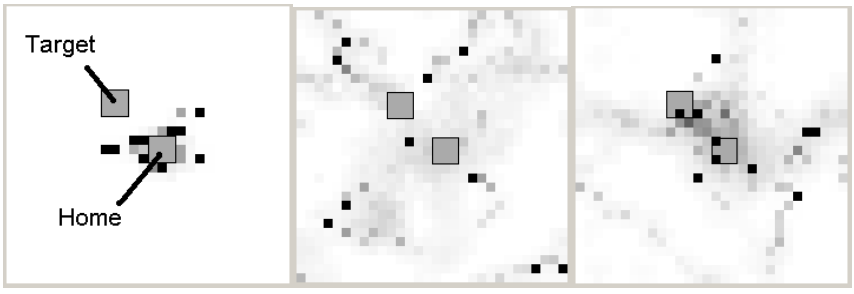

Figure 2: The computer model at 10, 100, and 300 time steps. Black dots are the agents. The shading is darker the more 'home-like' or 'target-like' a particular square is.

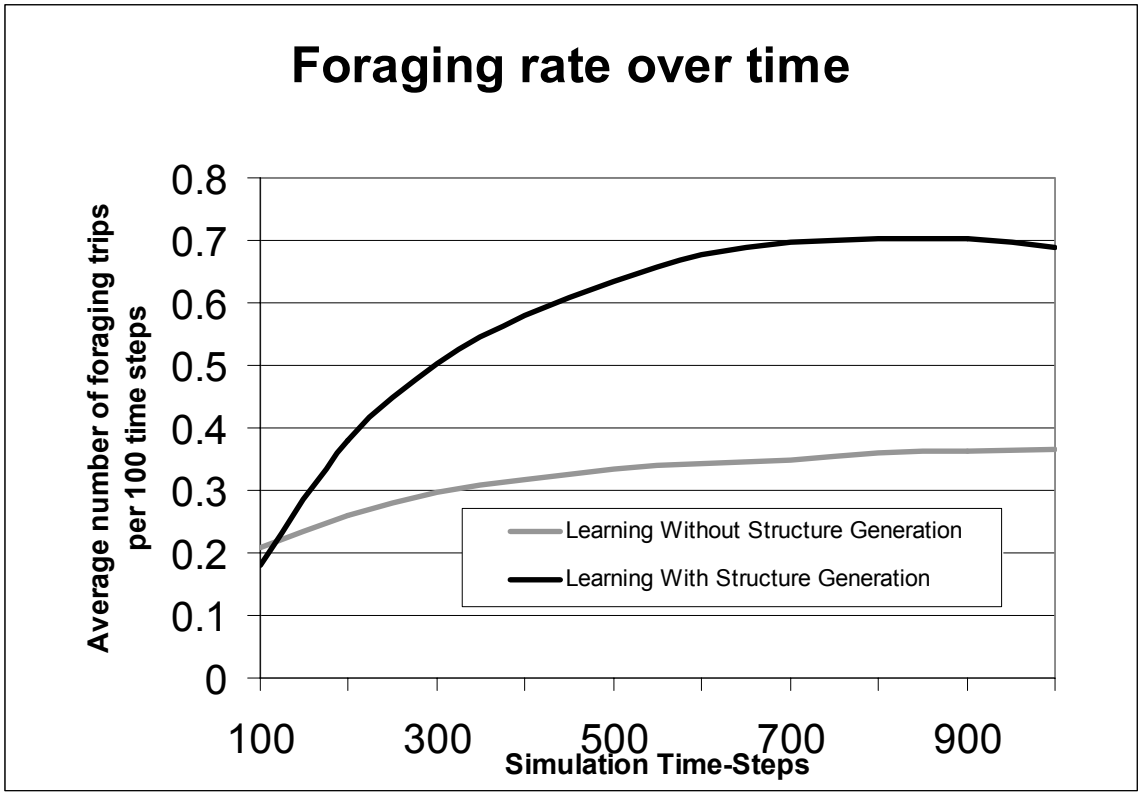

Figure 3: The effect of ES generation. The above figure is an average over 1000 runs of the simulation.

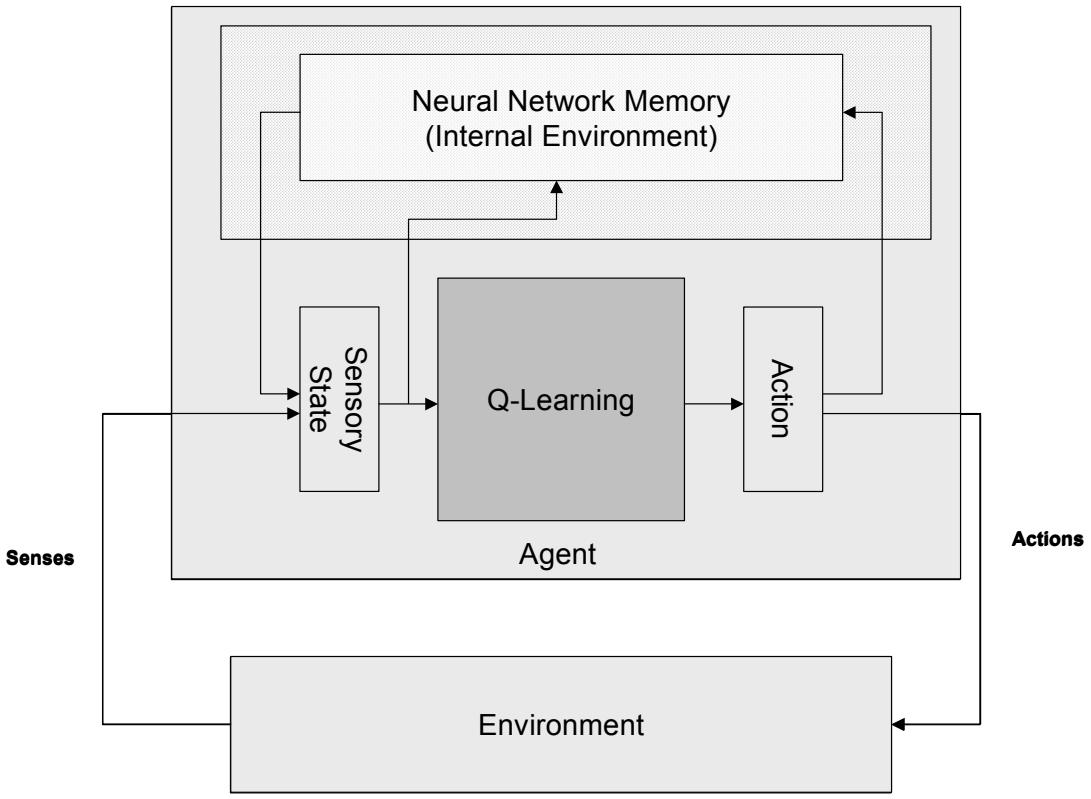

Figure 4: The architecture of the learning system for internal structures. Note that the only difference between this and the previous learning system is the neural network memory in the agent (upper segment).

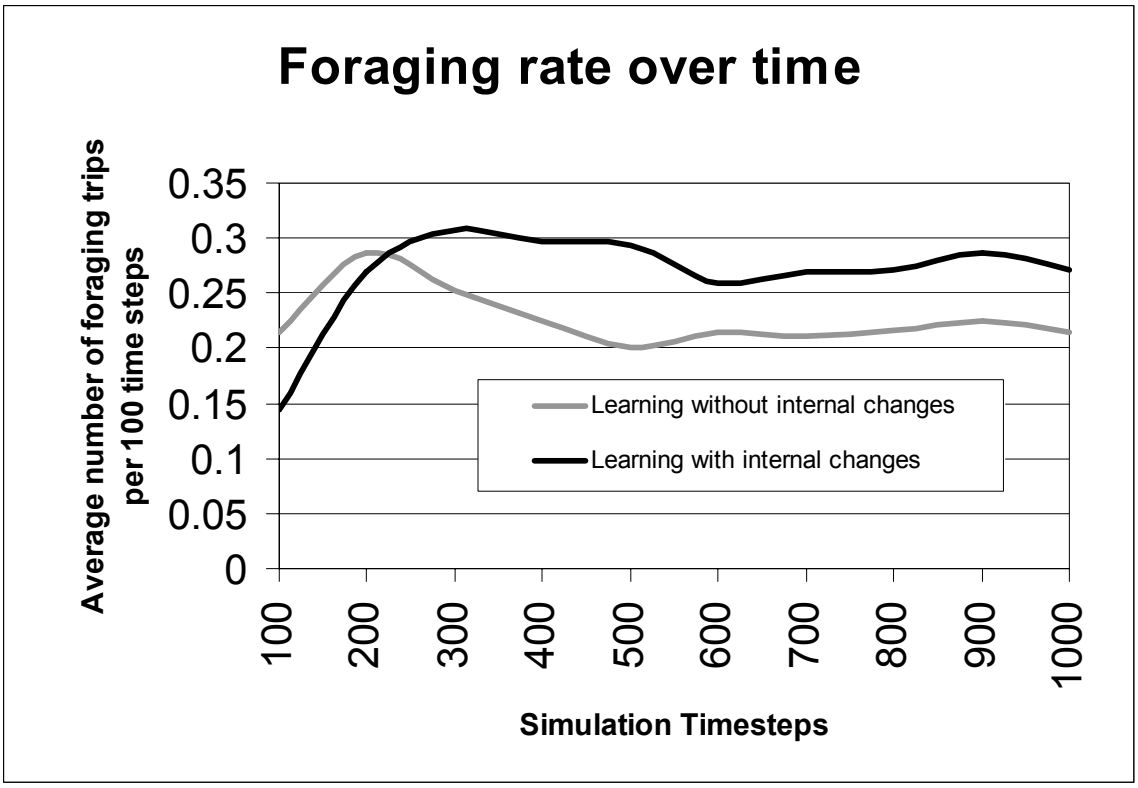

Figure 5: The foraging performance of the agent, with and without internal structures. An average over 1000 runs of the simulation.

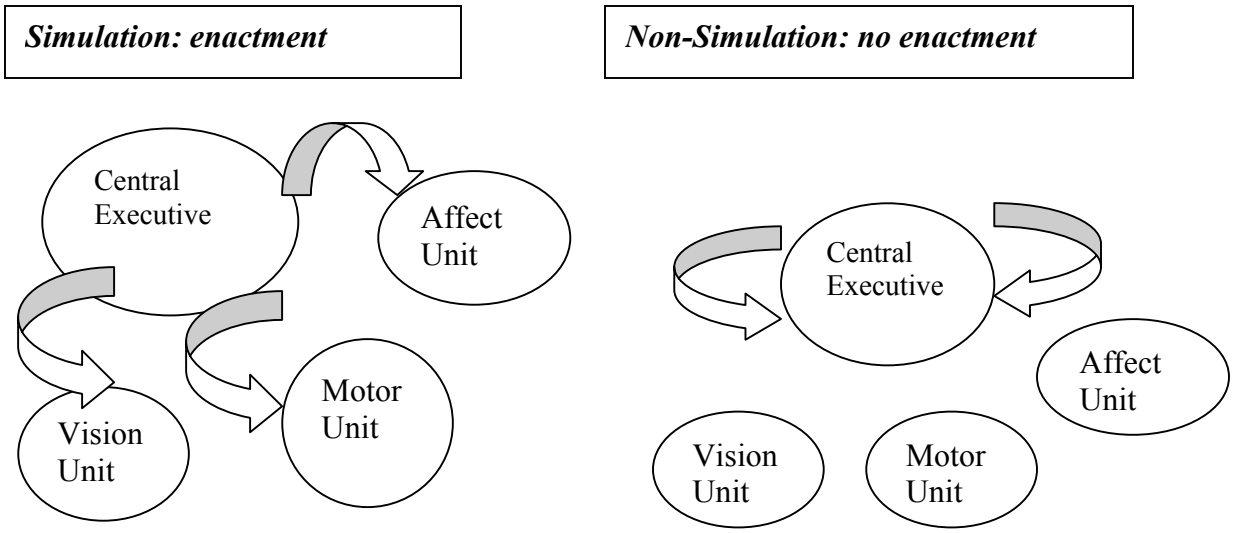

Figure 6: In the Simulation mechanism (left), the central executive is considered to pass processing of cognitive tasks onto the different component neural units, including the motor one, resulting in a process that is almost equivalent to the embodied agent acting in the world. In non-Simulation processing, the central executive is considered to process stored representations of the world by itself, with minimal or no input from the component neural units. This results in a disembodied process that is detached from the world. These two ways of processing (modal and amodal) need not be mutually exclusive and could be considered two ends of a continuum.

Table 1. Time spent performing various actions (ES generation).

Action	With Structure Generation	Without Structure Generation
Move randomly	10%	32%
Toward 'home-like'	19%	36%
Toward 'target-like'	13%	32%
Make 'home-like'	**35%**	
Make 'target-like'	**23%**	

Table 2. Time spent performing various actions (Internal Trace generation).

Action	With Trace Generation	Without Trace Generation
Move randomly	12.8%	30.4%
Go to home	37.2%	38.8%
Go to Target	27.5%	30.8%
Remember 1	**11.0%**	
Remember 0	**11.5%**	