

# How Do Place and Objects Combine? “What-Where” Memory for Human-like Agents

(an extended version of a paper presented on Intelligent Virtual Agents '09)

Cyril Brom<sup>1</sup>, Tomáš Korenko<sup>1</sup>, Jiří Lukavský<sup>2</sup>

<sup>1</sup>Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic

<sup>2</sup>Institute of Psychology, Academy of Sciences, Prague, Czech Republic

**Abstract.** Believable spatial behaviour is important for intelligent virtual agents acting in human-like environments, such as buildings or cities. Existing models of spatial cognition and memory for these agents are predominantly aimed at issues of navigation and learning of topology of the environment. The issue of representing information about possible objects' locations in a familiar environment, information that can evolve over long periods, has not been sufficiently studied. Here, we present a novel representation for “what-where” information: memory for locations of objects. On a simplified model of a virtual character living in a virtual house, we investigate how this representation is formed and how it evolves based on how objects are moved within the environment. The behaviour of the model is also compared with behaviour of real humans conducting an analogical task. The results allow us to conclude that this model can be incorporated to other memory models for virtual agents.

## 1 Introduction

Humans act in space. They furnish the space they live in with objects. Be it a van Gogh's painting or a pen, it pays to remember where ones' belongings are located. This makes life much simpler. Intelligent virtual agents (IVAs) usually act in space as well. This demands them to have similar “what-where” information as humans do. In many present-day applications, IVAs read this information directly from the world map, which corresponds to complete environmental knowledge. While this approach may suffice for static words or dynamic but fully observable words, it results in unbelievable behaviour in dynamic words that are not fully observable. The latter kind of environments is increasingly employed by today applications. For instance, think of non-player characters (NPC) from a role-playing game (RPG), or virtual/robotic companions required to orient themselves in humans houses. Here, a better approach is needed.

Why not to memorise all objects that an agent encountered? Assume we have a large, partially observable environment with objects that are passive but whose locations can be changed by external forces beyond the agent's capabilities. For instance, a pen can be moved by a fellow agent. In this situation, we can expect multiple memory records of a position of each object based on the history of the

object’s moves. Where is the pen: at the working table or next to the TV? A simple list of memory records can not answer this.

How can we improve the performance? Consider objects that humans use. Humans have some organisation of the placement of their belongings; things are not placed randomly within their surroundings, instead, they are clustered purposefully according to their needs and cultural norms. Some objects appear regularly at some places (newspapers inside a mailbox). Other objects are almost never being relocated (a van Gogh’s painting). Yet others are being relocated so often that it is not practical to remember their exact position. Consequently, when a human searches for an object, often, a sort of stimulus-response mechanism is employed. For a different object, several places are inspected in a specific order; sometimes, the whole house is scrutinised but starting at a specific place.

This brings us to the notion of *searching rules*, which are basically a sequence of places that should be inspected when searching for an object of a particular kind. Importantly, we mean by place any logically coherent space abstraction. These abstractions can differ in size and can be hierarchically nested; e.g. a bedside table, a place between this table and the bed, a corner of the living room, a living room, a house, etc. This corresponds to the way humans are supposed to cluster space [e.g. 8]. A “what-where” memory for IVAs acting in large, partially observable environments with movable objects should have the ability to develop searching rules.

In this paper, we present a model with this ability. To its advantage, it is quite simple. The model has been integrated with our agent with general memory capabilities [3] and investigated in scenarios mimicking a situation of a person moving into a new house. Specifically, we focused on the questions of how the initial representation is formed, how (and whether) searching rules emerge, and how the model relearn, measuring effectiveness of the model’ behaviour. We also investigated behaviour of the model with different parameters, aiming to find a point of “optimality.” Additionally, we conducted a simple experiment with human subjects that mimicked the agent’s task using a simple Flash application, and evaluated the model qualitatively against the acquired human data.

The results allow us to conclude that the searching rules emerge easily and quickly and that they are qualitatively similar to searching rules developed by humans in their version of the task. The model also relearns well. Additionally, humans’ data helped us to isolate one feature of human behaviour that cannot be explained by the notion of searching rules straightforwardly, but it can be added onto a top of the model easily. In overall, our opinion is that the model is now ready to use in real-world applications requiring plausible “what-where” memory.

The structure of the paper is as follows. Sec. 2 reviews related work. Sec. 3 details the memory model. Sec. 4 details the experiments. Sec. 5 discusses the results.

## 2 Related work

From the psychological perspective, the ability to locate objects is a faculty of spatial cognition, which is tightly connected to spatial memory. Spatial memory is conceived as a set of multiple interconnected systems rather than a monolithic block [4].

Recently, several general memory frameworks for IVAs were presented [3, 9, 10, 16, 18]; however, the degree to which they address any issue of spatial cognition is minimal. Fortunately, several works directly focussing on some aspects of spatial cognition for IVAs have emerged to increase believability of their spatial behaviour, including mapping, localisation, and navigation. Noser et al. developed IVAs learning topological structure of the environment and navigating using this structure [15]. In a psychologically more plausible manner, Thomas and Donikian [23] addressed similar issue. A mechanism for anticipating position of an object that can move itself, e.g. a sheep, was presented by Isla and Blumberg [11]. Unfortunately, neither of these works addressed sufficiently the issue of “what-where” memory for passive, but movable objects. Unlike these models, the mechanism presented in [20] directly aimed at representing both topological as well as “what-where” information and this information could gradually deteriorated when not refreshed. However, neither this model was designed to cope with objects that can be moved several times, lacking the ability to develop searching rules.

The field of gaming AI predominantly addresses believable and efficient path-finding and automatic construction of space representation [e.g. 7, 21]. To our knowledge, the issue of “what-where” memory is not addressed. Finally, we are not aware of any work either from computational psychology or robotics that could suit our purpose directly. Robotics tend to focus on the issues of localisation and terrain mapping [e.g. 13], which are “low-level” from the perspective of IVAs, pointing to the significant difference between application domains. Even designers of robots with episodic memory systems [e.g. 6] do not seem to consider the “what-where” issue as a crucial one for their discipline. Psychological experiments investigating spatial abilities of humans are of considerable interest to the field of IVAs [see 4, 22 for reviews of some], but computational psychology tend to produce special-purpose models replicating data gained during laboratory tasks [e.g. 1]. It is hard to imagine a meaningful application in which an IVA could be engaged in such a task. Additionally, regarding “what-where” information, psychology tends to investigate what would be called in our context short-term representations of positions of static objects and their mental rotations, e.g. for the purposes of elucidating the allocentric—egocentric tension [reviewed in 4].

### 3 Model

The purpose of this section is to introduce our model, the “what-where” memory, starting in an informal tone and proceeding to the model’s formalisation. The integration of the model with the rest of an IVA’s “mind” will be described in Sec. 4.

The model is a simple associative network. It is composed of two kinds of nodes: *objects* and *places* (Fig. 1). Place nodes represent places with different levels of complexity and they are hierarchically nested. Object nodes have weighted links with place nodes; these stand for “what-where” information: a possible occurrence of a particular object at a particular place. Now, if an object is found by the agent, or comes to the agent’s attention, the links to *all* the locations where it has been found are strengthened (e.g. the links from the glasses to a) the bedside table, b) to the bedroom, and c) to the whole house – see Fig. 1).

How could searching rules possibly emerge from this network? Note several things. (1) Links from an object node to nodes representing places at a similar level of complexity approximates the probability distribution of finding the object at given places. (2) Links to nodes representing more abstract places are strengthened more often than links to nodes of concrete places (“glasses are always in the house, but only sometimes at the bedside table”). Now, if queried for an object position, we can find the appropriate object node and past locations of the object’s occurrence via the object—place links. Assuming that the pattern of the object’s movement will be same in the future as it was in the past, we can arrange the place nodes in order of the strengths of the links leading to them (Point (1)). However, because of Point (2), before we do this, we need to scale the links’ strengths by an inverse function of the complexity of places; otherwise, the abstract places will be always first in the list.

The fundamental assumption is that with an appropriate scaling function, the result will be a list with balanced ordering of concrete and abstract places where the object can be looked for, a searching rule. Concrete places should be first on the list provided that there are only a few concrete places where the object can be found. Otherwise, an abstract place should be first or very close to the beginning of the list. Particular places can be searched directly (e.g. the bedside table) while abstract places should be inspected (e.g. scrutinise the kitchen). The fact that this mechanism really produces believable searching rules is demonstrated in the next section.

An important question is how to deal with distances. The present version of the model ignores distances, conceiving the searching rules as verbal answers on the question: “Where do you think is an object X?” Our searching rules are *disembodied*. Our opinion is that in a middle-sized environment, e.g. in the ground-floor house employed by our experiments, we can ignore distances letting the agent search for at most probable places, but distances become important in larger environments, such as in multi-floor buildings or cities. How to take distances into account? One can basically take disembodied rules as inputs for an engine solving the *travelling salesman problem* (TSP) penalising the improbable places. Ideally, the penalisation should reflect how people solve the same problem. Surprisingly enough, it seems the TSP with uncertainties has not been investigated in psychology until recently [24], and this work is not conclusive from the standpoint of IVAs. There are only many works revealing heuristics human use to tackle the classical TSP [24]. Thus, we assign the question of plausible penalisation of distances as future work.

**Definition.** Formally, the network is a triple  $\langle P, O, E \rangle$ .

$P$  is the set of all *place nodes*, each of which is a quintuple  $\langle p, up, down, level, size \rangle$ , where  $p$  is the node,  $up$  its super-location,  $down$  its sub-locations,  $size$  is a number of its sub-locations, and  $level$  is the level of abstraction. Abstractions are numbered from the bottom:  $level$  for a specific place is 1, and then the levels are enumerated by one towards the root of the hierarchy.

$O$  is the set of all *object nodes*, i.e. the object records.

$E$  is a set of weighted *edges*, each of which is  $\langle x, found, missed \rangle$ , where  $x \in P \times O$  is the edge, and  $found \in \mathbf{N}$  is the number of times the object was found (and taken) or seen at the particular place, and  $missed \in \mathbf{N}$  is the number of times the object was being searched for at the place but not found there.

In the present version of the model, we assume that  $P$  is specified in advance by a designer and fixed during the simulation (but see also Sec. 5).

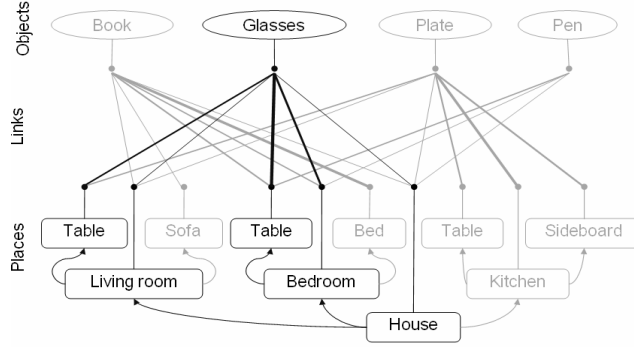


Fig. 1. Spatial memory. Some nodes from the experimental scenario are schematically depicted; the glasses’ links are highlighted. The width of the links denotes the size of the *found* variable.

**Learning mechanism.** When the model is requested to store positional information about an object, it first checks whether the corresponding object node exists and creates a new one if needed. Then, the *found* variable of links between the object node and place nodes of *all* the places where the object was found are increased by 1 (e.g. the house, the bedroom, the bedside table). When the agent is looking for an object at a specific place and this object is not there, the *missed* variable is increased by 1. If the agent is searching in a location and the object is not in any of its sublocations, *missed* variables for this location and all the sublocations are increased by 1.

**Rules formation.** This is perhaps the trickiest part. When the model is queried for an object position, so-called *size-normalised trustfulness* (SNT) is computed for every place the object has an edge with as:

$$SNT = f(found, missed) / complexity(level, size) \quad (1)$$

The function  $f$  determines influence of the *found* and *missed* variables on the estimation of likelihood of finding the object at the given place and *complexity* is the scaling function. There are more options how to choose  $f$  and *complexity*. In our experiments (Sec. 4), we will investigate how the model behaves with the following functions, where  $a$ ,  $b$ , and  $c$  are parameters to be found:

$$f = b \cdot found - missed \quad (2)$$

$$complexity = level^a \quad (3)$$

$$complexity = size^c \quad (4)$$

The places are arranged in order of SNT. More specific places after their superlocation are dropped from the rule. The sorted list presents a searching rule.

Notice that (3) completely disregards sizes of places, leading to assigning the same SNT to two rooms: one with six containers and another with only one container. Clearly, a human would prefer to inspect the latter room first. This motivates Eq. (4). Further discussion on how  $f$  can look like follows in Sec. 5.

## 4 Implementation and Experiments

**Motivation and Synopsis.** The most important questions relate to whether searching rules emerge and how quickly. In particular, we aim to investigate: A. Do searching rules emerge? B. Does the time to emergence depend on the frequency of usage of an object? C. Is searching effective? To this end, we designed the first set of experiments investigating the model with the SNT defined by Equations (1) – (3) and with fixed parameters ( $a = 2$ ,  $b = 1.3$ ).

Assuming positive answers, we ask further: with which parameter settings does the model behave optimally? The “optimality” has two components. On the one hand, we can ask: D. How quickly the rules emerge with a particular setting? E. How quickly the network relearns with a particular setting? F. Is there one setting optimal for all situations? To this end, we designed the second set of experiments in which we investigated how the behaviour of the model changes based on varying parameters  $a$ , and  $b$ , and also varying  $c$  after replacing Eq. (3) by (4). On the other hand, we should ask: G. Are the rules similar to those used by humans; are they believable? As discussed above, we are not aware of any psychological experiment that has produced data against which we can validate our model. Therefore, we designed a simple Flash-based application in which we investigated behaviour of human subjects in a task similar to the task used in experiments with the model.

**Implementation.** The “what-where” model has been integrated within our generic agent with episodic memory capabilities [3] (see Fig. S1 – note that all figures from Supplementary materials have  $S$  prefix). From the perspective of this agent, the model presents the long-term memory for positions of objects (LTSM). The agent also possesses an *attention filter* through which only some percepts can pass, a simple *short term memory* – an intermediate stage for object records that are to be later stored in the LTSM –, and an *autobiographic memory* with forgetting. For the purposes of forgetting, the agent features a simple valence-based *emotion model*. A simple *linguistic module* allows the agent to tell short stories about its “life.” The *action selection* mechanism is a derivation of the BDI [2]. Presently, we have five independent implementations, four of them employing a 2D grid world, the last one using a 3D world of the action game Unreal Tournament 2004 [12]. Here, we use a 2D world for simplicity. Given the abstract nature of the “what-where” map, we find a 2D world appropriate for our purpose and do not see the scaling to the 3D world as a problematic issue.

To investigate the experimental questions, we need an “ecologically” plausible scenario, which would be, on one hand, sufficiently complex to model the desired phenomena, and on the other hand, simple enough to allow for interpretation of the results. We adopt a “new house” scenario, where we simulate the agent living for several weeks in a house to which it moves at the beginning of the scenario. The house comprises 6 rooms, each having 4 containers, i.e. “atomic” places (24 in sum; Fig. S2). In a container, an unlimited number of objects can be located. The objects are state-less (for the sake of trialling). The environment is abstract; the containers are named (e.g. “a bedside table”), but these are merely labels. Movement of the agent: the agent “jumps” from a room to a room – it is not walking “continuously” (again, for the sake of simplifying the experiments). Once in a room, it can inspect each container instantly, without any effort, which makes the exact position of a container

in the room unimportant. The agent never leaves the house. Movement of the objects: First, the positions of objects can be changed by the agent. Second, they can be changed beyond its capabilities and unbeknown to it, a mechanism for imitating presence of another agent moving the objects.

The LTSM mirrors the topology of the environment containing 31 place nodes; one for the house, 6 for the rooms, 24 for the containers. There can be up to 24 different objects in the world. The time flow differs in each of the experiment – see below.

**Object classes.** The model will behave differently for objects with different patterns of movement (compare glasses with a PC). Hence, we defined five classes modelling prototypical behaviour of five distinct kinds of objects:

- “90%”: There is 90% chance that the object is located in a particular container, and 10% chance that it is located randomly (uniform distribution) in another container (e.g. a can opener).
- “3x30%”: There are 3 containers in a same room, in each of them the object can be located (3 x 30%). 10% that it is located randomly (uniform distrib.) in another container (e.g. glasses).
- “1/2x30%”: The same as 2, but two containers are in the same room and the last one elsewhere.
- “1/1/1x30%”: The same as 2, all containers in distinct rooms.
- “uniform”: The object can be located in any container (randomly, uniform distribution) (this is a “control” case, objects do not behave like this, though a pen approaches this pattern).

**Experiment 1.** This experiment was designed to investigate Questions A – C.

*Method.* How to investigate emergence of searching rules? Basically, we can ask the agent “where an object is” and observe how the agent’s suggestions evolve in time. Assume we ask the agent repeatedly about a particular object. If the agent lives in the house shortly, two consecutive answers will likely differ, specifically for an object changing its position often, e.g. glasses, because the memory does not hold enough records about possible positions of this object: the memory does not reflect the statistical nature of the pattern of movement of the object yet. After a time, two or more consecutive answers will likely be the same. In this situation, we can say that the rule for this particular object has stabilised. We defined operationally that a rule is stable *iff* both of these conditions hold: a) the rule was used at least 10 times, b) it was used in more than 80% of conducted searches (if there were more than 50 searches from the beginning, we looked only at the last 50 searches).

We conducted two variants of the experiment. In each variant, there were 24 objects in the environment and the agent’s task was to search for some of them, developing its LTSM. The first variant investigated Question A. We let the agent repeatedly, randomly (uniform distrib.), without any motive, to choose which object to pick up, measuring time in number of searches. In the second variant, investigating B and C, we needed to examine the frequency thus to break the uniformity of assigning objects to be searched for. We simulated casual life of the agent, defining 4 daily plans from which we chose one every day (Tab. S1). Individual tasks were represented by a BDI-like formalism. In each task, the agent used several objects. Thus the plans brought different and plausible frequencies of searches for objects

needed during each day. One time-step lasted 3 seconds, according to [19]. On average, the agent searched for an object 96 times day in the 2<sup>nd</sup> variant. For comparison, we can image that one day in the 1<sup>st</sup> variant also lasted 96 searches, i.e. 4 searches for an object a day on average. Each variant had three alternatives. The alternatives differed in classes of objects' dynamics: all objects belonged to classes "90%," "3x30%," or "uniform," respectively. The SNT was defined by Eqs. (1) – (3) with  $a = 2$ ,  $b = 1.3$ . Initial positions of objects were randomised. The results were averaged over 20 trials for each alternative.

*Quantitative results.* The 1<sup>st</sup> variant indeed showed that rules were stabilised in very short time (Fig. 2/Left), for the "90%" class just in 12 searches on average and for the other classes in 24 to 40 searches on average. Dispersion of the stabilization time, that was quite big (4 to 8 searches based on an object class), was caused by initial positions of objects. The 2<sup>nd</sup> variant demonstrated that time to stability depends both on the object's class as well as the frequency of searching (Fig. 2/Right, Fig. 3). Notice that the overall pattern of ordering of the classes ("90%" is the first, "3x30%" is the last) is still recognisable (Fig. 2/Right). In sum, the answers on Questions A and B are positive. Additionally, all searching rules for the three classes reached the optimum in less than 6 searches (Fig. 4), i.e. even sooner than the rules were stabilised, giving the positive answer on Question C.

*Qualitative results.* An important question is how do the searching rules look like? The rule developed for "90%" class always consisted of looking at the place having the 90% probability, then exploring the room with this place and then exploring the whole house. This rule typically emerged (and did not change from that time on) just in 3 searches. The emerged rule for "3x30%" class consisted of exploring the room

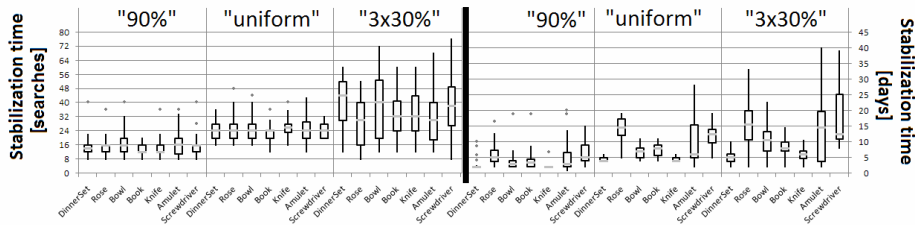


Fig. 2. Stabilization time. For brevity, only 7 objects from each class are depicted; other data are similar. Left: 1<sup>st</sup> variant. Right: 2<sup>nd</sup> variant. Note the different y-axes and their scales. On the left, the stabilisation time is given in nr. of searches, on the right, in days. On the left, 4 searches per object can be considered as one day.

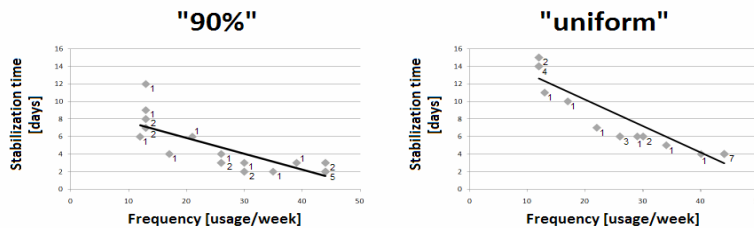




Fig. 3. Dependence of the stabilisation time on the frequency of usage; classes “90%” and “uniform.” The frequency is given as the average number of searches for a particular object during one week; the average is over all weeks from 20 trials of the 2<sup>nd</sup> variant. The numbers next to the points denote the number of objects at a same coordinate.

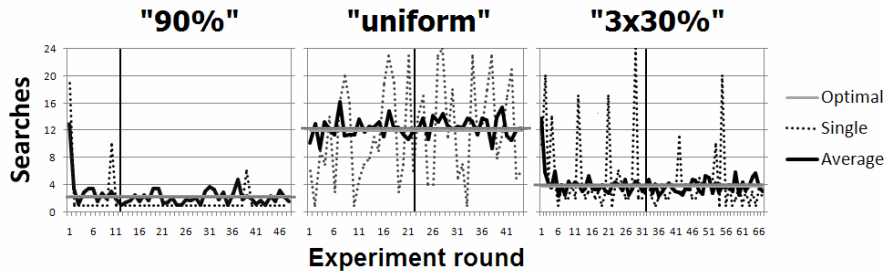


Fig. 4. The number of visited places during searching for 3 classes. One example strategy is compared with the optimal strategy and an average over all 20 trials of the respective alternatives of 2<sup>nd</sup> variant of Exp. 1. The abrupt jumps were caused when objects were not in the most probable places. The vertical lines denote stabilisation time. The optimal rules were chosen as follows: “90%”: inspect the “90% container”, then the whole house; “uniform”: inspect the house; “3x30%”: inspect the 3 most probable containers in a random order, then the house. The number of searches for the optimal rules were averaged over 10 000 simulations.

with the most probable places and then exploring the whole house. This rule emerged typically in 6 searches. Why the optimal rule “to inspect the three containers” is not preferred? Indeed, the network preferred it when  $b=3$  (instead of 1.3), which, however, slowed down the learning of “3x30%” rule three times. Apparently, we have stumbled on a trade-off which we will comment later. The searching rule for “uniform” class was: to explore the whole house.

*Summary.* Given the results, we conclude that searching rules emerge easily in the setting of a middle-sized environment, that the time to emergence of a rule depends on the frequency of objects’ usage, and that the emerged rules present near-optimal searching heuristics. What needs to be investigated now is whether the model is able to relearn after an object changes its class (e.g. “glasses should be in the case from now on”) and how does this depend on the model’s parameters, whether the model is robust with regard to different environments (e.g. “what will happen when having 6 containers instead of 4?”), and whether the rules are similar to those used by humans.

**Experiment 2.** This experiment was designed to answer Questions D – G. The main idea is to investigate how quickly the model learns and how quickly it relearns when the class suddenly changes unbeknown to the agent, varying the model’s parameters. This is actually an unnatural “guessing game.” In real world, people typically knows that the movement pattern of an object is about to change (“I will move the PC to another room”, “I will use the case for my glasses from that time on.”). However, this awareness of change can be implemented by increasing learning speed temporarily, which can only improve the performance. For the purpose of trialling, this temporal increase would make a sense only when the results without the increase are negative.

*Method.* We investigated all 25 possible variants of relearning (5 x 5: each class can be switched to each other). When switching a class to the same one, the containers and rooms with the highest probabilities were changed for the “90%” class and all “30%” classes. The switch from “uniform” to “uniform” had no effect. For each of the 25 variants, the model started with no previous information. It was requested to search for the same object (from the given class) for 20-30 times depending on the variant. About in the middle, the class was switched.

How to vary the parameters’ values? We first found iteratively a near-optimal value of  $a$  for each variant for the SNT function given by Eqs. (1), (2), (3) with  $b=2$  fixed (Exp. X1). This gave us an idea about meaningful intervals for  $a$ . Then, we iteratively found near-optimal values of both  $a$  and  $b$  for each variant (Exp. X2). Then, we investigated for each variant the case of SNT being given by Eqs. (1), (2), (4); i.e. the parameters  $b$  and  $c$  (Exp. X3). Finally, we searched for the best parameters across all the variants (Exp. X4). For each parameter setting, we run the experiment 10 times, averaging the results. For each run, the behaviour of the object was randomised (with respect to its class).

To help us to answer Question G, human subjects (students of our university,  $N=20$ , 20-23 years old, 16 males) were engaged in a simplified version of this task in a Flash application. The application depicted the same house in which the model had to search for objects (Fig. S2, Suppl. video 1). Searching was mimicked by clicking a mouse button. Initially, all rooms were semi-transparent. When a subject wanted to search in a room, he or she had to open it first by clicking at it, making it transparent. Then, the subject could inspect individual containers (again, by clicking at them). Thus, we penalised passage between rooms; it cost an extra click. Objects were invisible, but when the right container was found, the subject was noticed. Before the start, each subject was trained in how to use the application. Each subject was then instructed to find the object in “as low number of steps as possible”. Each subject was tested in all the 25 variants; the switches between classes occurred with the same timing as for the model. The order of variants was randomised across the subjects. After each variant, a subject was interviewed concerning his/her searching rule.

Two rules are considered as same *iff* equal except of permutation of places with equal probabilities. According to this criterion, we grouped human rules and picked a candidate from the largest cluster. This candidate was then compared with the model’s rule. *Effectiveness* is defined for each of the 25 variants as *searches/requests*. *Searches* is the number of inspected containers during the whole variant plus 1 for each passage from a room to another room (i.e. the number of clicks for humans). *Requests* is the number of requests for searching for an object within the variant.

*Quantitative results.* We were able to find distinct parameter values for near-optimal behaviour for each of Experiments X1 – X3 and for each of their variants (see Fig. S3 for an example). For these values, the searching rule approached the optimum within 3-6 searches for all classes, giving positive answer on Question D. In terms of overall effectiveness (which includes also the searches during which the model relearns) the model’s behaviour was in all variants of Exps. X1 – X3 comparable to effectiveness of humans (Fig. 5). Actually, in most cases, the model’s behaviour was more effective than humans’ behaviour, giving positive answer on Question E.

Despite expectations, it was not possible to find one common parameter setting (Exp. X4) that would suit well for all the 25 variants of relearning (Fig. 5). The best

values found, that is  $a = 9.6$ ,  $b = \langle 1, 15 \rangle$ , produced behaviour that was worse (summing over all 25 variants) than average human behaviour by 56% and worse than the best distinct settings for SNT given by Eqs. (1), (2), (4) by 89%. For example, in Exp. X4, in the relearning variant “90%, container I1” → “90%, container J2,” where the optimal strategy switches from [I1, House] to [J2, House], the model adopted the rule [I1, J2, House] after the switch and only later it learned [J2, House]. It took a while to forget I1 because of how the learning rule (Eq. (2)) is constructed (see also Supplementary video 1 and Tab. S2). We will comment on this point in Sec. 5. In sum, the answer on Question F is negative.

*Qualitative results.* It was not possible to find parameters that would approximate well emergence of humans’ searching rules: humans behaved differently than the model. However, we identified a simple cause: humans used an additional heuristic to search at a place where the object was found last time. Only after the object was not found there, the subjects turned to their searching rules. When we added this heuristic to the model, the model’s searching rules were qualitatively similar to those of humans (see Tab. S2). Thus, we conclude the answer on Question G is positive.

To investigate the robustness, we also compared how some of the optimal models from Exp. X2 and X3 behave when number of containers changes (e.g. to 1, 4, 4, 3, 4, 6 for each respective room). These tests suggested that the SNT given by Eqs. (1), (2), (4) (i.e. X3) was more immune to this environmental change than that given by (1), (2), (3), presumably because the change is reflected by the *size* parameter included in Eq. (4). More thorough trialling on this issue would be needed in future.

*Summary.* The experiments revealed several important points: First, it is possible to parameterise the model so that its behaviour is similar to human behaviour, but only if the heuristic of looking at the place of the last object’s occurrence is added. Second, the effectiveness of the model is similar to or better than humans’ effectiveness. Third, the model as well as human are able to quickly relearn when the pattern of dynamic of the object changes. Finally, we were not able to find common parameters for all kinds of relearning, pointing to the necessity of extending the model with a mechanism estimating classes of objects’ dynamics.

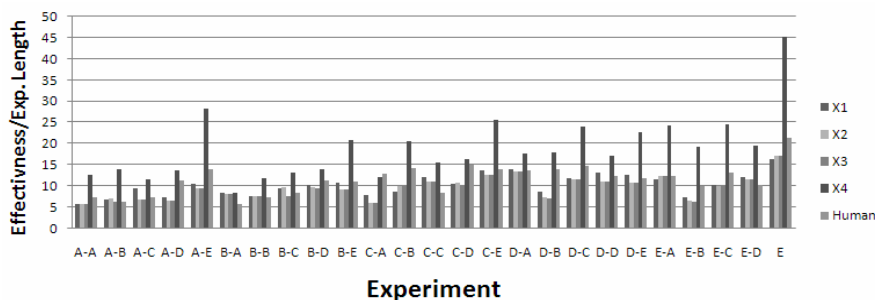


Fig. 5. The effectiveness for all cases of Experiment 2. A: “90%,” B: “3x30%,” C: “1/2x30%,” D: “1/1/1x30%,” E: “uniform.”

## 5 General discussion

The experiments showed that, in a middle-sized environment, searching rules emerge easily and the searching for objects is effective and comparable to the searching conducted by a human. However, there is a room for improvement. The subsequent discussion will be organised around following points: modulations and extensions of the basic network, the issue of automatic construction of abstract places, the question of plausibility of experimental scenarios, and the limitations of the network.

1) According with the current view held in psychology, which conceives spatial memory as a set of multiple interconnected systems, we suggest perceiving our network as a basic mechanism for “what-where” memory upon which other models can be layered modulating the basic mechanism. We have already seen one such modulation, quite important in fact: the heuristic of looking at the place of the last occurrence of an object. But there are also other possible modulations. We know that while the network behaves best when searching rules are computed using SNT with distinct parameters based on objects’ classes, which calls for a module for an automatic estimation of objects’ classes. Another already suggested module could help with intentional relearning; its function would be to increase the speed of learning temporarily due to “conscious” awareness of a change (“I moved the PC to a new room, I should remember it!”). Learning rate is a common parameter of neural networks, missing in our model. A module for making inferences based on general episodic (or semantic) knowledge could also improve the performance (“I have put my keys somewhere, but together with my wallet, hence finding the wallet will help me to find the keys.”). Our module can also serve as an input for other modules of an IVA’s mind, e.g. the emotion module. For instance, disappointment can be generated after an object has not been found at a highly probable place. The first mechanism – the heuristic – seems to be quite necessary. The decision whether to implement some of the other modules depends on the needs of a particular application.

2) Could the very structure of the environment, i.e. the place-nodes and the connections *up* and *down* between higher-order and less abstract locations be learned? There are many works on automatic, hierarchical clustering of space; the issue is whether the resulted clustering appears human-like (e.g. we need that several meaningful place-concepts develop for a working table, such as “in front of the monitor,” or “next to the mouse,” while none for the middle of a deserted corridor). We have actually already made a considerable progress on this issue, capitalising on the metaphor of place-cells, a neurobiological concept [17; see Suppl. video 2]. This work will be reported elsewhere in detail.

3) The experiment conducted with human subjects lasted for about 30 minutes. Thus, it tested short-term memory, while the model was engaged in long-term memory tasks. The heuristic revealed by the experiment with humans is valuable, but other human data should not be accepted as conclusive. A larger study would be needed; we would need more subjects acting in a same novel environment, e.g. employees after their company has moved or players of an on-line multiplayer game. A different possibility of validating the model would be to let human subjects to judge believability of behaviour of the artificial agent.

4) Will the memory work well in larger worlds? Here, the model has to be augmented with a module solving the travelling salesman problem with uncertainties.

Presently, we test the model in an environment of a small city, the description of which employs seven levels of abstraction.

5) There are several innate limitations of the model: First, it does not work with containers that can move including the objects located in them (a wallet with a credit card), with object that are semantically same but are being replaced very often (a new newspapers in the post box every day), with objects that can cease to exist (food), can be created, or otherwise can change their state, and with objects of the same kind with different dynamics (PC at home vs. PCs at a warehouse). While simple extensions of our model can be imagined for these situations, a radical extension would be need for another type of objects; those that can move themselves (a dog) (fortunately, this issue was already addressed in [11]). Second, relearning of some rules is relatively slow as demonstrated in Exp. 2. In terms of neural networks, Eq. (2) can be regarded as a link’s weight. In this terminology, weights grow and decrease linearly and without any bounds, implicating slow forgetting of a well learned weight. Humans claimed in our experiments that they considered only 3-4 last searches for the purposes of relearning, not the whole history, but recall that they conducted intermediate-term memory task. We already said that one way to forget more quickly is to increase the speed of learning temporarily. Another possibility is to change the learning rule. In fact, our present learning mechanism is a kind of Hebbian rule. Could a bounded Hebbian rule (see e.g. [5]) work better? This is our work in progress.

## 6 Conclusion

We have described an associative network addressing the issue of representation of “what-where” information that can evolve over long time periods. The mechanism is complementary to other models of spatial memory and cognition for IVAs, which tend to represent only topology of the environment or memorise locations of objects that never or only rarely change their positions.

The experiments showed that searching rules emerge easily and the searching for objects is effective and comparable to the searching conducted by a human. This means that the model can be integrated with other models, e.g. for the purposes of virtual companions or RPG agents. We do not expect any problem regarding the 3D scaling. The “take-home message” for researchers and practitioners willing to use this network is: a) the network works best in middle-sized environments when the heuristic of looking first at the place of last occurrence of an object is added, b) it would help if the parameters are adjusted for the purposes of a particular application; there is no miraculous setting that would work well for each and every case, c) it is better to scale the weights of the links to abstract places based on sizes of places not their level of abstraction, d) scaling the model to large environments, such as a city, would demand augmentation with a mechanism solving travelling salesman problem with uncertainties. This is our work in progress. Other work in progress includes learning the space abstractions automatically and changing the learning rule.

## References

1. ACT-R research group. Publications on Spatial Reasoning and Navigation. On-line: <http://act-r.psy.cmu.edu/publications/index.php?subtopic=23> [19.4.2009]
2. Bratman, M. E. *Intention, plans, and practical reason*. Cambridge, Mass: Harvard University Press (1987)
3. Brom C., Pešková K., Lukavský J.: What does your actor remember? Towards characters with a full episodic memory. In: Proc. ICVS, LNCS 4871, Springer-Verlag (2007)
4. Burgess, N.: Spatial memory: how egocentric and allocentric combine. In: *Trends Cognitive Sciences* 10 (12) (2006) 551-557
5. Dayan, P., Abbott, L. F.: *Theoretical Neuroscience*. The MIT Press (2001)
6. Dodd W.: The design of procedural, semantic, and episodic memory systems for a cognitive robot. Master thesis. Vanderbilt University, Nashville, Tennessee (2005)
7. Hamm, D.: Navigation Mesh Generation: An Empirical Approach. In: *AI Game Programming Wisdom 4*. Charles River Media (2008) 113-124
8. Hirtle, S. C., Jonides, J.: Evidence of hierarchies in cognitive maps. In: *Memory & Cognition* 13(3) (1985) 208-217
9. Ho, W., Dautenhahn, K., Nehaniv, C. Computational Memory Architectures for Autobiographic Agents Interacting in a Complex Virtual Environment. In: *Connection Science* 20(1) (2008)
10. Ho, W., Dias, J., Figueiredo, J., Paiva, A.: Agents that remember can tell stories: Integrating Autobiographic Memory into Emotional Agents. In: Proc. of AAMAS, ACM Press (2007)
11. Isla, D., Blumberg, B.: Object Persistence for Synthetic Creatures. In: *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems* (2002)
12. Kadlec, R., Gemrot, J., Burkert, O., Bída, M., Havlíček, Brom, C.: Pogamut 2 – A platform for fast development of virtual agents' behaviour. In: Proc. of CGAMES 07 (2007)
13. Kuipers, B. The Spatial Semantic Hierarchy. In: *Artificial Intelligence* 119 (2000) 191-233
14. Nelson, K.: The psychological and social origins of autobiographical memory. In: *Psychological Science* 4(1) (1993) 7-14
15. Noser, H., Renault, O., Thalmann, D., Magnenat Thalmann, N.: Navigation for Digital Actors based on Synthetic Vision, Memory and Learning. In: *Computer Graphics* 19(1) (1995) 7-19
16. Nuxoll, A.: *Enhancing Intelligent Agents with Episodic Memory*. PhD thesis, The University of Michigan (2007)
17. O'Keefe, J., Nadel, L. 1978. *Hippocampus as a cognitive map*. Oxford, UK: Clarendon Press (1978)
18. Peters, C., O'Sullivan, C.: Synthetic Vision and Memory for Autonomous Virtual Humans. In: *Computer Graphics Forum* 21(4) (2002) 743-752
19. Pöppel, E.: A hierarchical model of temporal perception. In: *Trends in Cognitive Sciences*. 1(2) (1997) 56-61
20. Strassner, J., Langer, M.: Virtual humans with personalized perception and dynamic levels of knowledge. In: *Comp. Anim. Virtual Worlds* (16) (2005) 331-342
21. Sturtevant, N. R., Buro, M. Partial Pathfinding Using Map Abstraction and Refinement. In *Proceedings of AAAI* (2005) 1392-1397
22. Taylor, H. A., Rapp, D. N.: Updating Human Spatial Memory. In: *Animal Spatial Cognition* (2006) On-line: [www.pigeon.psy.tufts.edu/asc/taylor/](http://www.pigeon.psy.tufts.edu/asc/taylor/) [19.4.2009]
23. Thomas, R., Donikian, S.: A spatial cognitive map and a human-like memory model dedicated to pedestrian navigation in virtual urban environments. In: *Spatial Cognition V, LNCS 4387* (2006) 421-436

24. Wiener, J. M., Lafon, M., Berthoz, A.: Path planning under spatial uncertainty. In: *memory and cognition* 36(3) (2008) 495-504

**Supplementary materials:**

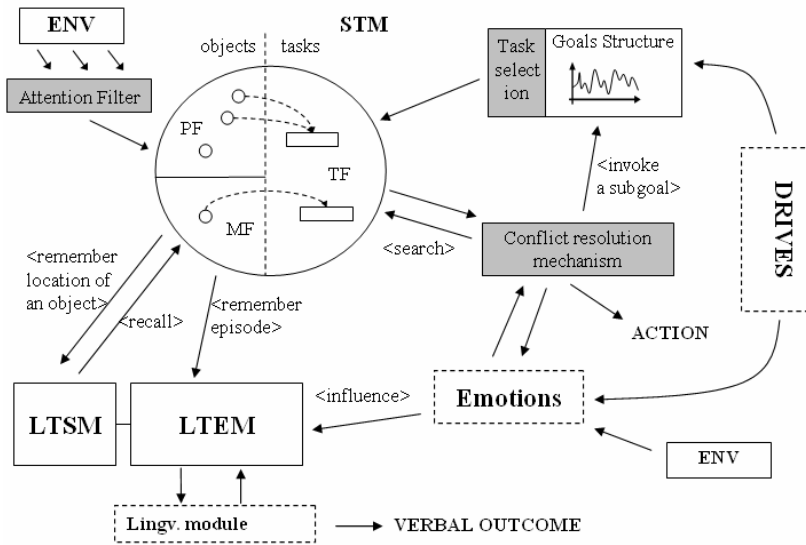


Fig. S1. The overall architecture of our agent. Note the perception—action cycle. ENV – the environment. PF – objects of the visual short-term memory (STM). MF – objects retrieved from the LTSM. TF – tasks field: tasks the agent is accomplishing at the moment. LTEM – the autobiographic memory. LTSM – the memory being investigated in the main paper.



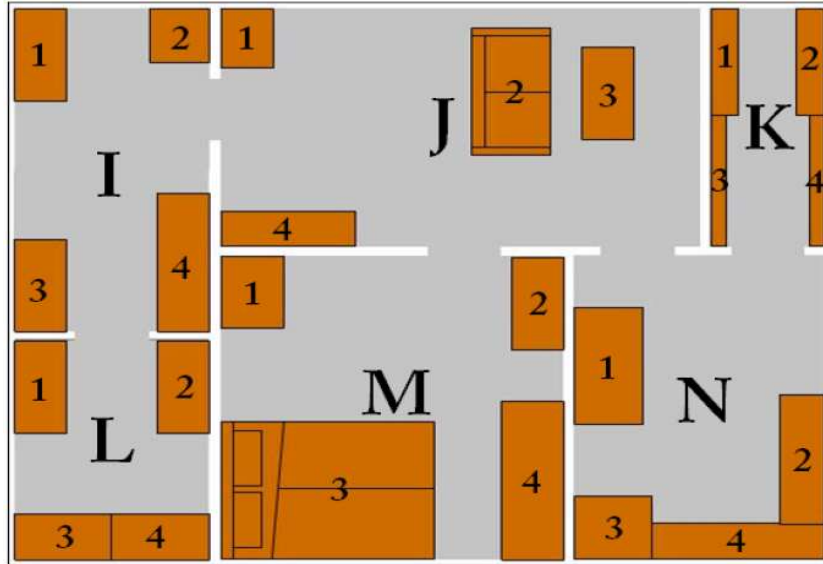


Fig. S2. The house in which the experiments were conducted: both on the agent as well as on the human subjects.

Process	Objects and quantity	Time (min)	Time of execution
Snacking	8xGrapes, 2xBowl	15	
Watering	4xCannikin, 12xRose, 3xWater	10	
Woodcutting	2xAxe, 12xWood	120	
Cooking & eating	6xMeat, 2xPot, 6xDinner Set, 3xKnife	180	
Conjuring	4xAmulet	30	
Lightening	2xTorch	10	
Reading	3xBook, 1xGlasses	60	
Repairing	4xPipe, 8xScrewdriver, 8xWrench	120	
Drinking	1xGlass, 2xMilk	5	
Sewing	3xLinen, 10xThread	120	
Smoking	1xTobacco Pipe	10	
Combing	1xComb	1	

Tab. S1. Example of one daily plan. Tasks, their sources, duration, and time of execution are depicted (the x-axis denotes time).

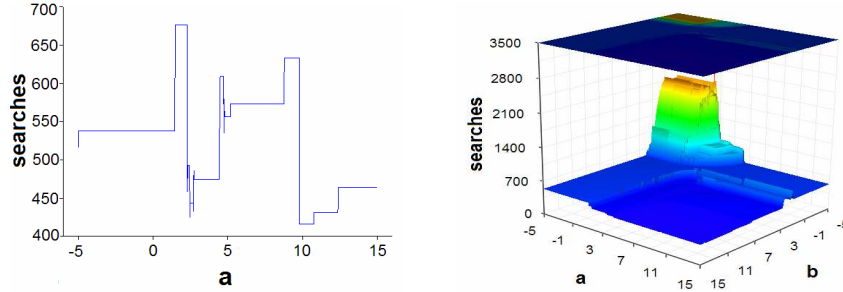


Fig. S3. (colour figure) The effectiveness for the case “1/1x30%” → “90%.” Left: Exp. X1 ( $b=2$ ). Right: Heat map for Exp. X2 (parameters  $a$  and  $b$ ).

	Human 1	Human 2	Human 3	Human 4	X3( $a=3.2, b=8$ )	X3(3.2,8) + Rule
L1	F	F	F	F	F	F
L1	L1, F	L1, F	L1, F	L1, F	L1, F	L1, F
L1	L1, F	L1, F	L1, F	L1, F	L1, F	L1, F
L1	L1, F	L1, F	L1, F	L1, F	L1, F	L1, F
L1	L1, F	L1, F	L1, F	L1, F	L1, F	L1, F
I2	L1, F	L1, F	L1, F	L1, F	L1, F	L1, F
L1	I2, L1, F	I2, L1, F	I2, L1, F	L1, F	L1, I2, F	I2, L1, F
L1	L1, F	L1, F	L1, F	L1, F	L1, I2, F	L1, I2, F
L1	L1, F	L1, F	L1, F	L1, F	L1, I2, F	L1, I2, F
L1	L1, F	L1, F	L1, F	L1, F	L1, I2, F	L1, I2, F
J2	L1, F	L1, F	L1, F	L1, F	L1, I2, F	L1, I2, F
K1	J2, L1, F	J2, L1, F	L1, F	L1, F	<i>LI, J2, I2, F</i>	J2, <i>LI, I2, F</i>
J3	K1, J2, L1, F	K1, L1, J2, F	K1, L1, F	L1, F	<i>LI, K1, J2, I2, F</i>	K1, <i>LI, J2, I2, F</i>
J2	J3, J2, L1, F	J3, J2, L1, F	J3, L1, F	J3, L1, F	<i>LI, J3, J2, K1, F</i>	J3, <i>LI, J2, K1, F</i>
J2	J2, L1, F	J2, J3, L1, F	J2, L1, F	J2, L1, F	<i>LI, J2, J3, K1, F</i>	J2, <i>LI, J3, K1, F</i>
J3	J2, L1, F	J2, L1, F	J2, L1, F	J2, L1, F	<i>LI, J2, J3, K1, F</i>	J2, <i>LI, J3, K1, F</i>
J2	J3, J2, F	J3, J2, L1, F	J3, J2, L1, F	J3, J2, L1, F	<i>LI, J2, J3, K1, F</i>	J3, <i>LI, J2, K1, F</i>
N1	J2, J3, F	J2, J3, F	J2, J3, F	J2, J3, L1, F	<i>LI, J2, J3, K1, F</i>	J2, <i>LI, J3, K1, F</i>
J2	N1, J2, J3, F	N1, J2, J3, F	N1, J2, J3, F	N1, J2, J3, F	<i>LI, J2, J3, N1, K1, F</i>	N1, <i>LI, J2, J3, K1, F</i>
K1	J2, J3, F	J2, J3, F	J2, J3, F	J2, J3, F	<i>LI, J2, J3, N1, K1, F</i>	J2, <i>LI, J3, N1, K1, F</i>
J3	K1, J2, J3, F	K1, J2, J3, F	K1, J2, J3, F	K1, J2, J3, F	<i>LI, K1, J2, J3, N1, F</i>	K1, <i>LI, J2, J3, N1, F</i>
J2	J3, J2, F	J2, J3, F	J3, J2, F	J2, J3, F	<i>LI, J2, J3, K1, N1, F</i>	J3, <i>LI, J2, K1, N1, F</i>
J2	J2, J3, F	J2, J3, F	J2, J3, F	J2, J3, F	<i>LI, J2, J3, K1, N1, F</i>	J2, <i>LI, J2, J3, K1, N1, F</i>
J3	J2, J3, F	J2, J3, F	J2, J3, F	J2, J3, F	<i>LI, J2, J3, K1, F</i>	J3, <i>LI, J2, J3, K1, F</i>
J3	J3, J2, F	J3, J2, F	J3, J2, F	J3, J2, F	<i>LI, J2, J3, K1, F</i>	J3, <i>LI, J2, K1, F</i>
K1	J3, J2, F	J3, J2, F	J3, J2, F	J3, J2, F	<i>LI, J2, J3, K1, F</i>	J3, <i>LI, J2, K1, F</i>
J4	K1, J3, J2, F	K1, J3, J2, F	K1, J2, J3, F	K1, J3, J2, F	<i>LI, J2, J3, K1, F</i>	K1, <i>LI, J2, J3, F</i>
J3	J4, J3, J2, F	J3, J2, F	J2, J3, J4, K1, F	J4, J3, J2, F	<i>LI, J2, J3, K1, F</i>	J4, <i>LI, J2, J3, K1, F</i>
J2	J3, J2, F	J3, J2, F	J3, J2, K1, F	J3, J2, F	J2, J3, <i>LI, K1, F</i>	J3, J2, <i>LI, K1, F</i>
K1	J3, J2, K1, F	J3, J2, F	J2, J3, K1, F	J2, J3, F	J2, J3, <i>LI, K1, F</i>	J2, J3, <i>LI, K1, F</i>

Tab. S2. Example of the model’s searching rules compared to those of humans for the case “90% [L1]” → “1/2x30% [J2, J3, K1].” The step of switch is highlighted. Notice, how long does it take to the model to abandon the L1 container in the 2<sup>nd</sup> part of the experiment (bold italic). The course of this experiment is also detailed in Supplementary video 1.