Charles University in Prague
Faculty of Mathematics and Physics

MASTER THESIS

Bc. Ondřej Burkert
Connectionist Model of Episodic Memory for Virtual Humans
Department of Software and Computer Science Education
Supervisor: Mgr. Cyril Brom Ph.D.
Study Program: Informatics, Theoretical Computer Science (Artificial Intelligence)

Acknowledgements

I would like to thank my master thesis leader Mgr. Cyril Brom PhD. for his insight and the effort he made to allow the completion and success of software project Pogamut 2 and consequently this work. I would like to express my gratitude to my friends Pavel Češka and Michal Bída who gave me the essential feedback on this text.

I hereby declare, that I wrote the master thesis autonomously using exclusively the sources listed in the bibliography. I approve its lending.

In Prague at 16 April 2009 Ondřej Burkert

Never say, that something is not possible.
There is always someone who does not know that and put it into life.
(source unknown)

Table of contents

Název práce: Konekcionistický model epizodické paměti pro virtuální lidi
Autor: Ondřej Burkert
Katedra (ústav): Kabinet software a výuky informatiky
Vedoucí diplomové práce: Mgr. Cyril Brom, Ph.D.
e-mail vedoucího: brom@ksvi.mff.cuni.cz
Abstrakt: Cílem práce je návrh a implementace prototypu epizodické paměti pro virtuálního člověka. Tato paměť je inspirována dostupnými znalostmi o lidském způsobu vnímání běhu času a fungování lidské paměti pro osobní zážitky (epizody). V práci shrneme relevantní poznatky z neurobiologie a kognitivní psychologie a na tomto teoretickém základu postavíme model paměti. Vycházeli jsme z rozhodovacího mechanizmu a modelu epizodické paměti Peškové. Rozhodovací mechanizmus je založen na BDI, teorii afordancí a AND-OR stromech. Epizodická paměť propojuje stromy časovými ukazateli. Stávající paměťový systém trpěl nedostatky stran časově specifikovaných otázek. Proto navrhovaný model pracuje s unikátním podsystémem pro vnímání času, který umožňuje realističtější ukládání a vyvolávání minulých událostí. Agent vybavený tímto modelem paměti je například schopen odpovědět na otázku, co dělal minulý týden po odpoledních, apod. Prototyp byl implementován na platformě Pogamut 2. Pogamut 2 je napojen na prostředí komplexního spojitého 3D světa hry Unreal Tournament 2004, což nám umožnilo ověřit chování modelu ve složitém prostředí. Následně jsme provedli sérii experimentů. Výsledky ukázaly, že navržený model opravdu rozšiřuje agentovy kognitivní schopnosti chápat časové koncepty, což následně umožňuje správně odpovědět přes vágní časovou specifikaci dotazu. Paměť má také limitovanou schopnost slévat podobné episody.
Klíčová slova: virtuální lidé, epizodická paměť, vnímání času

Title: Connectionist Model of Episodic Memory for Virtual Humans
Author: Ondřej Burkert
Department: Department of Software and Computer Science Education
Supervisor: Mgr. Cyril Brom, Ph.D.
Supervisor's e-mail address: brom@ksvi.mff.cuni.cz
Abstract: The goal of this work is to design and implement a prototype of the episodic memory for virtual humans. The memory is inspired by up to date research on function of human memory for personal events (episodes) and human time perception. We design a model of memory based on this theoretical knowledge. We took as the point of departure episodic memory system and decision making system of Peskova. The decision making system is based on the BDI, theory of affordances and AND-OR trees. The former episodic memory suffered deficiencies in the recall for time-cued questions. Thus proposed model is working with a unique subsystem for the time perception which allows for more realistic storage and recall of past events. The agent enhanced by this model can reply to questions like "What did you do last week afternoons?". The prototype is programmed in Java using the framework Pogamut 2. Pogamut 2 is connected to the complex continuous 3D world of Unreal Tournament 2004 which allows us to verify the design in the challenging environment. We have conducted several experiments. The results show that the model extends agents cognitive abilities with a capability to understand socially established temporal patterns. That allows him to answer to the questions with a vaguely specified time information. Moreover, the memory has a limited capability to blend similar episodes together.
Keywords: Virtual humans, episodic memory, perception of time

1 Introduction

The field of artificial intelligence draws often the inspiration from biology, psychology and natural sciences. Evolutionary techniques are based on Darwin's theory, neural networks use the knowledge about signal propagation in the brain and virtual agents become more and more plausible and believable thanks to discoveries in the area of cognitive psychology, sociology or neurobiology. We will pursue the research on virtual agents, more specifically we are going to introduce an enhanced model of episodic memory for them.

Intelligent virtual agent (IVA) is an embodied agent which has a graphical representation in the environment. Nowadays, there are many applications featuring IVAs including commercial games [1], serious games [2], therapeutic tools [3] and virtual storytelling [4]. Users of those applications expect agents to behave in an intelligent way. They must be believable – they should be able to act in their environment in compliance with user expectations. The pursuit of believability (Loyall [5]) puts high requirements on IVAs. There are many subtasks to settle such as the problem of path finding, emotion modeling, linguistic modules, decision making systems, etc. Nuxoll's comprehensive study [6] brings forward a role of episodic memory for IVAs. He stated that an episodic memory system can increase believability while boosting learning algorithms and improving agent's overall performance.

Episodic memory was first introduced in psychology by Endel Tulving in 1972 [7]. He divided our memory system into three modules – procedural memory, semantic memory and episodic memory. The procedural memory represents mainly learned manual or physical activities. We can throw a ball without thinking consciously about the exact hand movement and we can improve our throwing skills by repetitive exercises. The semantic memory keeps facts – all pieces of information which are not connected to a particular context. For instance, we know that New York has a lots of inhabitants but we have not counted them or we know that there are eight planets orbiting the Sun but we have never seen them in person. The episodic memory on the other hand stores our personal experience and memories. It helps us a lot in our everyday life. Our decisions are often based on information retrieved from the episodic memory. We are able to find our home and work, recognize our family, colleagues and friends, retell what happened to us and learn lessons from our failures and mistakes.

As was discussed by Nuxoll [6], the IVA enhanced with a model of episodic memory will be apt to: detect repetition, distinguish what is important, tell stories, explain its behavior, predict the outcome of the action, etc. On contrary nowadays, agents are for most part scripted their responses are hard-coded thus their believability is limited because they behave in the same way all over again. Thus it appears that a model of episodic memory would be a great advancement in the research and use of IVAs. Moreover, the episodic memory could help solve the problem with limited resources. We cannot keep the entire history of an agent in the memory as a log. It would be extremely time-demanding for retrieval and space-demanding for storage. That leads us inevitably to the question how to design and implement such a model.

The design and implementation of a domain independent model which would account for all features of episodic memory described in [6] would be a herculean task. Hence it is not an intent of this work to present the ultimate all-covering model of episodic memory but to advance current research.

We will follow up on the work of Peskova [8]. She introduced an episodic memory model which worked with a decision making system (DMS) based on AND-OR trees, theory of affordances [9] and the BDI (**B**elief, **D**esires, **I**ntentions) [10] which is a widely accepted paradigm for the design of action selection mechanisms of cognitive agents. The AND-OR trees are representing desires (goals) from BDI. Each tree represents a behavior for a goal. It defines a set of possible ways to achieve the goal by a hierarchy of subtasks and subgoals. The activation of desires is determined by the plan. Beliefs are the facts which can be retrieved from the the memory.

We will provide a comprehensive presentation of the DMS later. The most important part of Peskova's model was of course the episodic memory. It enhanced an agent with (a) the resource lookup mechanism which allowed him fast location of resources necessary for his actions and (b) the storing and retrieval algorithm which allowed for autobiographical memory. The episodes were stored using time pointers which defined the exact order of actions in the past. We can imagine time pointers as a string weaved into the forest of AND-OR trees. The agent was able to reply to questions like: "What did you do between time A and B?". The recall was done by a look-up of the time pointer corresponding to the time A and then following the string of time pointers to the time B.

The mechanism has some deficiencies. The time information is exact (1). The result of a query is exactly what is stored in the memory (2) – like a video recorder – neither it is error prone ("I think I was studying yesterday afternoon, but in fact, it was yesterday evening."), nor similar episodes can be blended ("I was studying every afternoon two weeks ago, I remember some details but I can't say to which day they belong."). The time information does not decay (3) ("I can recall I was working a lot, but I'm not sure whether it was three or four days ago.").

The proposed model should be able to deal with those problems. The problem (1) will be addressed by introduction of the time perception module. This module will enhance an agent with internal *time concepts* which will cluster time into longer periods creating the notion of morning, afternoon, etc. We will tackle problems (2) and (3) with a connectionist network between activities and time information (time concepts, days, weeks).

This work is organized as follows. Chapter 2 outlines the introduction into the psychological background of the memory in general, the episodic memory and the perception of time. Following chapter 3 discusses in depth our motivation for the research, expected outcome and the summarization of related works. Then, chapter 4 presents the model for episodic memory, its key components and a reasoning about crucial design decisions. Chapter 5 describes used environment, development tools and simulation specifics. Next chapter 6 is dedicated to the methodology used in experiments and the questionnaire inquiry we carried out to gather a set of usual questions for an agent. Chapter 7 submits description of experiments, their results and the consecutive discussion. The work is concluded with future works (chapter 8) and conclusion (chapter 9).

2 Psychological Background

The AI of IVA's is closely connected with neurobiology and cognitive psychology. Even though we are not aware of a psychological model of human memory that would explain all phenomena and give a guidance for implementation, we can use some notions, metaphors and empirical evidence from human related research for the design of the model.

The origin of human memory compelled researchers from late 19th century and many different models and theories emerged from their research. For the sake of brevity, we are going to review only the latest models and notions we used.

A good starting point is the classification of memory. Psychology usually distinguish three different types of memory regarding its persistence. Those are sensoric, short-term and long-term memories. The sensoric memory keeps a snapshot of our perception before it enters into the short-term memory for processing – for instance, if we move a torch very fast in the dark we can see its trace though the light is not there anymore. The short-term memory stores information from last few seconds to few minutes. For example, if someone address us a question while we are not paying attention, we can use this memory to recall the question and then respond to it. The information from the previous chapter can be retrieved from this memory as well if we assume that the reader reads continuously. The least but definitely not last is the long-term memory (LTM). This system represents what we usually call "memory". Humans would not be able to operate in everyday life without it. It stores all important information like names, locations, skills or any other knowledge. We will now dig deeper into the LTM.

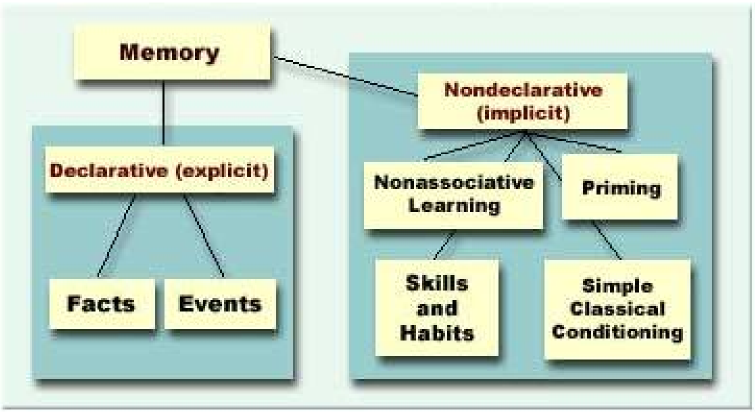


Fig. 1. Larry Squire's [11] memory taxonomy which divides memory to declarative and nondeclarative parts.

LTM accounts for many different phenomena which suggests the presence of modular system (see fig. 1). Our main interest lies in the field of declarative memory thus we will omit the non-declarative memory and address directly to semantic and episodic memories. The distinction between the semantic and the episodic memory was first introduced by Endel Tulving in 1972 [7]. In short, the semantic memory represents facts while the episodic memory represents personally related events and experience. Unfortunately they are not disjunct. The answer to the same question can be retrieved using one of those or both. For instance, we can remember certain semantic information thanks to an episode which helps us remembering it.

The retrieval from the memory can be either intentional – we are asked to recall some particular record – or unintentional – we recall a record just because some thought or stimulus triggered it. When using the intentional retrieval we often approach the problem by reconstructing the demanded record using a number of different cues [12]. For instance, we have left our glasses somewhere. We query our memory for the location of the glasses. The recall starts usually with the question: "Where have I put them?" which leads to "When was the last time I had them on?", etc. That brings up an episode of us preparing to go to bed, drinking some fresh beverage from the fridge while leaving the glasses on the shelf inside. The time is often one of the cues supplied to the episodic memory for the query. The role of time will be discussed in detail in section 2.2.

Another compelling property of human memory is its capacity. There is an evidence [13] that humans can successfully recall memories which are more than 60 years old with a remarkable precision of time information as well as abundance of details related to it. Our memory system is able to keep a vast amount of semantic data as well as a variety of stories ranging from our childhood to yesterday. Researchers tried to measure the capacity of the memory but there is no evidence so far which would find its limits. Moreover, it is difficult to test LTM as people often create false memories [14], which could be, for instance, phantoms of real memories, which were altered during some memory processes. We will speak about processes involved in the memory creation and consolidation in the section 2.1.

Another question is whether we can draw some real implementable inspiration from human memory functionality. The neuroscience still cannot answer reliably the question how exacly are things stored in our brain. For instance, if we want to know, what should be the right answer of our agent to some question, a neuropsychologist would likely say that it depends on many different factors like agent's internal context, gender, emotions and age. Thus we can only use the phenomenological evidence of what our memory is capable of and draw inspiration from theories and experiments.

2.1 Episodic Memory

The episodic memory is a storage of personally related events and experience. It is a vital part of our memory system and we use it in our everyday life. It is the key to many human cognitive functions (adapted from Nuxoll [6]):

- **Sensing:**
  - **Noticing Significant Input** – the ability to determine what is important in the situation by its relative familiarity.
  - **Detecting Repetition** – the ability to notice that something is happening again allowing then better outcome prediction for example.
  - **Virtual Sensing** – for instance, our ability to navigate in the environment is based on the internal representation of space, that allows us to recall where a street heading to etc.
- **Reasoning:**
  - **Action Modeling** – prediction of outcome of an action.
  - **Environment Modeling** – prediction of changes of environment.

- **Recording Previous Success/Failures** – helps to improve our future performance.
- **Managing Long Term Goals** – people are able of long term planning which differs them from animals. The episodic memory helps keeping track of accomplished subgoals.
- **Sense of Identity** – understanding own behavior in context of other people.

- **Learning:**
  - **Retroactive Learning** – learning from past experience.
  - **Reanalysis of Obtained New Knowledge** – relearning from an experience which outcome was altered by a recently acquired knowledge.
  - **Explain Behavior** – to retell past actions to allow for experience sharing.
  - **“Boost” to Other Learning Mechanisms** - the cognitive psychology provides a several techniques we can use while learning a lot of data, if we manage to learn them in some varied, interesting contexts, we can expect better retrieval as the episodic memory stores them more reliably.

The impact of the full episodic memory on the agent’s behavior is enormous. Unfortunately, despite many theories and studies [15], there has not been established any all-covering theory which would completely explain the origin and functionality of human episodic memory. Nevertheless, we can exploit existing theories and use them in the design. We will shortly review theories of how the episodes are stored, updated, forgotten and retrieved.

Psychological studies suggest that the storage of certain fact or episode is not done simply by coding its content into the brain. Firstly, there is a process which acts during the construction of traces for the episode in the memory. It alters a story while saving it. Bartlett (1932) [16] showed that different people recall different details from the same text which implies that they infused stored record with their own schemata and structures. Secondly, there is a process which is active during the recall. It does not only reconstruct an episode according to the traces from its creation. It can work in the constructive fashion adding new information or altering existing information hence making it compliant with the current base of knowledge of the person. Alternation of memories can occur as well via interference. The interference happens when there are two memories competing for the same space, which can result in alternation of records as well as in the oblivion of weaker one.

There are two basic theories about forgetting: already mentioned interference and extinction. The extinction theory asserts that the memory traces for the event slowly extinct over the time.

Forgetting brings forward another very important aspect of memory – the influence of contextual information. Many studies [17] showed that people recall better episodes with richer contextual information. Moreover, those episodes persist longer. As was discussed in [18] people are able to notice and retain a remarkable quantity of details when they are in some unusual conditions – life-threatening or emotionally extreme situations, etc. Those contextual information as well as the apparently specific internal context helps maintaining the information.

Context cues are then used during the recall. We do not always query our memory for a certain record. It often happens that some memory occurs to us unconsciously on the pretext of the context we are in. For instance, if a pick-pocketer stole us our purse at a certain square then we enter the place again, it can trigger that episode and we will pay more attention to our possessions.

One of the contextual information as well as the recall cue is the time. We are particularly interested in the development of a model for time-representation and time-perception. We will discuss corresponding theories in the following section.

2.2 Memory for Time

We have stated in the introduction that former Peskova's model has some deficiencies concerning time. More specifically agent dates things exactly or rather exceedingly accurately. He can say that something happened at 13:15, but he has no notion of ordinary temporal concepts like morning, afternoon or night. He does not have a human-like perception of time. Before we submerge into the proposed model for modeling time perception we should review known theories about human time perception. There are several different theories (adapted from Friedman [19]). We will shortly review those as we are going to refer to them in the following text.

Distance-based Theories:

The strength theory is based on the notion that every event in our life has a strength that declines over the time through either decay or interference with the subsequent event. When we query our memory for the time when something happened. The response is given according to the strength of a trace attached to the record.

The chronological organization theory states that events are organized in the order of occurrence. We can use the metaphor of moving conveyer belt on which we put the event as a bag when we store it. Then the time distance of a record is guessed according to the distance on this belt.

Location-based Theories

The time-tagging theory presumes that each memory is tagged with information about its emplacement in time. The time of occurrence is determined by retrieving information from this tag. That puts forward an interesting notion of landmark. A landmark is compliant with the time-tagging theory as it is an event for which we remember exactly the date and time. Landmarks are usually remarkable, often revised and retold events like a birth of a child, the end of the university studies, a wedding, etc. We use them often to anchor other stories in the flow of time.

The reconstruction theory does a side step from the previous theories saying that the linearity of time is mere illusion and the perception of time and recalls of exact times of memories are based more on the contextual information stored along with the memory of the event. Friedman's experiments [20] show that we use contextual information stored with the event to refine our time estimates using temporal patterns. For instance, if the recalled event is featuring snow, we know that it happened in the winter narrowing months down to December, January, February and March. When we recall some story from high school studies, we know that we were 16 to 19 years old which helps determining the year.

The supporting evidence for the reconstruction theory can be found in work of Larsen, Thompson and Hansen [21]. They developed the idea of cyclic precision providing evidence that people can often be very accurate in one cyclic scale (hours, days, weeks, months, years) while being wrong in another. For instance, people can remember exact time, day of a week and year for the event while being wrong on the month scale.

Relative Time-based Theories:

The order codes are defining the order of different events. When combined with landmarks they can give another point of view on when something happened.

Conclusion.

Friedman [18] concluded that none of these theories can consistently account for all phenomena of human time perception thought it seems that our mind combines some of these theories together to get to the final time estimate for a past activity. Friedman even proposes the work-flow for the time estimate retrieval.

We are not going as far as Friedman as we have still in mind a demand on simplicity for the time perception model for virtual agent. It is desirable that we conclude the work by the implementation of a working prototype. Nevertheless, there are some implications to the model based on earlier-presented theories. First of all, people construct time patterns and concepts to help themselves locating events in the flow of time. Those concepts can be either days, parts of a day, seasons of the year, etc. Second of all, when people query the memory they search for different time-scales relatively separately which suggests to store different time-scales separately. Then the strength theory is a dead end if we want some accurate time estimates. On the other hand, since our agent does not have to be ecologically plausible we can use the analogy of a conveyer belt from chronological organization theory to mimic forgetting (by discarding old bags).

3 Motivation and Related Works

We have roughly outlined our main interests in the introduction. This chapter will be dedicated to the explanation of the motivation for the research accompanied with the comprehensive description of the context of the work in the ongoing research. The recently created AMIS [22] group at the faculty of mathematics and physics of Charles University, led by Mgr. Cyril Brom Ph.D., is working on the problem of believable virtual agents. That covers many different areas from AI, cognitive psychology, sociology and neuroscience and there are many possible research paths to follow in each domain.

One part of the problem is the memory in general. Nuxoll [6] argued in his dissertation that the episodic memory is a powerful instrument for IVAs enhancing them with many useful skills and abilities. There is just one problem. The definition of exact requirements on the full episodic memory (FEM) has not been established yet. Brom tried to address this issue in [23]. According to his findings the problem is twofold. Firstly, we cannot take the direct inspiration from the psychology and neurobiology as the up to date findings do not provide a comprehensive and implementable model of episodic memory. There are a lots of useful metaphors and evidence, nevertheless, we do not know how our episodic memory works. But we can use relevant findings to constraint our research of artificial episodic memory. Secondly, we pursue a goal of believable IVAs which should not be confounded with the goal of a faithful human simulation. Therefore we can think out of the box while trying to mimic properties of human episodic memory.

Unfortunately, the related work in the area of artificial episodic memory is rather scarce. Apart from the work of Nuxoll and Peskova we have found only domain specific models like FearNot! [3] (a therapeutic application focused on helping bullied children) or work of Johnson (1994) [24] and Dodd (2005) [25], Ho [35].

Much more related work was presented by Peskova [8]. Actually her work is a direct predecessor of our research. Peskova designed and then implemented a prototype of an agent enhanced with the episodic memory. The agent was living in a simplified discrete 2D world composed of 9 interconnected rooms. He was performing planned tasks or activities which were triggered by interesting items. There were two main goals of the project: first was to show, that an agent enhanced with an episodic memory can perform better while locating the resources necessary for desired activity. Second goal was to enrich the agent with a capability to tell stories from his past. We will briefly describe the model (fig. 2).

The action selection mechanism followed widely used BDI notion (Bratman [10]). Agent's believes were obtained from the memory module. Desires were modeled by a scheduler, which defined activation for each root desire. The agent was equipped with a set of available behaviors (AND-OR trees) to satisfy his desires.

The agent was ruled by a hierarchical reactive planning. Available behaviors (desires) were represented by *AND-OR trees*. The notion of AND-OR tree can be explained by a simple example. Let's imagine that we want to eat. We have various options how to achieve the goal of eating (the root *desire – OR node*). We can go to a restaurant, we can cook something or just eat an apple. Each of those possibilities is represented by a *task – AND node*. A task ends successfully if all of its subgoals (OR nodes) ended successfully. For instance, we want to eat at the restaurant. That demands us to locate an appropriate restaurant, order food, eat it and pay before we leave. Thus an AND-OR tree is representing a hierarchical decomposition of the goal providing agent with various routes how to achieve it.

Every task may require acquisition of some *resources* (objects, places) before it can be performed. The task which cannot be further decomposed is called *atomic action*.

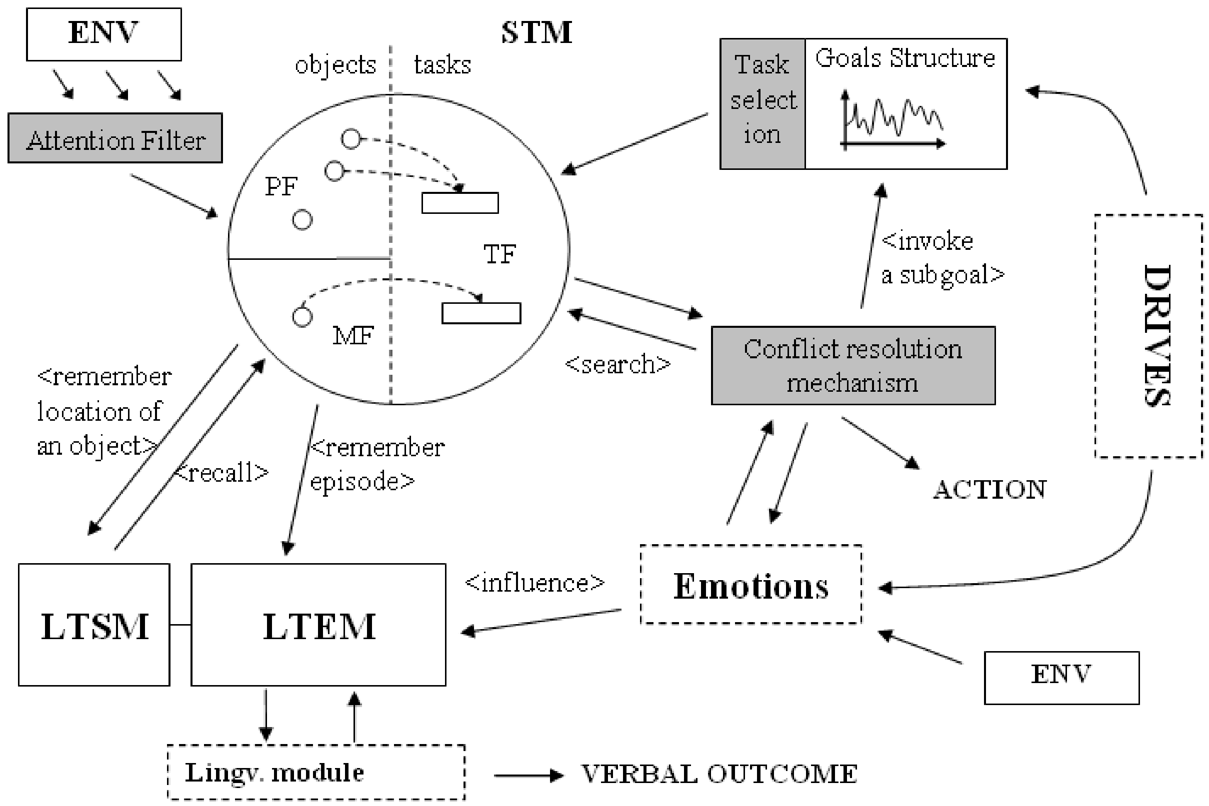


Fig. 2. Architecture of Peskova's agent – adopted from Brom [23]. The short term memory (STM) stores in perception field (PF) information about the perceived environment (ENV). The memory field (MF) stores results of memory queries (virtual sensing) from long-term episodic memory (LTEM) and long-term semantic memory (LTSM). The agent is driven by drives. The pre-active tasks are stored in the task field (TF).

The competition between tasks is done by a comparison of their activities. The activation can be either defined by the plan or by the attractiveness of an interesting item which triggered the action. One task can be interrupted by another more active task. The interrupted task can be resumed when the interrupting task terminates or can fadeout if it is not used for some time.

The most intriguing part of Peskova's work was the *long-term episodic memory* (LTEM). The LTEM stores all trees which represent all possible actions, events. They are interconnected by *time pointers*. Time pointer leads from one activity (node of a tree) to another and have the time of addition attached to it. The sequence of time pointers creates a string of time which is woven into the forest of desire trees. Moreover, the choice of AND-OR trees as a representation of behaviors allows for forgetting unimportant details of past activities (lower levels of the tree). The agent can then reply to simple time-cued questions like "What did you do between time A and B?". The recall is effected by a location of the time pointer at time A and then following the string of time pointers to the time B.

There are several problems connected with the autobiographic component of Peskova's LTEM. Firstly, time-pointers are weaved in the structure of AND-OR trees with an exact time information on them. As we know from the time perception theory we usually do not expect an agent to answer to questions like "What did you do yesterday at 13:15?" but rather to answer a question which include a vaguely defined time pattern like afternoon. Secondly, the agent can forget details of an episode thanks to the hierarchical structure of his plans but he can never blend two similar episodes into one. For instance, agent is studying regularly every morning for a week as he is preparing for an exam. Few weeks later this should be blended into just one episode with less exact time information. Like „I remember I was studying every morning that week, but I can't recall at what time I started each day. Thirdly, the time information does not decay over time. It is either there, exact as recorded, either not there forgotten with the whole episode. The agent cannot reply that he did something last week but is not sure which day exactly it was.

Sketching out the problem leads us ultimately to the solution which will be presented in the next chapter. In general our response to problems described in previous paragraph will be to enhance the agent with time perception which will consequently allow for desired phenomena of vague dating and forgetting.

4 Model Definition

We have briefly summed up the theoretical basis and background for our model of episodic memory thus we can step forward to the model definition. This chapter will be dedicated to the description of the design of the agent and its episodic memory system accompanied with the discussion of important design choices.

The DMS of an agent was adapted from Peskova (see fig. 2 for model overview), reimplemented from Python to Java, debugged and then deployed in much more complex environment. The agent is driven by a hierarchical reactive planning. His behaviors – desires – are represented by a set of AND-OR trees and the action selection is done via competition between goals where the most active goal is executed. We will describe our adaptation of the model in the first section of this chapter. We should also note here that we have focused on the problem of episodic memory thus the elaboration of the DMS is limited to the extent necessary for our experiments.

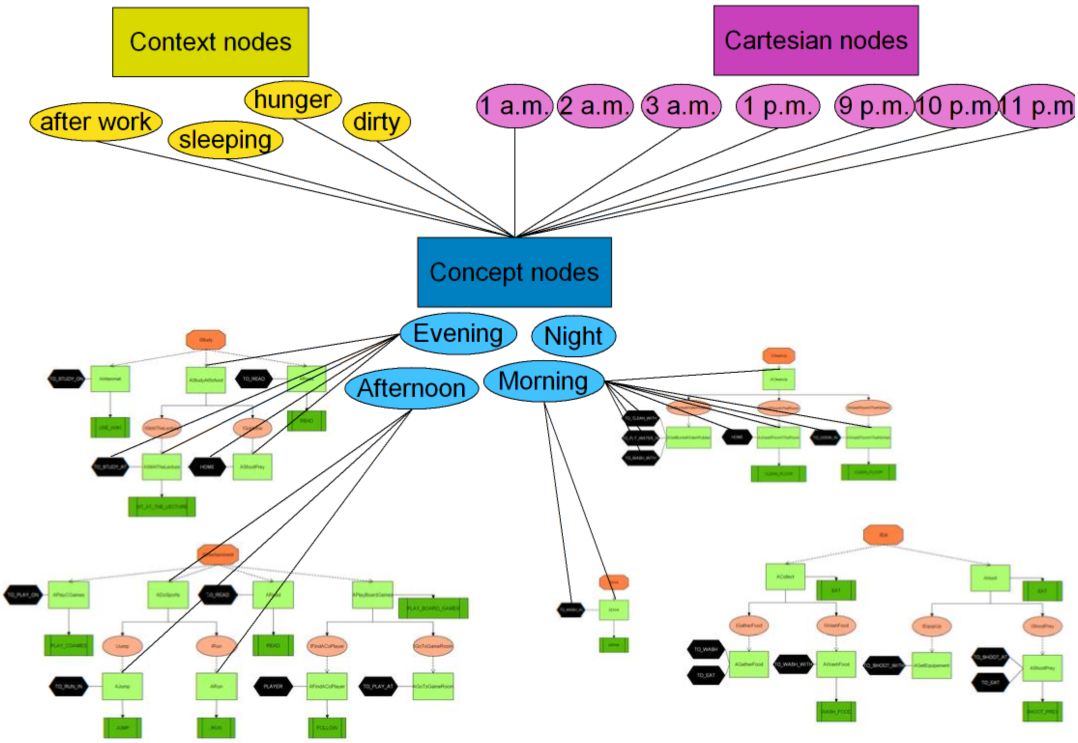


Fig. 3. Episodic memory design outline. The context layer (top left) and cartesian layer (top right) are connected with concept nodes (center). Concept nodes are connected with AND-OR trees of desires (at the bottom).

The episodic memory is divided into two main modules – item memory and neural memory. The item memory is rather simple and will be described in the section 4.2. Its purpose is to enhance an agent with information about whereabouts of resources (items, places, other agents) to speed up their lookup. The neural memory is much more complicated. In short, it is composed from three layers of "neurons" which are interconnected by weighted links (fig. 3). First layer consists of *context nodes* which describe agent's internal or external state (context) and *cartesian nodes* which represent time or biorhythms (their activation changes regularly over time). First layer is connected with the layer of time concepts. *Concept nodes* are associating together particular context of agent with some period of time thus representing time concepts. Agent learns them automatically during the simulation. They provide him a vaguer notion of the time. They allows him to answer questions which contain for instance a part of a day (e. g. morning) as the time information.

The next layer of nodes consists of AND-OR trees which are linked together with objects (items, places) and time nodes – either concept nodes, either nodes for days, weeks, etc.

Now we will dig deeper into the model definition. First we will introduce our adaptation of Peskova's DMS. Then we will present the item memory and we will conclude with the comprehensive description of time representation and the connectionist memory.

4.1 Decision making system

The decision making system (DMS) was adopted from Peskova (for more information please consult [8]). We will give here a brief introduction into its features and functionality. It is based on the theory of affordances (Gibson [9]), the notion of believes, desires and intentions (BDI) and a hierarchical representation of desires by AND-OR trees.

Theory of Affordances.

An *affordance* is a feature of an entity (object, place) which defines for what actions we can use it. The theory of affordance is based on the observation of how do we locate and work with resources for our actions. When we observe surrounding environment, we do not treat items according to their physical characteristics like height, weight, volume, exact shape or color, but according to the set of possible actions we can do with them. For instance, if we want to eat something, we can look either for some particular object or for anything "eatable". The eatability is then the affordance of apples, hot-dogs, lasagne's, etc.

We can easily assign various sets of affordances to every item, place or agent which are present in the environment. When we apply this theory we do not only enhance an agent with a semantic information about his environment but it also makes every extension and change to the environment or to the set of agent's actions easier. The agent can successfully work in another environment if we define him a new assignment of affordances to resources. We can give him another item to use without the need to teach him how to use it. All we have to do is to provide the new item with a set of affordances.

Beliefs Desires Intentions (BDI).

BDI is a well-established and widely used paradigm for the design of cognitive agents. As the name suggests it involves three different entities: *believes*, *desires* and *intentions*. *Beliefs* represent the informational state of the agent – his beliefs about the surrounding world. Beliefs do not necessarily represent facts as they do not have to be correct. In our case beliefs are represented by the assignment of affordances to resources and locations of objects which can indeed change over time. *Desires* mirror goals an agent want to achieve. Each desire is represented by an AND-OR tree. We model the priority of desires during a day by the *day master plan*. *Intentions* are desires to which the agent has committed. Therefore intentions are particular plans of how to achieve the goal. Those plans are not static as they are defined by AND-OR trees and agent can choose one of the possible paths to fulfill the goal. Moreover, the tree is infused with dynamically added intentions which are responsible for the retrieval of resources.

4.1.1 Functionality of the DMS

First we will explain how AND-OR trees work, then we will proceed to the action selection and the work of short term memory (perceptive field, process area and virtual sensing), inventory and scheduler. We will conclude with an overview of the architecture.

AND-OR Trees.

Trees are used as the representation of plans which lead to satisfied desire. They are representing the decomposition of the goal into the hierarchy of sub-tasks. Following a plan means executing actions in the hierarchy defined by the tree. AND-OR trees are composed from AND and OR layers (fig. 4).

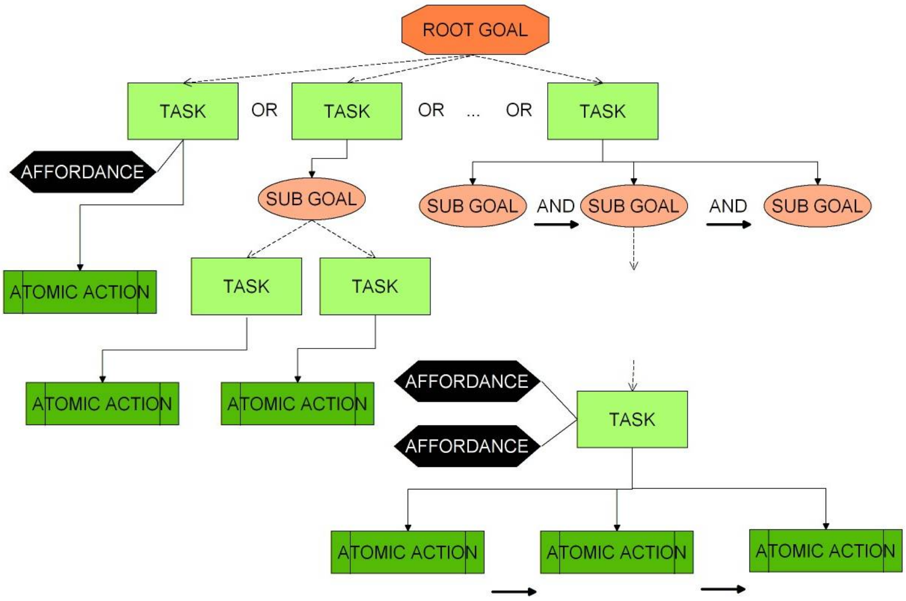


Fig. 4. The general structure of AND-OR tree. The root desire – the root goal – can have several options (tasks) how it can be satisfied. Each task can be further decomposed into subgoals. The task terminates successfully if it accomplishes all subgoals, localize the resources (affordances) and executes atomic actions.

A *goal* – OR node – can contain several *tasks* – AND nodes – which represent different possibilities how to fulfill it. The success of a goal means that execution of one of its subtasks succeeded. The goal fails when the agent had tried all available subtasks and they all failed.

AND nodes are represented by tasks. Each task can contain a set of subgoals, affordances and atomic actions. Three conditions must be met before the task terminates successfully. Firstly, all of its subgoals must succeed. Secondly, it must locate resources for all of its affordances. Thirdly, it must perform all *atomic actions*. For example, the task *sleep* has just the atomic action *sleep* and the affordance *to_sleep_in*. It will fail only if it cannot locate a resource with the affordance or if the atomic action fails and it will succeed in the other case (for more complicated example of a tree see fig. 5).

Localization of resources brings us to the *Want* goal. It is a special type of goal which is dynamically infused into the set of goals of a task due to the localization of its resources (affordances). As there is only one goal being pursued every moment, it looks for one affordance at the time. Want contains four tasks – actions: *Search Pocket*, *Search Environment*, *Search Memory* and *Search Random*. Those tasks are executed in the presented order (which differs from the action selection for general goals). First action looks in the agent's inventory. Second turns agent around so it can notice the item in his neighborhood. *Search Memory* looks up occurrences of items with the affordance in the item memory. This query returns usually few places with different credibility and then agent runs to the most credible place (or visit places with credibility higher than some bias). *Search Random* searches randomly the environment. Thus affordances represent dynamically added Want desires.

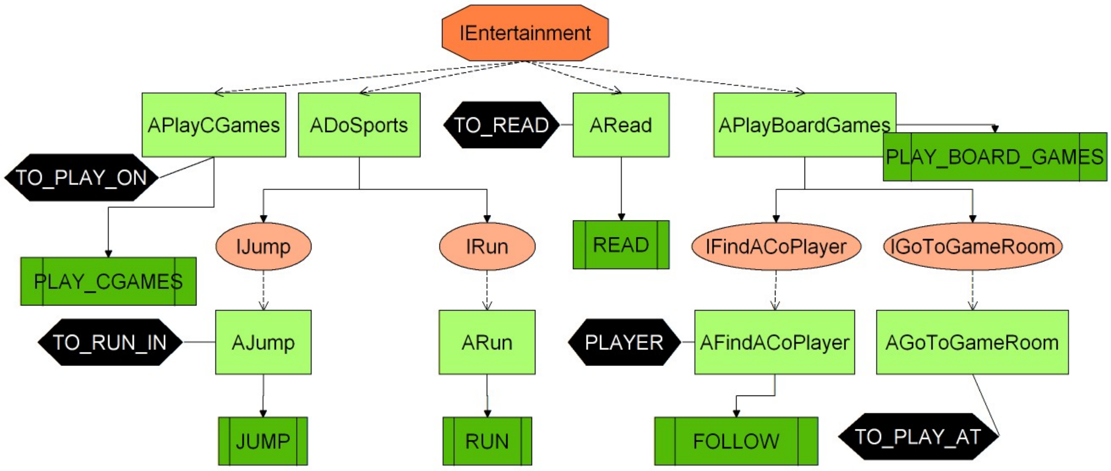


Fig. 5: Example of AND-OR tree for entertainment. The agent can either play computer games, do sports, read or play board games. For instance, if he chooses reading, he has to localize something to read (e.g. a book) and then he can read.

The Short Term Memory

The short term memory serves as a storage of important information necessary for the execution of agent's actions. It has three parts: the *perception field* which contains all resources agent is perceiving at the moment, the *process area* which holds the tasks agent is considering at the moment and the *memory area* which serves for the virtual perception.

The Perception Field.

The perception field is an entity responsible for keeping up to date the set of perceived items. It was shown by [15] that human can hold 7±2 words in the short term memory. We have used this result to limit the number of perceived items at the given moment. We have implemented a mechanism which creates a perception filter that ensures agent is perceiving only a limited number of items at once. The probability that agent become aware of a resource depends on its attractiveness as well as on how much he needs it to accomplish the current goal. For instance, if he searches for a TV, the apple would have only its basic attractiveness but as soon as he starts searching for something to eat, the attractiveness of apple arises.

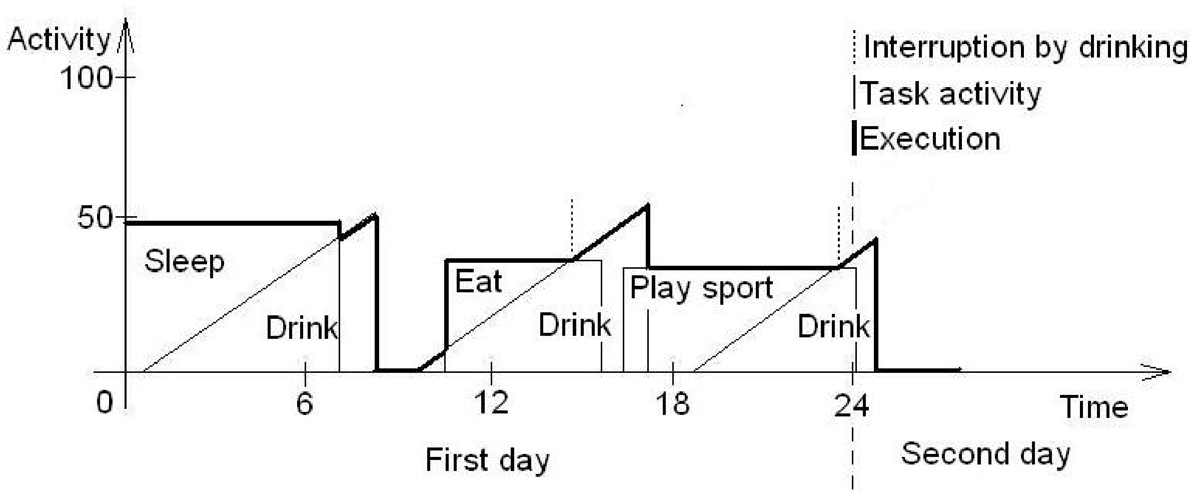


Fig. 6. Example of the competition of activities of various root desires. The sleep ends and is followed by drinking. The activity of drinking drops as soon as agent drinks something. The lunch is disturbed by the drinking. Then agent chooses the most active activity – play sports – which is again interrupted by the more active drinking desire.

The Process Area.
The process area contains all tasks agent is considering at the given moment. Stored tasks can be either in pre-active state (they are about to be executed), active (in execution) or postponed (either interrupted by another more important task, either postponed by their subtasks in execution). The activity of each task is derived from the activity of the root desire (fig. 6) of the tree it belongs to. New tasks are added into the process area when some new goal become active or when a new subtask is chosen. Tasks are removed either (1) after they failed or succeeded – which leads to the activation of its parental goals – either (2) they can fadeout if the agent is not choosing them for a long time.

The memory area.
The memory area serves for *virtual sensing*. It contains results of queries imposed to the memory during the search for resources. Thus it supplies agent with the illusion that he perceives a resource which is in fact located somewhere else in the environment therefore he can go fetch it.

4.1.2 Action Selection Algorithm

The action selection algorithm operates over the set of desires (each represented by a tree) and the process area which contains tasks for consideration. Every iteration of the action selection should end with the execution of one atomic action (from the most active task). The action selection proceeds in following steps:

1. Update agent's internal state (biological needs etc.).
2. Update activities of desires according to their activation functions.
3. Perceive – look around for interesting objects.
4. Choose the most active desire (from the root and pre-active desires).
5. If it is not already in the process area, add it there and select randomly a task by which it will be satisfied and add it to pre-active tasks.
6. Choose the most active pre-active task from the process area and execute it.
7. Check if it did not lead to its success or failure.
   1. If it did → rollback.

As we are choosing new task/goal, we leave its parental task/goal postponed, hence it will not be considered for execution. The rollback can lead to failure or success of the whole tree. When a task fails, it means that parental intention – goal has to choose another task to try to satisfy itself, if there is not any left, the goal fails and the failure can propagate up to the root. Similar mechanism works for the success of a task/goal. For more details see the documentation.

4.1.3 Scheduling

Agent is not executing the same plan for a day all over again. He is enhanced with a simple *scheduler* which processes agent's *life schedule*. The schedule is specifying what actions he would like to do, which day of week he would like to do them and the probability that he would actually schedule them in the particular day plan. The scheduling proceeds in the following fashion:

1. **Schedule basic activities** – eating, sleeping, hygiene. These goals are creating the frame for agent's day. The day start is counted based on agent's alarm clock. As we do not eat everyday on the same time each activity is dispersed a little around its usual start. In general, agent wakes up between 7 and 9 a.m., takes a breakfast. He takes a lunch around the noon and takes a dinner around 7 p.m. He goes to bed at midnight.
2. **Schedule regular activities** – the frame of a day from the step 1 provides us with three free intervals between the basic activities. Agent's life schedule contains entries which specify the activity, part of a day, day of a week and the probability, that agent will do it. We schedule agent's regular activities for a day according to those entries.
3. **Fill up the free time** – last step serves to fill the gaps in planning by either entertainment, clean up, study, etc. A *gap* stands for at least one hour of free time. For example, agent will go to school after the lunch. He has only one lecture. That leaves him with three hours of free time before the dinner, thus he will play computer games.

There are few reasons why we plan agent's days this way. Firstly, we want to have an agent that has a defined lifestyle – we can define a notion of ordinary day for him. The regular lifestyle should help him with the formation of time concepts. Secondly, we want to account for a variability as well thus the actions do not start every day at the same time. Thirdly, some actions are executed just occasionally creating significant differences between days. The example of the resulting day simulation is depicted on the fig. 7.

Another aspect of scheduling is the scheduling of desires for biological needs. We drew inspiration from introspection and decided to implement a model for biological needs. Hence agent has internal biologic variables for thirst, hunger, etc. The value of those variables determines the activation of corresponding desires. When such a desire becomes more active than other desires in consideration agent starts executing corresponding plan to satisfy the biological need. We perform the evaluation of the status of biological needs approximately every 20-25 minutes to prevent over-exceeding disturbance created by dynamically infused intentions. Most importantly this model helps with the creation of agent's context for the episodic memory and it allows us to omit the planning of goals satisfying biological needs as they can be planned dynamically.

```
Mo, 00:22, day 0. SLEEP started.
Mo, 07:22, day 0. SLEEP succeeded.
Mo, 07:46, day 0. TOILET started.
Mo, 07:50, day 0. TOILET succeeded.
Mo, 08:19, day 0. SHOP started.
Mo, 08:34, day 0. SHOP succeeded.
Mo, 08:37, day 0. COOK started.
Mo, 08:56, day 0. COOK interrupted.
Mo, 08:57, day 0. DRINK started.
Mo, 09:01, day 0. DRINK succeeded.
Mo, 09:01, day 0. COOK started.
Mo, 09:13, day 0. COOK succeeded.
Mo, 09:13, day 0. EAT started.
Mo, 09:24, day 0. EAT succeeded.
Mo, 09:37, day 0. WORK started.
Mo, 12:04, day 0. WORK interrupted.
Mo, 14:13, day 0. EAT started.
Mo, 14:24, day 0. EAT succeeded.
Mo, 14:32, day 0. TOILET started.
Mo, 14:36, day 0. TOILET succeeded.
Mo, 14:36, day 0. WORK started.
Mo, 16:30, day 0. WORK interrupted.
Mo, 16:54, day 0. DRINK started.
Mo, 16:59, day 0. DRINK succeeded.
Mo, 17:09, day 0. SHOP started.
Mo, 17:24, day 0. SHOP succeeded.
Mo, 17:29, day 0. COOK started.
Mo, 18:00, day 0. COOK succeeded.
Mo, 18:00, day 0. EAT started.
Mo, 18:10, day 0. EAT succeeded.
Mo, 18:40, day 0. SELL_GROCERIES started.
Mo, 18:51, day 0. SELL_GROCERIES succeeded.
Mo, 19:31, day 0. STARE_AT_TV started.
Mo, 20:21, day 0. STARE_AT_TV interrupted.
Mo, 20:33, day 0. TOILET started.
Mo, 20:36, day 0. TOILET succeeded.
Mo, 20:36, day 0. STARE_AT_TV started.
Mo, 21:43, day 0. STARE_AT_TV succeeded.
Mo, 22:07, day 0. SHOWER started.
Mo, 22:17, day 0. SHOWER succeeded.
```

Fig. 7. The log from simulation of one day in agent's life. It shows only executed atomic actions but it gives a notion about the way agent is simulated. Approximately on third of the actions is triggered by dynamically added intentions (drinking, shower, urination). The scheduler is responsible for the rest.

4.2 Item Memory

Agent is performing a lots of various tasks and most of those demand one or more resources (prerequisites) before they can be performed. A resource can be in general anything and agent is constantly facing the localization problem when obtaining resources for his actions. Therefore agent needs a representation of space or some memory for resources. This work is focused on the episodic memory thus we will model the memory for items in scarce and simple fashion.

The item memory is storing every item, place and agent he has met. It stores an additional information with each record about the last time he saw it, how many times he found it and how many times he failed to find it on the location. Those information are then used in the simple formula:

$$credibility = (2 * found - missed) * 10 + seen \quad (1),$$

for determining the *credibility* of the record. For instance, if the agent saw the item on the particular place twice, then he did not find it there three times and he passed by it 10 times, the resulting credibility is (2*2 – 3)*10 + 10 = 20%.

The agent queries item memory for items of desired affordance, sort the result by credibility in the descending order and visit obtained places. The counters assigned to each item could grow beyond all measures thus we apply a normalization and decay on those records. Each midnight *seen* value is decrease by 10, *found* and *missed* are decreased by one. If the credibility decreases bellow 5% or if the last update of the record is older than a day, the entry is discarded.

4.3 Time Representation

People do not perceive time as a linear homogeneous continuum. They form various time patterns and concepts to be able to date past memories. The exact algorithm which is responsible for their creation is unknown. We will try to develop a mechanism that would consequently enable agent to learn time patterns. We believe that if we enhance an agent with a better time perception we will also enable him to provide more believable responses to questions concerning his past.

Firstly, we would like to account for a problem of exactness of the previous time-pointer based model of episodic memory. The agent needs a way how to perceive and then describe the time more vaguely. We propose a model that represents time concepts as nodes of a neural network. The concept nodes start as anonymous nodes connected with nodes for agent internal state – context nodes – and nodes for the biorhythms or time flow – cartesian nodes. Weights of interconnecting links are set to random values and the value is then altered via Hebbian learning mechanism. Consequently, after few "days" of simulation, time concepts should emerge.

Secondly, we would like to solve the problem that the previous model was not error prone. It reflected exactly what was stored and what was stored was exactly what happened. We would like to account for that on different levels – time scales. For instance, temporal concepts from preceding paragraph will blur time information on the scale of minutes and hours. We propose a similar mechanism for days which would then make agent remember older information with a lower precision.

The section is organized as follows. We will represent the memory for time concept consolidation with all its components and features, then we will continue with a section about the connectionist memory.

4.3.1 Neural Network for Parts of a Day

The time perception model constitutes of a neural network with two layers (fig. 8). First layer consists of *context nodes* (CX) and *cartesian nodes* (CA). Second layer is formed by *concept nodes* (CP). Hence the network is a tuple $(CA, CX, CP, M_{CX,\ CP}, M_{CA,\ CP}, f)$ where $M_{CX,\ CP}$ and $M_{CA,\ CP}$ are matrices of weights between *CX*, *CP* and *CA*, *CP* respectively. *f* is the sigmoid function which puts the activation to the interval (0.0, 1.0). CA and CX nodes serve as an input for CP nodes. CX mirror agent's internal state as well as its external context. CA are activated along with the flow of time.

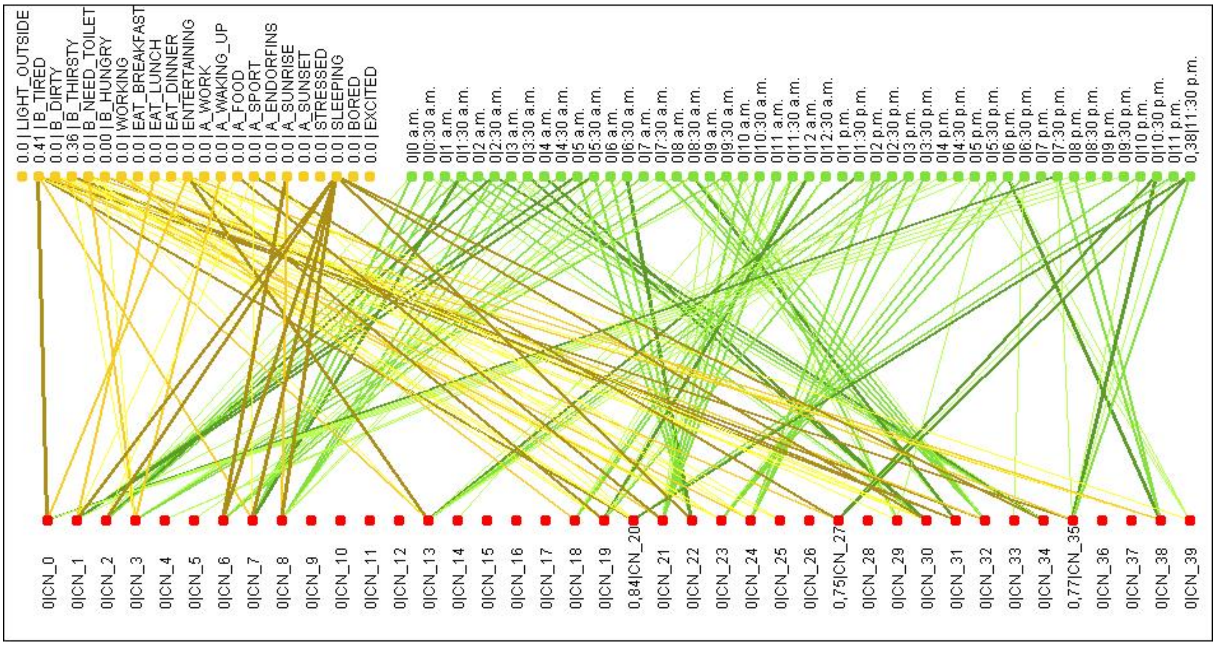


Fig. 8. Learned neural network for parts of a day. There are in the top left (yellow circles) – context nodes, in the top right (green circles) – cartesian nodes and on the bottom (red circles) – concept nodes.

4.3.2 Context Nodes

Context nodes represent one side of input into *concept nodes*. They mirror the context in which the agent is at the given moment. There are about 20 of them in the current model (see Appendix B for complete list). Their activation range is between zero and one. The mechanism which is counting the magnitude of activation depends on the type of a context node. There are three basic types: biological, external and internal. Biological context nodes represent state of agent's biological needs – e. g. hunger, weariness, thirstiness. External context nodes stand for the state of environment – light outside, snow on the ground, sunset, sunrise etc. Internal nodes reflect what is he sensing or doing or what he was doing recently.

Biological Needs.

Agent contains a very simple model of biological needs which takes care of the simulation of natural drives like the urge to go to the bathroom from time to time, take a shower, drink something or take a rest.

The model keeps up to the date variables determining how thirsty, hungry, dirty, tired agent is. The update is called every round of the logic and rises those values. The activity of the corresponding context node is counted simply from the bias for the biological variable and its value. The activation ranges between 0.0 and 1.0. When a biological need become urgent – high value – it can be satisfied by the corresponding action. This action is dynamically added to agent's plan with activation proportional to the activity of the context node. The agent considers addition of dynamic actions approximately every twenty to thirty minutes. When the action succeeds the biological variable is restarted which renders the corresponding context node inactive – e.g. hunger is deactivated by successfully accomplished eating.

Moreover, as our weariness, hunger etc. don't grow linearly and sports and other activities can rise them faster, we have introduced a way how to change the speed of growth of the biological variables during the simulation.

Biological context nodes and the whole concept of biological needs improves agent's believability and helps to make his daily plan more real.

External Context.

External context nodes represent the state of environment that is not influenced by the agent. In the case of our prototype those are supplying him with information about night and day and sunrise and sunset. Of course in some more elaborated scenario those can represent anything relevant like a snow on the ground, outside temperature or some abstraction of the surrounding environment.

Internal context.

Internal context nodes express agent's internal state. That is a very vast domain. Our prototype is working with contexts like excitement or work (meaning an activity we have to do). The activation of internal context nodes is triggered by a certain atomic action and the magnitude of activation is hard-coded. We can connect at this point the emotional model.

4.3.3 Cartesian Nodes

The *cartesian nodes* represent parts of the day-time continuum. We can look at them as on the time nodes which fire depending on how far is the current time from their peak time. The cartesian node is defined by a triple *(peak, height, width)*. The activation of each node is defined by a Gaussian curve with the peak aligned with the position of the node on the time axis (*peak* of the node). For instance, noon has its peak at 12 o'clock and its width is about one hour, the height is 1.0. We can view the layer as watches with a 24-hour watch face where nodes are situate,d for instance, at the place of hours (or at every 5 minutes) (see fig. 9). The activation travels around those nodes like the big hand on watches. The activation of node is counted based on its Gaussian function:

$$activation_{jt} = height(j) \cdot e^{\frac{(t - peak(j))^2}{width(j)}} \quad (4.1)$$

counts the activation of the cartesian node j in the time t. The activation is highest when the "hand" points into the *peak* of the node.

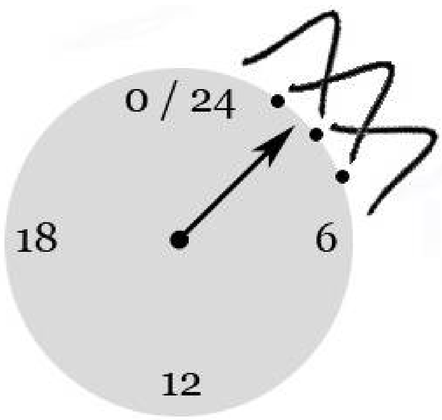


Fig. 9. Activation (the Gaussian curves) of the cartesian nodes (black dots on the perimeter).

Afterwards we can think about introducing a hierarchy of those biorhythms by extending them with nodes for longer periods of time like few hours together, days of a week etc. This extension could then create differently grained concept nodes which could represent entities like "Monday morning" or "ordinary day afternoons". We have kept in mind our main objective thus postponing this course of research.

4.3.4 Concept Nodes

Each *concept node* is representing some period of time and the context agent is in during that period. Therefore it is aggregating together a few context nodes and a few cartesian nodes.

Concept nodes are connected with all nodes from previous layers. Those links are weighted and they are represented by two matrices, one for context-concept links, another for cartesian-concept links. Weights are randomly initialized from the interval (0, 0.2). The activity of all concept nodes is calculated by following steps:

1. $R_{CX} = V_{CX} \cdot M_{CX,CP}$, where R_{CX} is a vector with partial result for context nodes, $M_{CX,CP}$ context-concept weight matrix, V_{CX} active context nodes.
2. $R_{CA} = V_{CA} \cdot M_{CA,CP}$, where R_{CA} is the vector with partial result for cartesian nodes, $M_{CA,CP}$ cartesian-concept weight matrix, V_{CA} the vector of active cartesian nodes.
3. $R = R_{CX} + R_{CA}$, where R contains activations for concept nodes in the given step.

The most active node is then linked to the action which is performed at the moment and to the item/place in use but more about that later. Now we will focus on the learning mechanism of the network. We use the Hebb's learning rule (adopted from Dayan, Abbott [26]):

$$\tau_w \frac{\delta w}{\delta t} = v u \quad (4.2)$$

where τ_w is a time constant at which the weights are updated (in our case every step of the logic), v is a vector of weights and $\boldsymbol{u}$ is a vector of changes. The rule says that the simultaneous pre- and post-synaptic firing increases synaptic strength. Hence we strengthen links between co-activated nodes.

We have modified the learning rule. The weight of a link in the next step is:

$$v_{t+1} = f(f^{-1}(v_t) + c) \quad (4.3)$$

where f is a sigmoid function. Thus the stored weight is actually the direct activation of the link (when multiplied with the incoming activity). The link is strengthened by the addition to the value from which is counted the activation for the next step.

As Abbott points out, solely strengthening would never lead to the stable state of the network. One way to stabilize Hebb's rule is to introduce some form of weight normalization which is following the notion that a post-synaptic neuron cannot support more then a limited sum of synaptic weights. Then the increase of one weight leads to a decrease of another. We have chosen the subtractive normalization with non-negative weights of links. In compliance with Dayan, Abbott [26] the formula is following:

$$\tau_w \frac{\delta w}{\delta t} = v u - \frac{v(n \cdot u) n}{N_u} \quad (4.4)$$

where n is a vector with all components equal to one and N_u is a matrix of ones. The normalization is applied on both types of inputs to the concept nodes ensuring that there is a limited weight count from either context nodes either cartesian nodes side.

We will present results obtained from learning in the chapter 6, now we will conclude this chapter with a specification of the connectionist memory.

4.4 Connectionist Memory

The *connectionist memory* is the key module of our episodic memory system because it is there where the episodes are stored. It stores various entities which are linked together by weighted links. The model is designed for easy extensions thus we can interconnect virtually anything with other nodes. At present, we are linking together time concepts, days, AND-OR trees and items. We can link with those as well emotions, abstractions of agent's internal state or any other representation of contextual information we want to store along with the episode.

There are several questions which must be answered before we can implement such a system:

1. How exactly do we link entities?
2. How do we select the right set of entities to link together?
3. In what structure should we store links?
4. How can we count or estimate the interestingness of an episode or its part?
5. How can we mimic forgetting?
6. Can we really blend episodes?

We will try to give answers to those questions in the following text.

Linking.

First observation we made was that we cannot work with the links in the episodic memory as with the links of the time network. The weights of the time network are updated every step of the logic. If we do the same with memory links there will be very strong links for long lasting activities which would eliminate any short yet interesting event. Thus we have discretized actions and add or strengthen links only when there is a change in the situation. Links are strengthen in compliance with Hebb's learning (equation 4.2).

That leads us to the question 2. We need to choose right entities to define the episode. Fortunately we can use some features of the DMS. When agent executes an atomic action it means that he has chosen a path through the tree of the most active goal and found all resources necessary for the execution. All those information are stored directly in the tree. Hence we know what particular item he is using, what is the root goal etc. Therefore we collect all intentions (goals) and tasks from the bottom of the tree to its root and add other entities we want to link with the action like days, items and the most active concept node at the moment. All items in the resulting set are interconnected by links – each item with all other items (fig. 10).

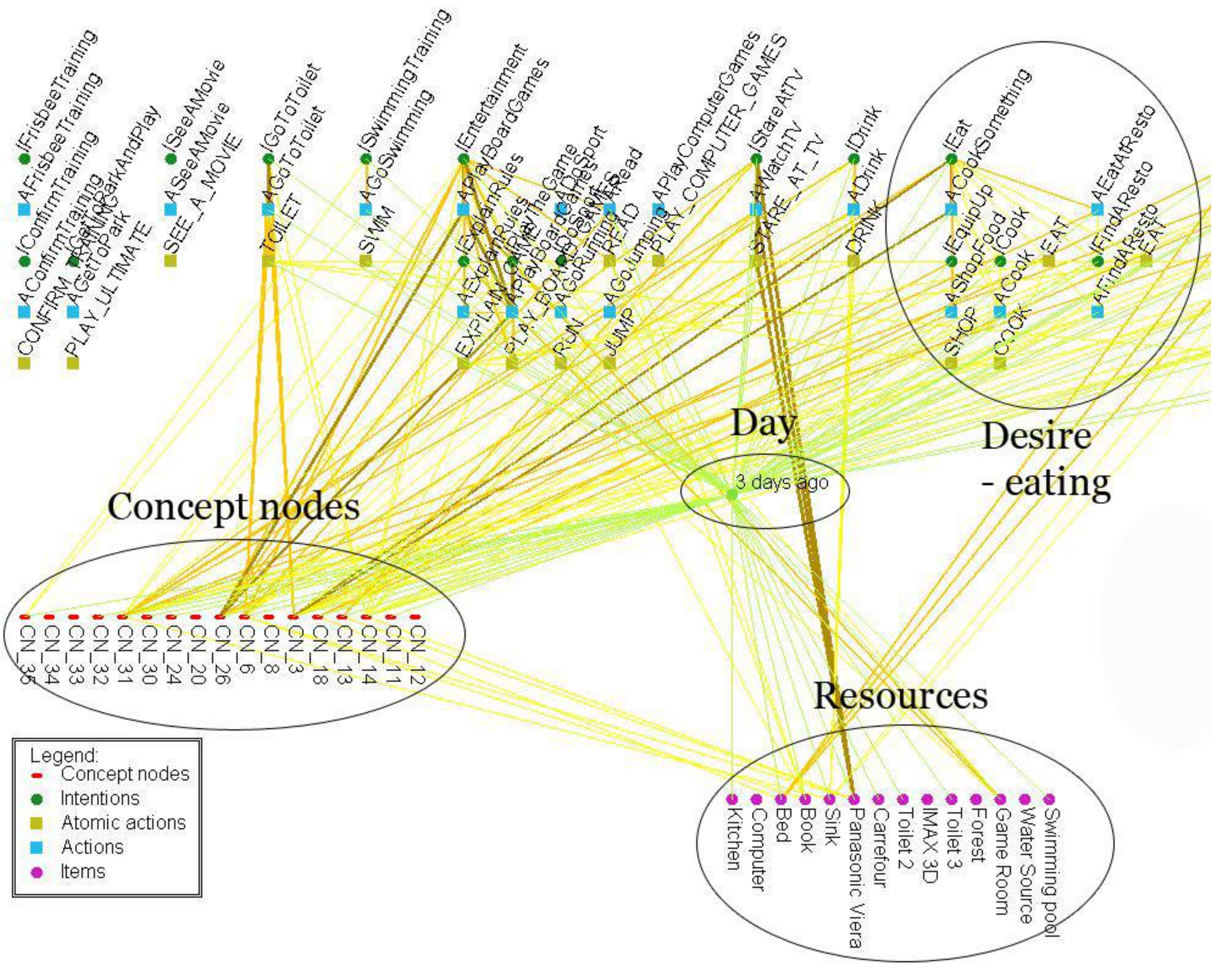


Fig. 10. The visualization of the part of the connections between the nodes. Yellow lines are links between items, tasks, atomic actions, goals, and concept nodes. Green lines represent connections with a day node (the node for "three days ago" at the moment).

The set of active nodes evolves even within the same episode. For instance, the most active concept node changes, agent starts execution of another subgoal etc. If we link all nodes together again whenever something changes the weights will depend on undesired factors like the number of children of the node. Hence every time we want to add a new link, we check first if the same link was added the same day and do not add it if so. We also forbid multiple strengthening during a day as it could give an unwanted attention to repetitive actions. Hence the strengthening of the particular link occurs at most once a day.

The algorithm for link-strengthening is:

1. Collect all active linkable entities – the day, the most active concept node, items in use, actions and intentions from the tree which are on the path from the atomic action in execution to the root.
2. For each possible link between entities in the set:
3. Verify if the link does not exist.
4. Strengthen the link between the nodes in the set.

Storage.

We have some options for the storage of links to consider. The first apparent idea would be the matrix. There are many nodes but the resulting matrix of edges is very sparse. Hence the matrix is not appropriate fit for our requirements. We have chosen a straight-forward solution which stores links to other nodes directly in the node. There are practical reasons for this decision. It enables us to search swiftly through the structure and we can easily define additional rules for different types of nodes and we can perform various queries.

Days.

Another issue to address is the problem of storing many days. Up to now, we have a structure which can keep information about yesterday. When we start extending it with nodes for days we have account for the fact that days are different thus it might be necessary to extend further the structure. There are three approaches we can consider.

Single Structure.

The easiest solution would be to store links for time concepts in the same structure no matter the day. When we add/strengthen a link we just check whether the same link was not stored the very same day. The link can be indeed strengthen every day thus we have to solve the problem with weights that are rising beyond any measures. We use the subtractive normalization from equation 4.4 to normalize the total amount of weights from one node to a certain bias.

The structure does not demand much of resources. There is an upper limit for the number of links between the concept nodes, events and resources and there is a set of links for each day. But we expect the structure to lose its capability of answering questions about past events. If we query it with a part of a day and a remoter day, it will only return the usual activity of that part of a day using a day cue as a clue to decide between multiple activities which belong to the same part of a day. We will test this hypothesis in the chapter 7. If verified it renders this structure useless. But since the structure can retain the image of the ordinary day, we can keep it for future use.

Multiple Structures.

We have reasoned that one structure for all days is not enough hence we have developed another model featuring multiple structures – for each day one separate structure which holds links between CPs, events and resources. That way we store all information about the day and then, using a conveyer belt metaphor from the chronological organization theory for time (chapter 2.2), we put it all as a bag into the past and create a new clean structure for the following day.

This approach solves one problem and rises another – the forgetting. We cannot keep all the information at once in the memory. Moreover, we do not have to as people do not remember every detail from the past either. Thus we have introduced a simple forgetting mechanism which is based on the multiplication of weights by a coefficient $\lambda < 1.0$ until the weight reaches certain bias and the link is discarded. If we combine this approach with a slower decay of links for days we can get a memory which remembers relatively accurately last two weeks (depends on the value of λ) and then it remembers only important events which happened on single days.

Combined Algorithm.

We have denoted at the end of the description of the single structure that we can use its properties later. We are working on the prototype of the model for episodic memory thus the issue of used space is not our main concern but it will be in the long-term perspective. We propose to combine the two algorithms and exploit their amenities while eliminating their downsides. The combined algorithm stores both structures which provides us with at least two possible ways for lowering episodic memory space demands. First, every time we want to add a link to a day structure, we can check first if it is not present in the ordinary day (single structure) and then add only a link from the day to the event. Hence we are keeping in the day structure only interesting events and the single structure stores regular events. Second, we can forget details faster while relying on the ability to reconstruct at least some story based on the ordinary day. We are going to provide detailed comparison of the first two algorithms in the chapter 7.

Estimating Interestingness of an Action.

Current problem of the model is that links in the multi structure model are weighted but they weight all the same as they are strengthened only once a day. The mechanism that evaluates interestingness of an action is missing.

We can think of a few solutions to this problem. One of the elegant solutions would be to use the output of some model of agent's internal state which could include emotions, for instance, and would produce an excitation coefficient. The weight can be then multiplied by the coefficient. Another way how to do it would be to annotate actions and intentions with an interestingness property. But that would be hard-coded ergo more demanding for the designer. Then there can be an external mechanism which would count the interestingness according to the output of the memory. Or we can try to deduce the interestingness from the differences in the context of the new episode.

For the present we have implemented a mechanism for the interestingness of actions on the day scale. We presume that if an action does not happen every day, it is more interesting than other actions. Thus when we connect a node with a day we check first if there are same links for previous days. The strength of the new link is determined by the algorithm:

```
newWeight = 1.0;
for (int i = 1; i < 5; i++) {
       if (linkExists(link, days.get(i)) {
              newWeight *= i / 5;
              found = true;
       }
}
if (!found) {
       newWeight *= 2.5;
}
```

where days.get(i) means the day for *i* days ago. Thus the new weight of a link is lower if the activity repeats every day and higher if it is rare.

Multi-days.

We are left with one question unanswered. Question 6.: "Can we blend episodes?" We should first define what is meant by blending. Two episodes blend into one if they loose the time information on some scales. For example, if an agent works on Monday afternoon and then works on Tuesday afternoon, he can remember that he worked in the afternoon on both days creating a new episode for the two-day time scale. The link from this new episode will be stronger than any of the two links to the separate days. Consequently one-day traces will be forgotten faster than the two-day one leaving agent with a memory of himself working on some day from the two, maybe on both. The notion of two-days can be easily extended to three-days, weeks, etc. The advantage from the implementation point of view is that we do not need to define new mechanisms for multi-days and use the current one for days. Multi-days require a bit more elaborated algorithm for querying though.

There are other possible improvements and enhancements to the episodic memory module that we can think off. We will list them here for the future reference:

- We can assign a coefficient to various types of links (depending, for instance, on the types of nodes they are linking together). The coefficient will be then used during a learning/link-strengthening resulting in a way how to manage the lifespan of different types of links. For instance, links with concept nodes can deteriorate faster then links with days.
- Tweak the parameters. There are many parameters that could be tweaked – the link-decay ration, the limit of weights for a node in the single structure, decay of days, the shape of the function for decay etc.

4.5 Conclusion

We have designed a connectionist model for episodic memory. The model achieves presented objectives:

1. It enhances agent with the notion of social and biological time concepts like afternoon and evening enabling him to answer questions which contain such cues and reply to questions using these patterns.
2. It does not store the information in the exact manner thanks to the time concepts.
3. It can blend similar episodes.

Nevertheless, the design hides a trade-off. The memory system is more space demanding that would be a simple log of agent's actions but the demand is closely linked with the settings of links' decay. The forgetting mechanism prunes records and we can get lower requirements after several days.

5 Methods

The following chapter is dedicated to the description of the environment for the agent, used integrated development environment and some specifics of the simulation. The preceding prototype of Peskova showed that the DMS works properly in the discrete simplified 2D world where agent can reach for items instantly, perceive them when he enters the room and move without dealing with path-finding problems. To take another step forward we have decided to implement the prototype in a complex continuous 3D environment which would be closer to the usual environments for IVAs. The successful implementation in such an environment will show that the DMS and memory work in a real-time applications as well as in the real worlds which are compliant with laws of physics. Hence we have searched for a platform that would allow for fast prototyping of the virtual human in some world-like 3D environment.

We have chosen the platform Pogamut 2 [27]. Pogamut 2 is the only freeware integrated development environment for virtual agents in the complex environment on the market, see Burkert et al. [28] for detailed discussion. Moreover, it provides developer with some effective tools which can facilitate the development.

5.1 Unreal Tournament 2004

As we are using Pogamut 2 for the development of the prototype implementation, we have to use Unreal Tournament 2004 (UT2004 [29]) as the environment for the agent. UT2004 is a first-person shooter 3D game. It provides us with a complex environment with a continuous time flow. The agent is making decisions in steps on some frequency of course but he has a limited time to decide what to do next (to do one iteration of the decision making system).

Fig. 1: Eating agent in the UT2004.

The game is supplied with an editor which permits easy modification of existing maps (locations) and creation of one's own. Thus we can save time using the abundant on-line source of maps created by the community of players, picking up the most convenient one and modify it.

5.2 Pogamut 2

Pogamut 2 is the IDE for development of IVAs. It has three main components (fig. 11): Gamebots2004, Client (abstraction of agent), and IDE. The Gamebots2004 are managing export of information from UT2004 and reception and execution of commands. The Client is handling the connection to the server, parsing messages and providing libraries of atomic actions and basic sensors. Netbeans™ [30] plug-in supplies us with the IDE. Not only it is an IDE for Java but it contains as well some extensions for IVAs development like the server controller, log viewers or introspection. For more extensive description see [31].

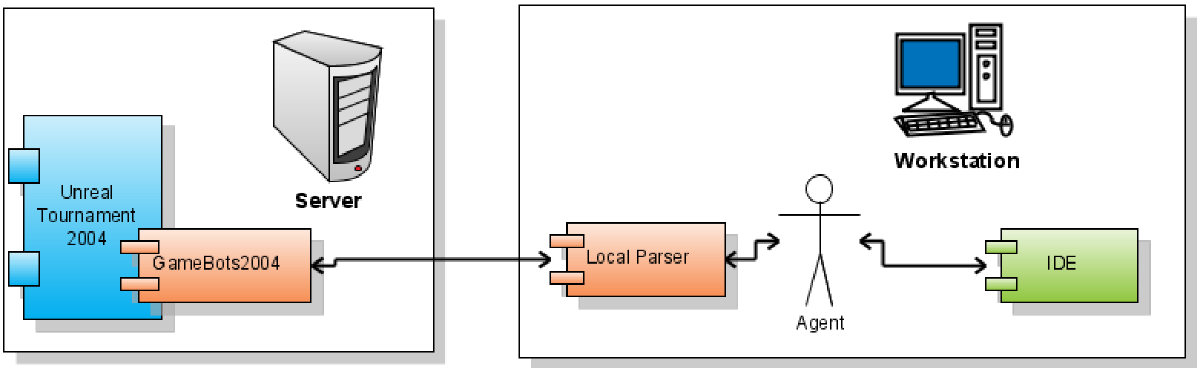


Fig. 11. Overview of the architecture of the Pogamut 2. UT2004 and Gamebots2004 can run on a separate machine (server). The information about the environment as well as commands sent to the environment are processed by a parser which translates them to/from Java objects. The agent is managed by the IDE.

The only significant thing that is lacking in Pogamut 2 is a better support for non-violent scenarios and storytelling. So far it is not easy to add new items and use different graphical models for agents. Those issues will be solved in the new version along with a connection to other environments. Furthermore it is not as important as we are working on the prototype hence Pogamut 2 is sufficient for us.

5.3 Lifestyle

Our main goal is to implement and test a model of episodic memory for *human-like scenarios*, which implies that we cannot test the model on some labyrinth-style experiment but rather on an agent who is "living" in the ordinary world. Thus we need to define a *lifestyle* or rather several lifestyles for a comparison and then examine how is the model behaving.

The lifestyle has to be compliant with following constraints:

- Relatively monotonous life with regularities.
- No sleeping during the day.
- Three reasonably distributed meals per day (exact start times depend on the time agent wakes up)
- Breakfast starts between 7 a.m. and 9 a.m.
- Lunch starts between 12 a.m. and 2 p.m.
- Dinner starts between 6 p.m. and 8 p.m.
- School/work activities should be scheduled in the morning. Sports, culture and other entertainment should be scheduled in the afternoon and in the evening.
- About 8 hours long sleep.
- Hygiene twice a day in the morning and in the evening.

Our main objective there was to give the agent some patterns he can recognize and use for the formation of concept nodes. The agent without any day frame would live only according to his biological needs taking a nap when tired, eating when hungry and killing the time by random activities in the meantime.

5.4 Simulation – Scenario Description

The simulation was conducted on a laptop. We modified the map DM-Crash – one of the original maps supplied in the UT2004 distribution (fig. 12). We used three different lifestyles which varied in the timetables of activities (see tab. 1 for illustration), the alarm clock and types of activities. The complete list of items, their affordances as well as plans can be found in the documentation.

hours	Mo	Tu	We	Th	Fr	Sa	Su
7	*Breakfast*	*Breakfast*	*Breakfast*	*Breakfast*	*Breakfast*	*Breakfast*	*Breakfast*
9	Work		Work		Work		
11	Work		Work		Work		
13	*Lunch*	*Lunch*	*Lunch*	*Lunch*	*Lunch*	*Lunch*	*Lunch*
15	Work	Work	Work	Work	Work	Work	Work
17	Work	Work	Work	Work	Work	Work	Work
19	*Dinner*	*Dinner*	*Dinner*	*Dinner*	*Dinner*	*Dinner*	*Dinner*
21		Theater(0,7)	Movie(0,2)		Theater(0,3)		Movie(0,8)
23		Theater(0,7)	Movie(0,2)		Theater(0,3)		Movie(0,8)

Tab. 1. An example of a week plan of an agent. The number in brackets represents the probability that he will perform the task.

Another aspect of simulation is the frequency at which agent is making decisions and the speed/time of simulation. Firstly, we are limited on the side of the environment. UT2004 or more specifically Gamebots2004 are providing us with information updates every 250 ms thus it is useless to use a frequency higher than 5 Hz (5 decisions per second). Secondly, we have to give an agent enough time to move between places and items as it is not an instant action any more. Thirdly, we have to consider a plausible conversion between the game time and the real time. As suggest [32] human beings are making decisions approximately every 3s. Taking into an account the 5 Hz limit it leaves us with a simulation time for a day equal to 96 minutes. That would mean that a simulation of a month (a usual length of an experiment) would take two days. As there were plenty variables to tweak and many different research paths and options to try out we have decided to increase the *time per decision* up to 15 and 30 seconds. Nevertheless, the program is able to run with all different settings of time per decision.

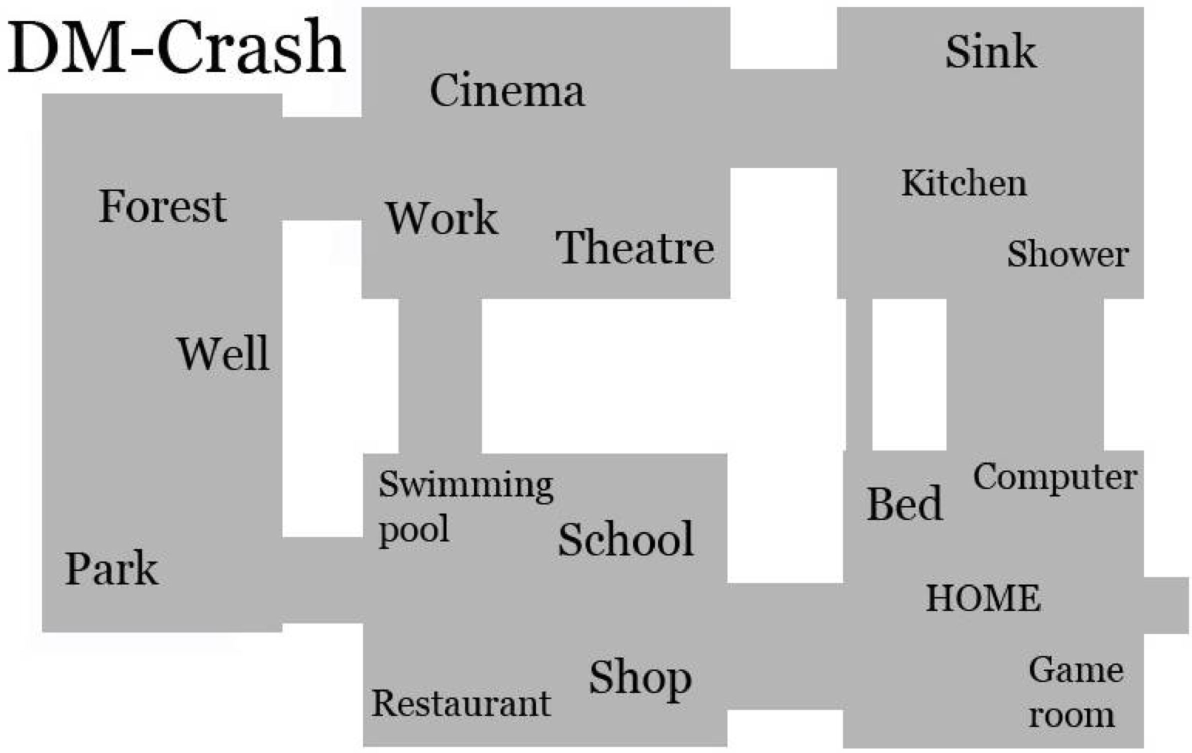


Fig. 12. Disposition of places in the map used for simulation.

Last but not least to mention is the lifespan of agent and biological constraints. The lifespan of agent is in weeks. We have successfully run the simulation for 70 agent's days but the usual length of an experiment is usually 20 agent's days. The agent is not exactly plausible from the biological point of view. He gets tired, he feels thirst and hunger, but he will never die because of it. It will only cause very high activation of corresponding context nodes.

6 Methodology for Model Evaluation

We have designed and implemented the model described in chapter 4. Consequently we want to validate the model using a set of experiments which test its capabilities. This chapter is dedicated to the description of the mechanism which allows for timezone traveling and better concept node quality verification, the questionnaire which helped with creation of the set of questions agents are usually asked and the outline of the querying mechanism for the episodic memory.

6.1 Greenwich Mean Time (GMT) – Timezone Shifts

We were constantly facing the problem of verification of the quality of time concepts during the development cycle. The issue was in the metrics or method which can determine that concept nodes are learned properly. Thus we propose the timezone experiment. The agent is traveling between timezones. He uses the time network to synchronize his internal clocks with the real time. The premise was that if concept nodes actually represent some time period, agent should be able to adapt to the new timezone.

The learning of timezone shifts proceeds in the following schema. The agent starts simulation in the initial timezone (GMT 0). He lives according to a certain lifestyle. First, the time concepts are learned via time network. Then the time concepts are used to synchronize internal clocks with the real time. When the agent moves to another timezone while living to the same lifestyle his internal clocks should resynchronize with the new real time.

Hence the agent has its own *internal clock* which indicates him the time he thinks it is at every moment or more specifically the timezone he thinks he is in. Using the analogy of the ball-bearing, agent's internal (thought) time is the internal ring (fig. 13) while the real time is the external ring. Those rings are turning around together making one round a day. The relative positions of the two can change – mainly by the change of the time zone (fig. 13 – middle). Now how do the internal ring discovers that the external has shifted? It does not, but it is slowly synchronized with the real time until they are aligned again (fig. 13 – left).

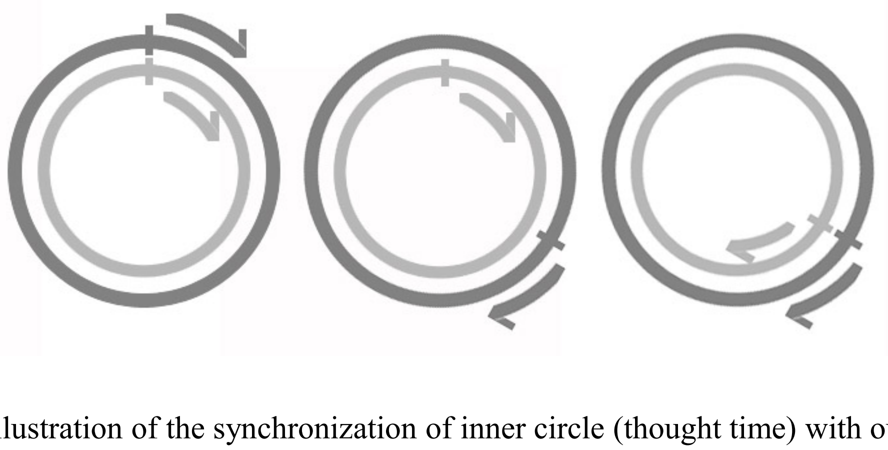


Fig. 13. Illustration of the synchronization of inner circle (thought time) with outer circle (real time) which follows the change to the timezone +9 GMT.

Synchronization.

First the internal time has to go through the learning phase. In this phase agent is "looking always at his watches" and is ignoring the internal time. Learning algorithm is using back-propagation of the activity from concept nodes. The algorithm is following:

1. Determine the most active concept node.
2. Infuse activation 1.0 to it in the direction of cartesian nodes and collect the resulting vector of activations for cartesian nodes.
3. Align the vector with the GMT wheel using the real GMT shift and the real time.
4. Add corresponding values to the internal time nodes.
5. Normalize activations of internal time nodes to the given limit.

Hence every step of the logic, there are few concept nodes activated. The activation is back-propagated from the best concept node to cartesian nodes via weighted links. Received activity is then added to the activity of corresponding node in the *internal time ring* (fig. 14). We use the real time while determining the correspondence. The activations of internal time nodes are normalized after the addition. If the internal time nodes constantly correspond with the real time – the back-propagated activity increments on the corresponding internal time node for GMT 0 – the initial time zone.

Fig. 14. GMT wheel. The strongest internal-time node represents the magnitude of the time shift. The activation is depicted by a column over the node.

For example, it is 4 a.m. and agent is sleeping. He is in the initial timezone. The most active concept node is connected to three cartesian nodes (for instance 3 a.m., 4 a.m. and 5 a.m.). The two rings are synchronized at the moment. That means that the zero on the internal ring (means 0 GMT – no change) is aligned with 4 a.m. so we add the activations to internal time nodes -1, 0 and 1 respectively. When the learning period is over the most active node should be zero – 0 GMT.

Another example, the agent moves to a new timezone +6 GMT. It is 11 a.m. and the most active concept node is connected to 10 a.m. 11 a.m. and 13 a.m. The internal ring has the highest activation on 0 GMT. But as we are in the relearning phase we omit that and align the vector of activation of cartesian nodes according to the real time and real GMT. Thus the node for 11 a.m. is aligned with the internal time node for +6 GMT (real GMT – thought GMT). In the end we add the activation to nodes +5, +6 and +8 respectively.

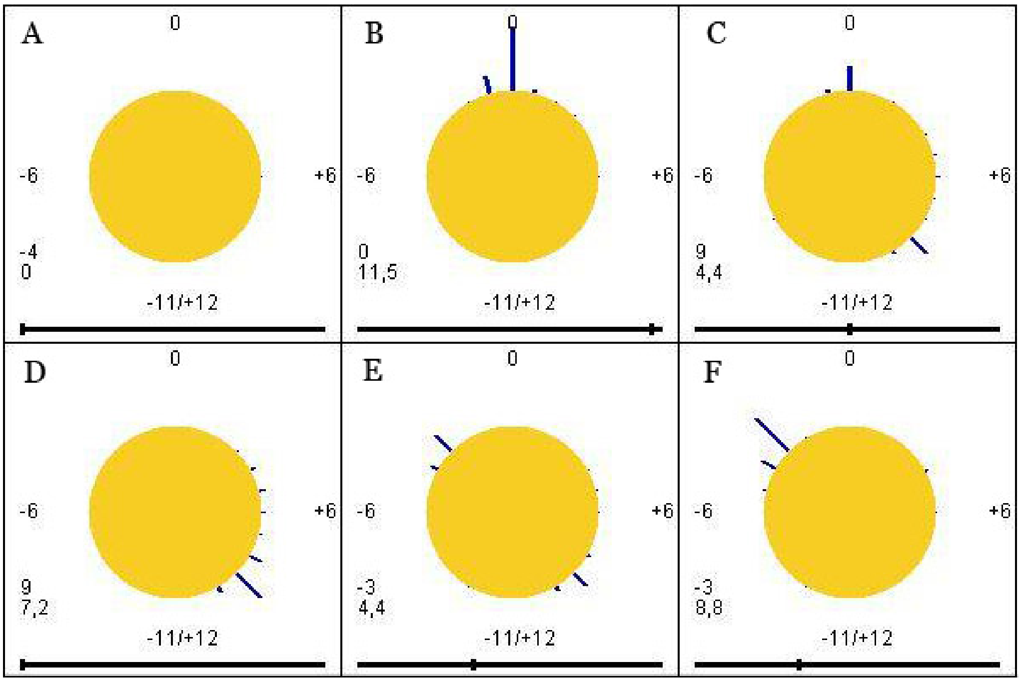


Fig. 15. Example of learning the changes of time zones. The activity on the internal time ring (A) at the start, (B) at the end of learning period, (C) during the switch to +9 GMT, (D) in the new time zone, (E) another switch to -3 GMT, (F) the end of the learning.

Time Zone Change.

The agent can start using the internal time since the synchronization between external and internal ring is firm. He uses the thought time to determine the most active concept node for the memory purposes. When the timezone change occurs, agent has to look more at his “watches” again which means he is using the real time to learn the new timezone. The real time is used to determine the activation of concept nodes for use in the time zone learning. The criterion for successfully learned time-shift is that the internal clock is synchronized with the real time for at least one day.

In the end, such a model has another interesting advantage. It makes the agent more believable as he is not sure about the time first two days in the new time zone. If asked for time estimates, he can reply using the time compliant with the former timezone. Moreover, it is fully automatic.

6.2 Questionnaire

The episodic memory can enhance agent on many different levels. It can provide him virtual sensors, repetition detection, improved reasoning and learning. Moreover, it has the autobiographical component which allows agent to answer questions concerning his past. The requirements on the autobiographical memory has not been established yet [23]. We do not know which questions are typically asked by users of IVA applications. Thus we have carried out a questionnaire to determine the set of these questions to constraint requirements on the memory.

The questionnaire was carried out in two rounds. First, we asked 7 participants to write down a set of questions which they would ask a NPC (a non-player character) in a role-playing game (RPG). Then we have sorted out questions which were not related to the episodic memory and time. The resulting set of questions was then extended by a set of similar questions with altered specification of time (e.g. "What did you do yesterday afternoon?" → "What did you do on Friday at 13:15?").

1. How many customers did you have yesterday?
2. What did you do on Friday morning?
3. When was the last time your well was dry?
4. Could you describe me in detail the orc raid on the village?
5. How were the orcs who attacked the village armed?
6. What did you do 23. February?
7. Do you have problems with a criminality here?
8. Where are you going every Sunday after eight?
9. When exactly did you wake up last Friday?
10. Have you met someone interesting in the pub yesterday?
11. How long did it take you to lunch yesterday?
12. Do you know where can I buy some healing potions?
13. They brought you new goods yesterday. How long did it take the merchant to unload the third crate?
14. When exactly do guards change in front of the castle gate?
15. Have you ever been robbed?
16. Have you seen three warriors passing by last week?
17. What did you do yesterday from 12:35 to 13:15?

Tab. 2. List of questions used in questionnaire. Questions in bold were obtained from the first round, the questions in italics are artificial, other questions aim on semantic information.

We have merged questions together mixing questions from the first round with artificial questions (tab. 2). The questionnaire was given to 30 participants (27 men, 3 women). Half of them had played a lot RPG games in the past, half of them played them a little or not at all. They were asked to rate questions on the scale from 1 to 5 where one indicated weird question while five indicated valid question one would ask an agent.

The questionnaire specification given to the participants was following.

The purpose of this questionnaire is to create a set of questions for a non-player character (NPC) of the world of RPG game. Let's imagine that the game works with a vast world which include several towns. The player can move freely around the world. Our NPC is living in one of the towns. He observes his surroundings hence it is possible to give him questions concerning his knowledge about the world, his personal data and most importantly about his personal experience.

Let's imagine that you are in the following situation: you have just finished a quest and you arrived to the town you have never been to before. You are trying to acquire a new quest. You walk around the town and chat with NPCs asking them questions. You meet our NPC in a general merchandise store. He is the merchant.

The setting concerned the acquisition of a quest which can explain high values (fig. 16, tab. 3) for questions related with criminality, orcs, guards and healing potions (4, 5, 7, 12, 14, 15). But we have also found, that most of the artificially added questions with the accurate time information were rejected (6, 9, 13, 17) as well as closely personal questions (2, 8, 11) as they were not important to the interrogator. On the other hand, personal questions with some relation to possible quest or the overall situation were rated high (10, 15, 16). Moreover, agent can of course use his episodic memory to obtain semantic information for questions 12, 14, 15 or to give a vaguer description of the raid on the village (4, 5).

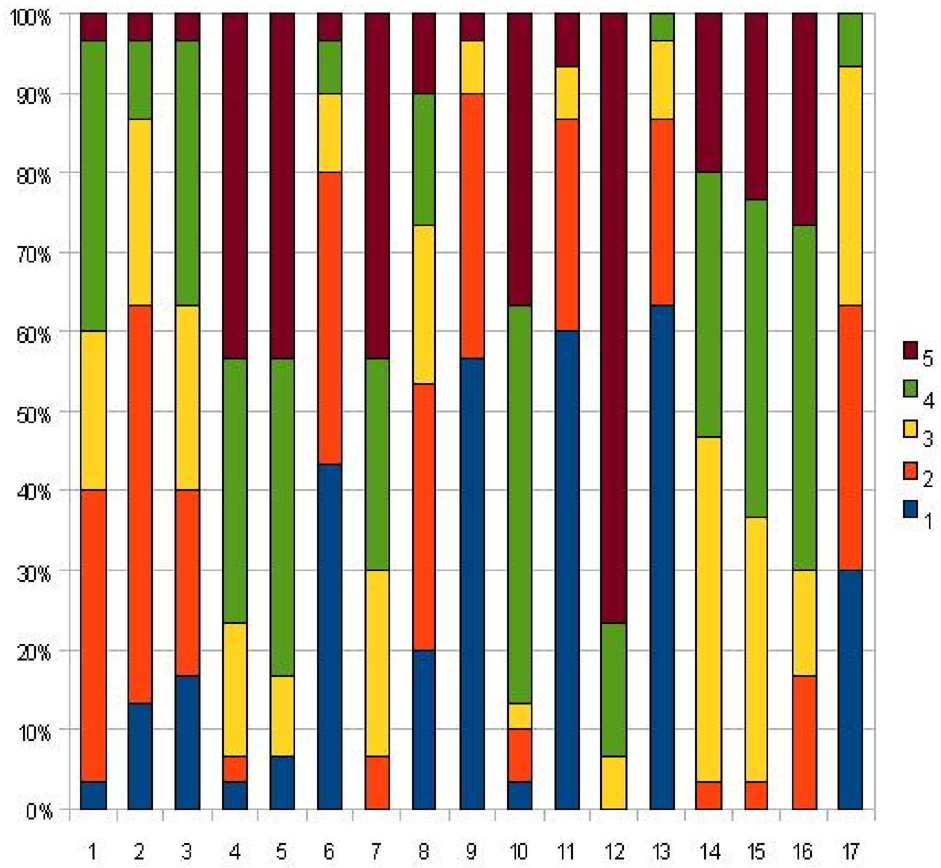


Fig. 16. Distribution of rates for each question.

Question	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
Average	3,0	2,4	2,8	4,1	4,1	1,9	4,1	2,6	1,6	4,1	1,7	4,7	1,5	3,7	3,8	3,8	2,1
STD	1	0,95	1,16	1,01	1,06	1,04	0,96	1,25	0,88	0,98	1,07	0,59	0,81	0,82	0,82	1,01	0,92

Tab. 3. Average response and the standard deviation for each question.

We can assess, that people would use rather socially established time patterns like parts of a day, “last week” than exact time cues (exact date or time) in their questions. They would not ask for detailed depiction of personal concerns of the agent but rather ask about potentially interesting events in his past. It should be noted that the results can be affected both by the composition of the questionnaire (the quest scenario, rather strange artificial questions) and the structure of participants (most of them were undergraduate students of computer science).

6.3 Query Module

Once we know what questions to expect we can board the issue of how to query the episodic memory for answers. We will start by summarizing types of queries from which will consequently emerge issues we have to address.

Usual time-query pattern is (1) “When did you do/use A?” or (2) „What did you do at time B?”. *A* can be replaced by any intention, action, atomic action or item agent used during the simulation. How do we retrieve time-related information from the network for *A*? We use *A* as the input point for the influx of activity. Then we observe where in the network drains off the highest amount of it. The amount is recorded along with the node. The result of the query is a list of weighted nodes. The list can be searched for concept nodes, day nodes, etc. The credibility of the result can be estimated from the weight. There are more problems hidden behind the propagation of activity through the network and we will address these later.

The second case is more complicated. *B* can be further decomposed into any combination of exact time, part of a day and/or day. The queries with a single time information (hour, part of a day, day) return usual activity at the given time. On the other hand combined queries for time and day or part of a day and day should return the activity at the specified time period. That reveal few issues to address:

1.The mapping from time to concept nodes.

2.The propagation of the activity via network.

3.Combination of results from different types of sources (e. g. different time scales).

We can use properties of the time network to solve issue (1). If the query contains exact hour or part of a day we infuse activity to the cartesian nodes corresponding to the provided hour or part of a day. Part of a day can be translated to a set of hours using some common definition of the part of a day (morning is from 7-10 a.m.). Then we collect the most active concept nodes – preferably few of them – and use them as inputs for the query to the connectionist memory. The given day (days) should be processed separately as it represents different time scale.

There are few options available for the issue (2). One of the possibilities is to propagate the activity from node to node with the limited total activity in the network. Then we are searching for an equilibrium and result is defined by the set of nodes in the equilibrium. More transparent and simpler solution is to propagate the activity from inputs only to adjacent nodes in the network. When the set of inputs contains several nodes (for instance, a set of concept nodes for part of a day) we sum up the activation from all sources for each node of the connectionist memory.

The last issue to board is how to combine activations from different types of sources (3) – for instance, different time scales like days and hours. There we used normalization of the vectors of weights from different sources and added them together creating ultimate result.

Hence the result of a query is represented by a weighted list of nodes of the network for connectionist memory. We have not implemented any linguistic module to create ordinary sentences from this output.

7 Experiments

Successful implementation of the proposed model provided us with the opportunity to test various hypotheses, see the model at work and evaluate its performance. We have conducted several experiments but before we represent our findings we outline the description of the setup.

The agent was simulated on a single machine – a personal laptop with AMD Turion64 X2 1.6 GHz CPU and 2GB RAM. The frequency of agent's logic was 14Hz – UT2004 was accelerated three times to allow for higher frequency. One step of logic represented 15 or 30 seconds in real life. The usual lifespan of an agent varied from experiment to experiment from two to four weeks. The agent did not have any real needs which if unsatisfied would lead to his death. He was following hard-coded master plans which were responsible for the basis of everyday plans. He was equipped with a simple model for biological needs which enriched the static plans with dynamically added desires. The agent had a fixed set of context nodes. There were two available sets of cartesian nodes – first with one node per hour, second with one node per half an hour. The initial number of concept nodes was 40.

The memory system contains many parameters which implies multiple paths for exploration. We have focused on the basic evaluation of the model. We have investigated the impact of different settings for cartesian and context nodes on the quality of learned concept nodes, the ability of agent to learn different timezones, the space requirements of the connectionist memory over time, the accuracy of recalls and the episode blending.

The chapter is organized in the following order. First section will be dedicated to the experiment concerning comparison of different lifestyles and the ability to learn new timezone. Then we will study the accuracy of recalls and its correspondence with empirical evidence from psychological studies. After that we will investigate the impact of different settings of cartesian and context nodes on the quality of concept nodes and the consecutive storage of episodes. Next experiment will examine the relation between agent's concept nodes and socio-biological temporal patterns. Afterwards we will under scrutiny the space demands of connectionist memory. The last experiment will explore the blending capabilities.

7.1 Timezone and Lifestyle Changes

Motivation.

We have stated (in section 6.1) that timezone changes can help us with the evaluation of the quality of concept nodes. We hypothesize that agent forms concept nodes properly disregarding the particular lifestyle. Consequently the quality of resulting concept nodes can be verified via agent's ability to adapt to different timezones. Thus we propose an experiment which features timezone shifts and three different lifestyles and we examine if the agent is able to learn the initial timezone and adapt when it shifts. The experiment also provide an opportunity to test the independence of the model on the particular lifestyle.

Settings.

The experiment proceeds according to the following scenario. The agent first learns time concepts – the time network – which takes him 8 days. Then he learns the initial timezone – 2 days. Then the timezone changes to +9 GMT and agent has seven days to live in that timezone. Finally the timezone changes to -3 GMT and agent has to adapt to that. We observe the magnitude and the position of the peak on the GMT wheel during the simulation.

We have prepared three different lifestyles to probe the independence of the model on particular lifestyle. Those were:

- **A travel salesman** is a man who wakes up every day at 7 a.m., goes to work, takes lunch at noon, works again, then go home where he usually watch TV after dinner. He occasionally goes to see a movie or a play but normally he lives quite normal and regular life.
- **A student** unlike the majority of students is attending lectures and studying fairly often. He is waking up at 8:30 a.m.. He practices swimming and ultimate frisbee in his free time and he goes occasionally to see a movie or a play.
- **A millionaire** is savoring his life freed from craving for money. He is waking up at 10 a.m. He is using his free time for variety of activities from sports, computer games and reading to some cultural events.

We should mention here the definition of the criterion which defines when agent has successfully learned a timezone. If agent stays in a state where real time equals thought time for one day, he learned the time-shift successfully.

We have run the experiment three times for each lifestyle. The agent lived for 21 days. The history of executed events for runs for the same lifestyle differed a little because of the influence of biological needs as well as the randomization of starts of activities (see section 4.1.3 – scheduling).

Results.

Illustration of resulting time networks is displayed on figures 17, 18, 19. Each figure provides a view of the time network with context nodes plotted in the top left, cartesian nodes in the top right and concept nodes at the bottom. Lines which are connecting them represent weighted lines between the neurons. Yellow lines connect contexts with concepts, green lines connect cartesians with concepts. The width of the line and deepness of the color depict the weight of the particular line.

All three different agents (governed by different lifestyles) were able to learn the initial time zone as well as the following shifts (fig. 20).

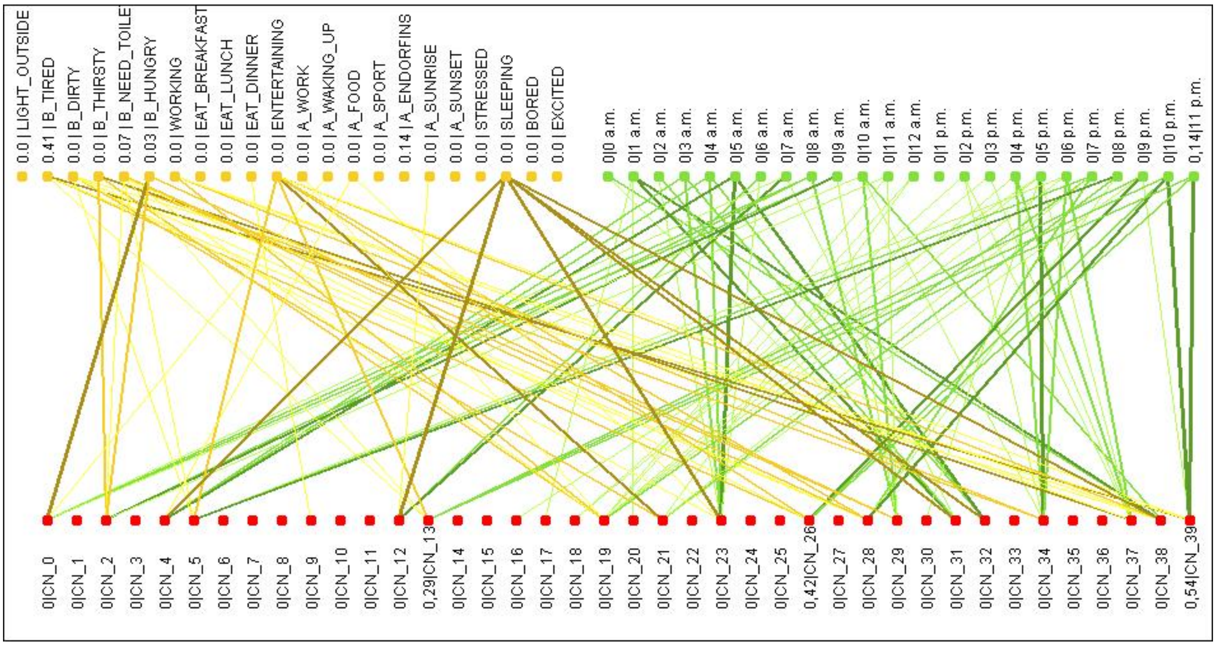


Fig. 17. A travel salesman network after 21 days of simulation.

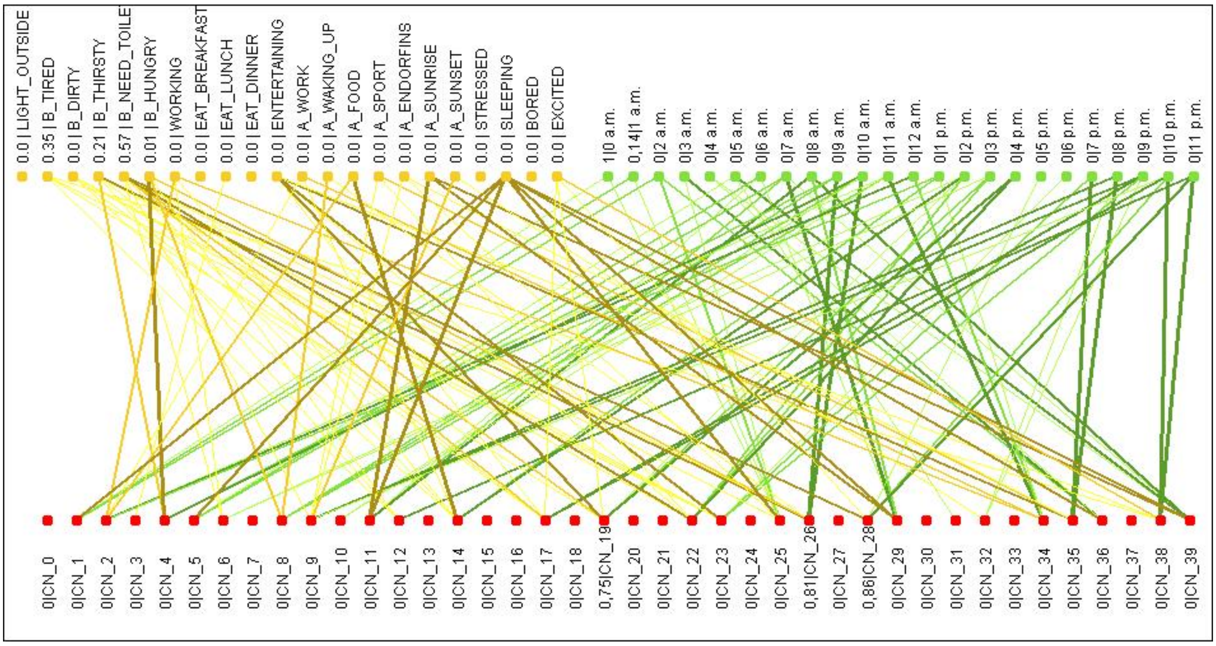


Fig. 18. The student's time network after 21 days of simulation.

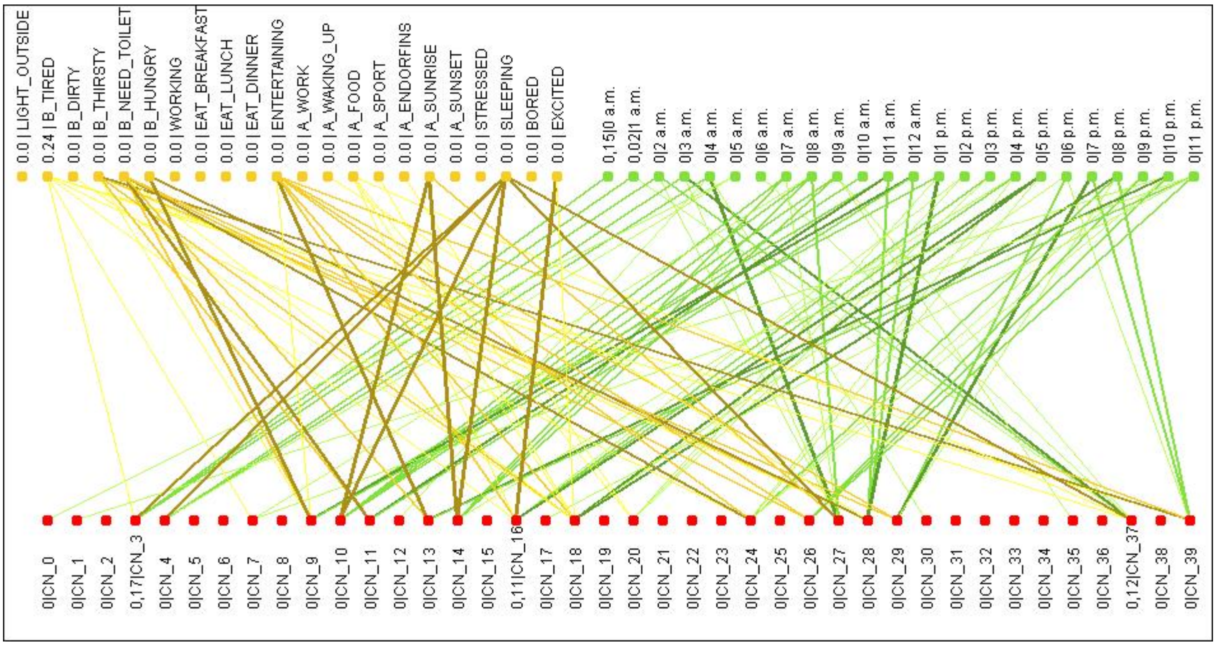


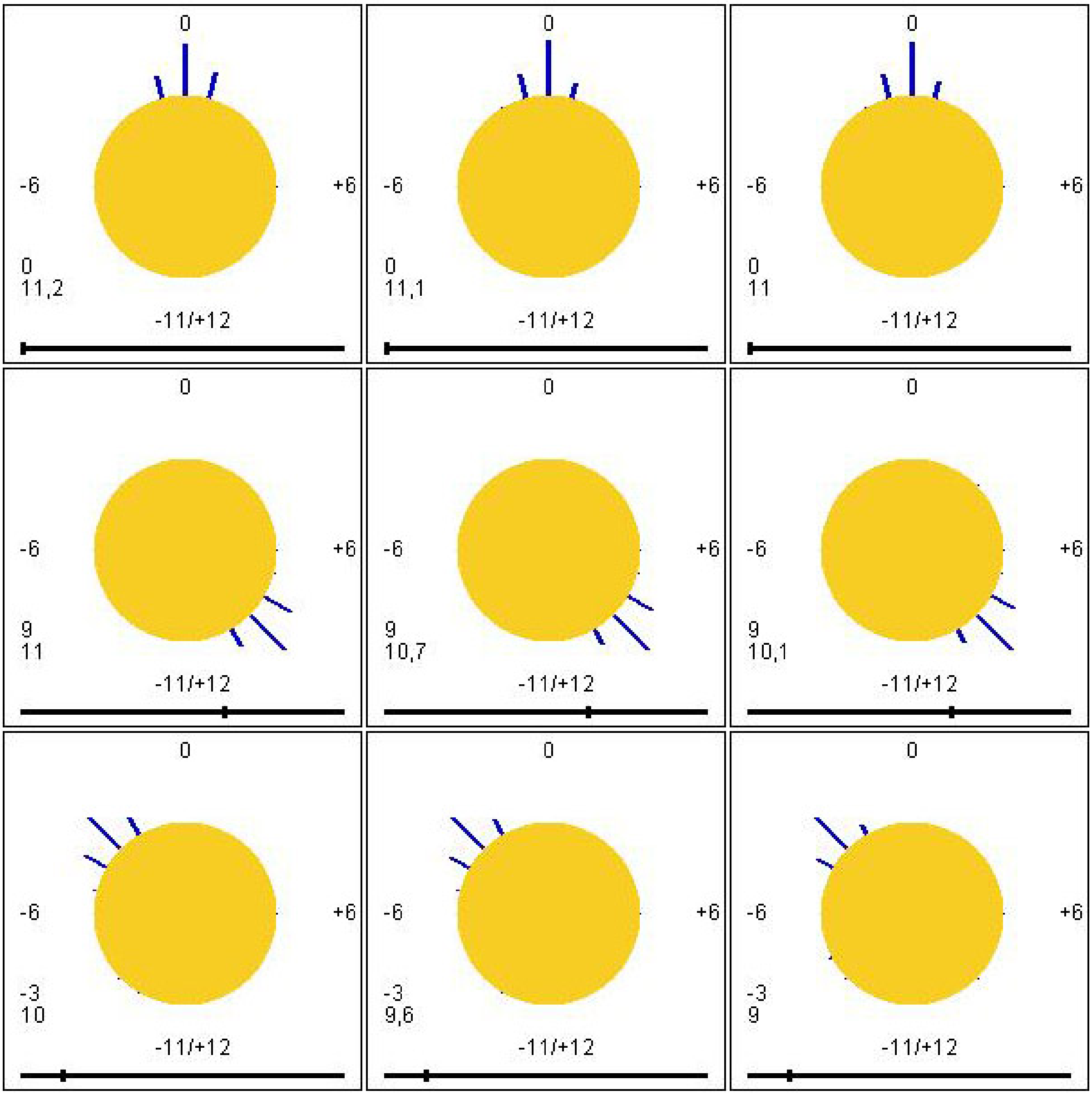
Fig. 19. The millionaire's time network after 21 days of simulation.

The visualization of weighted links to concept nodes does not reveal much. Nevertheless, we can deduce information like when is the agent sleeping or when he feels very hungry from it.

	Student	Worker	Millionaire
+9 GMT	37,7	41,75	37,75
-3 GMT	37,2	36,7	37,2

Tab. 4. The average times (in hours) of learning phase for the two changes of timezones for the three lifestyles.

We have not observed any discrepancies in the way agents learned time shifts (fig. 20). We found that they learned time concepts disregarding the differences between their lifestyles. Another thing to note is the time it took agents to learn a time shift. It took them usually day and half to adapt to the new timezone (tab. 4). Nonetheless, this number is arbitrary as it is parameter-dependent and can be further modified to match with an evidence obtained from psychology. We are not aware of any relevant research concerned about usual times for adaptation to a new timezone thus we cannot adjust the parameter correspondingly at the moment.



.**Fig. 20.** Results of timezone learning for different lifestyles – travel salesmen in the left column, student in the middle and millionaire in the right column are all learning 0 GMT first, then +9 GMT and finally -3 GMT. The numbers in the left bottom corners show the label of most active internal time node (for instance, GMT 0 for top left figure) and its activation (11.2 for the same figure).

Discussion.

We have observed that the timezone learning mechanism does not have a memory. It takes agent approximately the same time to do the transition forward and backward. In fact, if the agent lives in the new timezone for a very short time (less than two days), it can adapt back to the former timezone faster because the activity from the former timezone resides in the internal time structure for a while.

We can also explain the activations on the adjacent internal time nodes. The concept nodes are usually connected to few neighboring cartesian nodes. Thus the activation from the most active concept node is often propagated to the node in the middle as well as to his neighbors. Eventually the highest activity emerges in the middle.

7.2 Accuracy of the Connectionist Memory

Motivation.

The accuracy of the memory is one of the most important characteristics of the model thus we have carried out an experiment to measure the accuracy of the memory. We have measured the accuracy for two different modes of storing the episodes – the single structure for all days vs. a separate structure for each day (see 4.4). The hypothesis is following: the single structure is relatively accurate at the beginning of the simulation but become inaccurate over time due to the high interference of various events for various days. Thus it is outperformed by the multi-structure which does not suffer the interference problems.

The experiment concerned the recall of times and events for yesterday. We were asking the memory questions: (1) "What did you do yesterday at X o'clock?", (2) "What did you do yesterday Y", where Y is a part of a day, and (3) "At what time did you do activity Z yesterday?". In the cases 1 and 2, the answers were confronted with the data stored in the day log (which represented the perfect memory) in the following fashion:

1. The query (answer to the question) returns an event.
2. The event is localized in the day log.
3. The time of event obtained from the day log is compared with X or Y respectively.
4. If there is an overlap of intervals the difference is 0. Otherwise it returns the minimal distance of intervals.

The procedure for the question 3 was different.

1. Fetch all root goals which has a record in the day log.
2. For each root goal query the memory for time concepts corresponding to the goal and the day (yesterday).
3. Determine the part of a day which corresponds to the best concept node (highest credibility/weight of the result).
4. Determine if the event overlap with the part of a day (same with the step 4 from previous algorithm).

Obtained differences for hours/parts of a day/actions were plotted into a box-plot-whiskers graph every midnight. Thus the graphs show the progression of the accuracy over the lifespan of the agent.

Settings.

We have tested the accuracy for the two different modes of storing memory links – single structure which holds all links between concept nodes, events and resources in one structure for all days and multi structure which keeps this representation separately for each day. The key parameter for single structure is the weight limit for a node, which is used during the normalization. We have set this parameter arbitrarily to 10.

The simulation was run three times for the single structure scenario and three times for the multi structure scenario. We used the modified student lifestyle. The student was studying every morning of the week. In the afternoon he did four consecutive times ultimate frisbee and three times swimming. In the evening he went to the cinema or to the theater on a rotation basis. The rest of the activities was defined either by the frame of the day (sleep, nourishment) or by the dynamically invoked desires from biological needs.

Results.

The results are depicted in the graphs bellow – fig. 21, 22, 23. For first few days they can be a bit uncertain as the underlying time network undergoes its learning phase. Then the results for multi structure become steady whilst the interference starts affecting the accuracy of single structure – commence after six days. Then the difference between the accuracies steps out and we can observe the difference between multi structure and single structure.

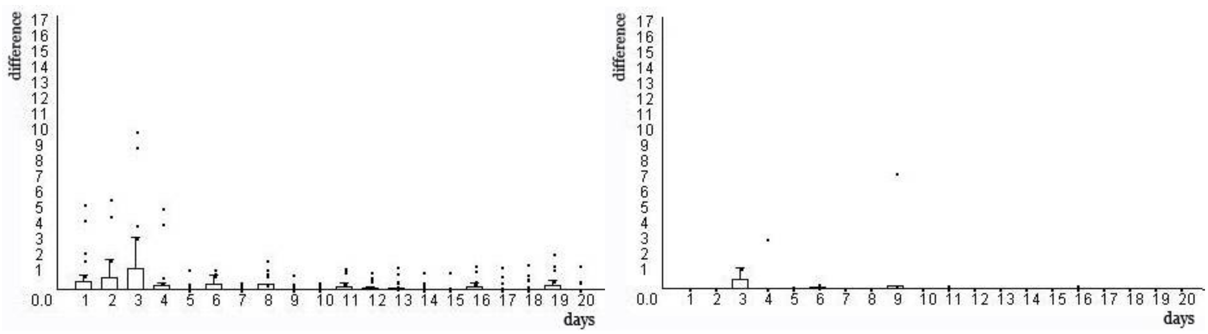


Fig. 21. Memory accuracy benchmark for hours (left), day parts (right) – multi structure, student lifestyle.

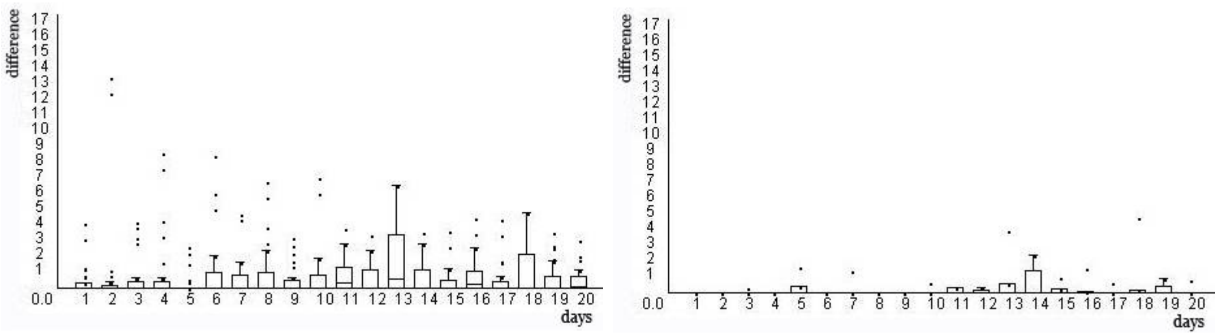


Fig 22. Memory accuracy benchmark for hours (left), day parts (right) – single structure, student lifestyle.

Discussion.

The error of the multi structure was almost zero for the parts of a day and very low for hours. On the other hand, the single structure average errors increased after five days and stayed superior to the results for multi structure. The results suggest that the single structure is not sufficient as a faithful representation for different days and is outperformed by the multi-structure (fig 21, 22).

Another interesting aspect is the comparison with psychological data. We use the data from experiment of Larsen, Thompson, Hansen [21]. They asked people to date events by choosing a group of hours from the 24 hours of the day. The most people had chosen groups of 3-7 hours (93%) and the correct time estimates were found twice as often as predicted by a chance. The accuracy was over 75%. Along with the described experiment we have recorded another graph which represented differences for the dating of actions – responses to question: “When did you do activity X?”. The results (fig. 23) shows remarkable accuracy of the memory for dating yesterday events.

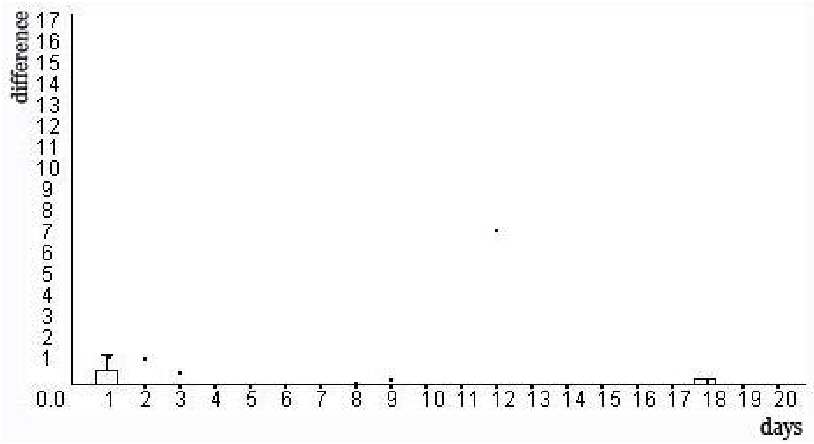


Fig. 23. Memory accuracy benchmark for actions – multi structure, student lifestyle.

We assess that the single memory accuracy deteriorates over the time along with the rising impact of the interference. There is a place for further investigation of the settings of the node weight limit for normalization and the capacity of the single structure. The multi structure behaves outstandingly on the task of dating yesterday events and giving answers to time queries for yesterday.

7.3 Impact of Cartesian Nodes on the Formation of Time Concepts

Motivation.

The formation of time concepts is strongly dependent on the activation of context and cartesian nodes therefore we have conducted two experiments inspecting the impact of different settings on the quality of time concepts. The initial motivation for the time concepts is that they should represent few cartesian nodes (hours, half an hours) and few context nodes (hunger, work) which are activated during the same period of time. It seems that it would be better if the cartesian nodes were from the same vicinity because the most active concept node is consequently linked with the event in execution. If the most active concept node links together distant cartesian nodes, it can lead to confused recalls (for instance recall of sleeping in the afternoon). But what has the impact on the clustering behavior of concept nodes? We presume the following hypothesis. If the cartesian nodes overlap the concept nodes group cartesian nodes from the same vicinity.

Hence we have defined a metrics to measure how much is a concept node *dispersed* over the time – cartesian nodes. The following formula:

$$\tau_i = \frac{\sum_{j=0}^{j=|J|} (\alpha(j) \cdot w_{ij})}{\sum_{j=0}^{j=|J|} (w_{ij})} \quad (7.1)$$

counts the *center of gravity* for a concept node i. Index j iterates over the set of cartesian nodes – J. w_{ij} is a weight between concept node i and cartesian node j. $\alpha(j)$ is the peak hour of the cartesian node. The weighted dispersion for the set of concept nodes is then counted:

$$dispersion = \frac{\sum_{i=0}^{i=|I|} \sum_{j=0}^{j=|J|} |\alpha(j) - \tau_i| \cdot w_{ij}}{cp\, nodes\, count} \quad (7.2)$$

The initial weights can shift the average to the middle thus we count only weights over 0.3 (initial random weights are up to 0.25).

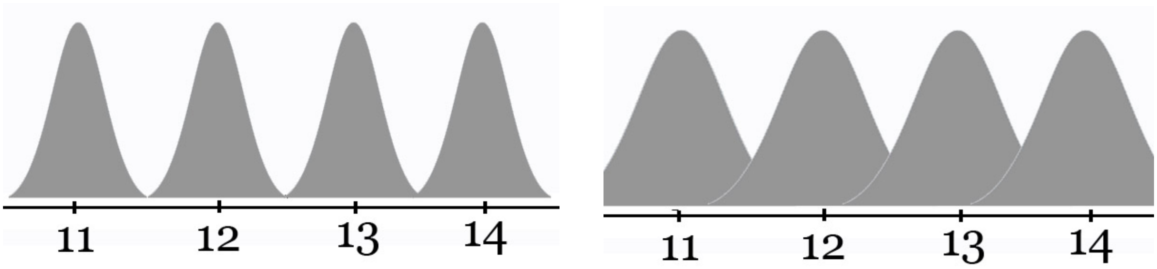


Fig. 24. Non-overlapping (left) and overlapping (right) Gaussian curves of activations of cartesian nodes.

Settings.

We have tried two different settings of the overlap of activities of adjacent cartesian nodes. First, the overlap of the nodes is was close to zero (<0.05) and second, when node's activity reached its peak (1.0) the neighbors had activation 0.36 (fig. 24). The weight limit for normalization for cartesian input into the concept nodes was set to 2,5 (resulting in 2-4 connections per node). The experiment was run for two different lifestyles – worker and student. Both were stationary, which means they scheduled the same actions for every day.

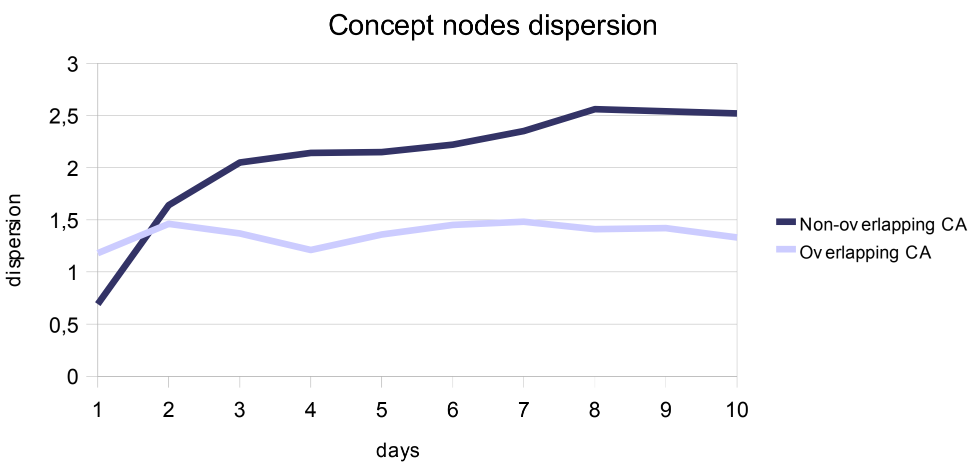


Fig. 25: The average dispersion over four runs (two for each lifestyle) for overlapping and non-overlapping cartesian nodes.

Results.

The experiment was run twice for each lifestyle and each setup of cartesian nodes. Fig. 25 shows that the dispersion is lower for overlapping cartesian nodes which supports our hypothesis. Low values for non-overlapping cartesians at the beginning are caused by the initial learning phase.

Discussion.

We have stated the presumption that compact time concepts are better as they are not linked with two unrelated time periods and consequently with two unrelated events. We have gathered the data about the accuracy of the memory along the experiment. We used the multi structure algorithm for days and plotted differences as was described in the previous section. The data clearly shows that non-overlapping cartesian nodes lead to malfunction of the memory for hours (fig. 26). The difference for day parts is not distinct (fig. 27) but we can observe a slight impairment for non-overlapping cartesians. We searched for the reason for the malfunction for hours and we found concept nodes which were connected to distant cartesian nodes (fig. 28) which resulted in recalls of eating in the middle of a night as well as sleeping in the afternoons.

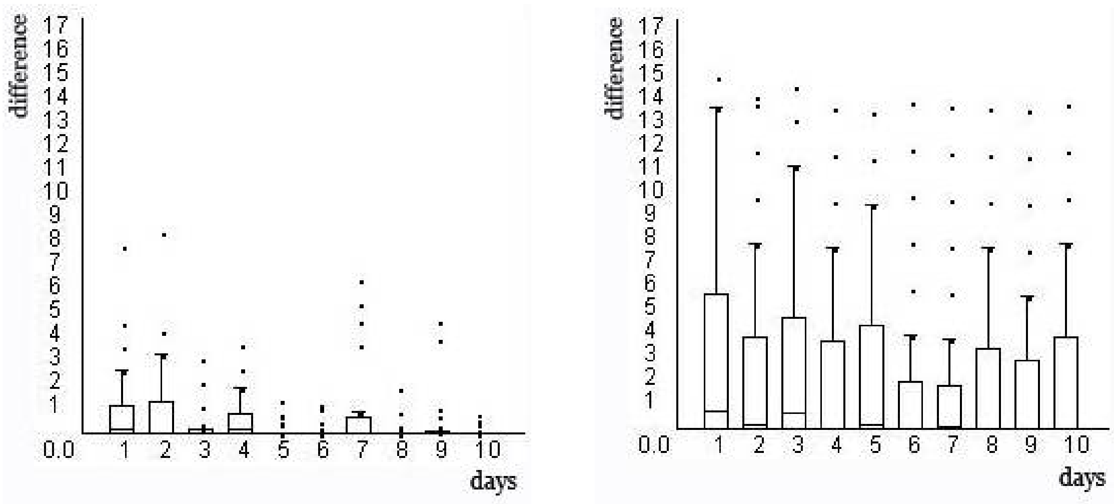


Fig. 26. Memory accuracy for hours – overlapping (left) and non-overlapping CA nodes (right).

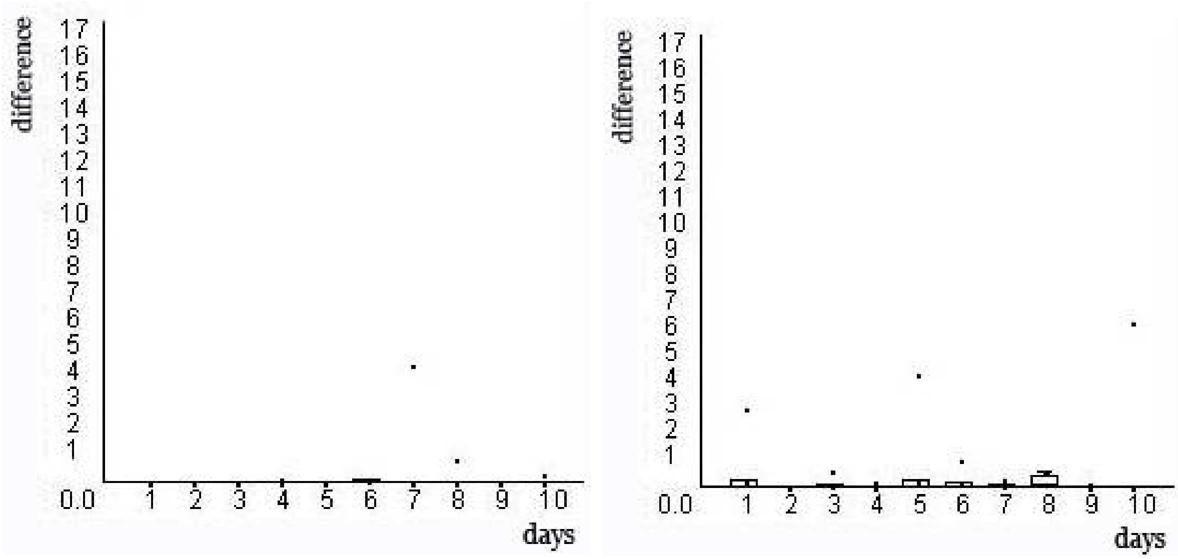


Fig. 27. Memory accuracy for parts of a day – overlapping (left) and non-overlapping CA nodes (right).

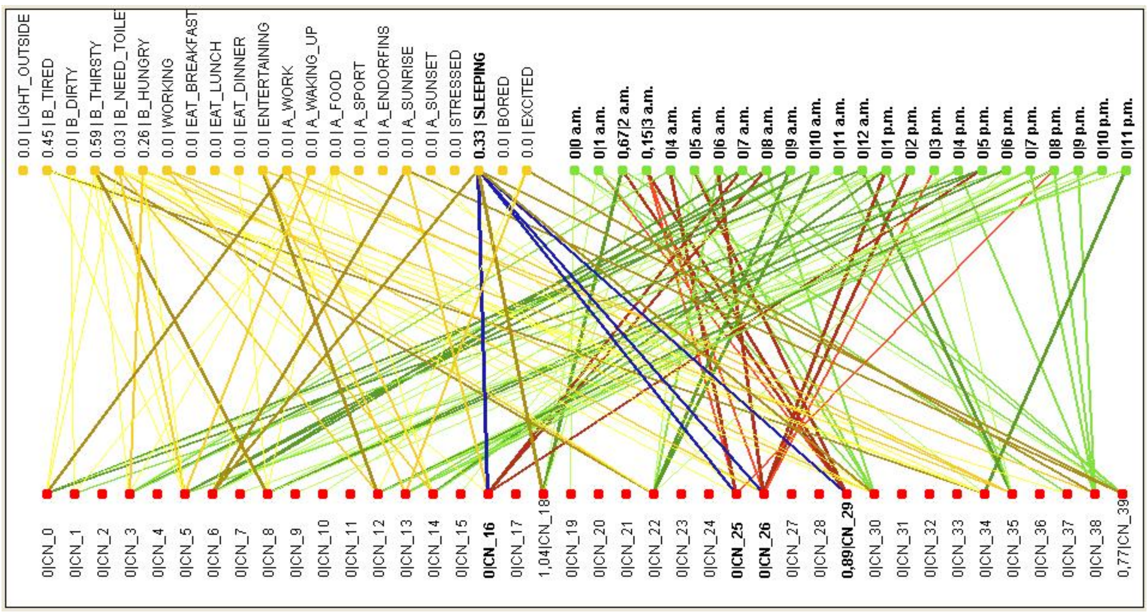


Fig. 28. The example of dispersed concept nodes – 16, 25, 26, 29 – of the time network for non-overlapping CA. The concept nodes are connected (red lines) both to CA nodes in the night and in the afternoon which results in recall of sleeping (blue lines) for afternoon hours.

Hence, we can assess that overlapping cartesian nodes form better time concepts than non-overlapping cartesians.

7.4 Impact of Context Nodes on the Formation of Time Concepts

Motivation.

The formation of concept nodes is further influenced by the activation of context nodes. It can be either decreasing, increasing or monotonous. A context node can be active as short as 20 minutes and as long as few hours. The activation can range from 0.0 to 1.0. Thus there is an exponential explosion of parameters and it is vital to explore the impact of activation of context nodes on the time concept formation.

Settings.

We have run several exploratory experiments with various settings. We used overlapping cartesian nodes with student's lifestyle. The experiments were following:

1. **No context activation.** The activation of context nodes was suppressed to zero. The formation of concept nodes relied only on the activation of cartesian nodes.
2. **Little cartesian activation.** The time network went through the usual learning phase (8 days) and then we have set the activation of cartesian nodes to 5% and measured the accuracy of memory recalls.
3. **Doubled activation of context nodes.** The activation of every context node was doubled while keeping the activation of cartesian nodes on the same level.

Results.

We have run each experiment twice and then examined the resulting time network and memory accuracy. The results were following:

1. **No context activation.** The concept nodes were clustered suitably (fig. 29). The accuracy of memory recall was comparable with the accuracy of standard settings.
2. **Little cartesian activation.** The accuracy of recalls was unimpaired despite of the lower magnitude of cartesian input.
3. **Doubled activation of context nodes (fig. 30).** The recall was moderately impaired for hours and unimpaired for day parts.

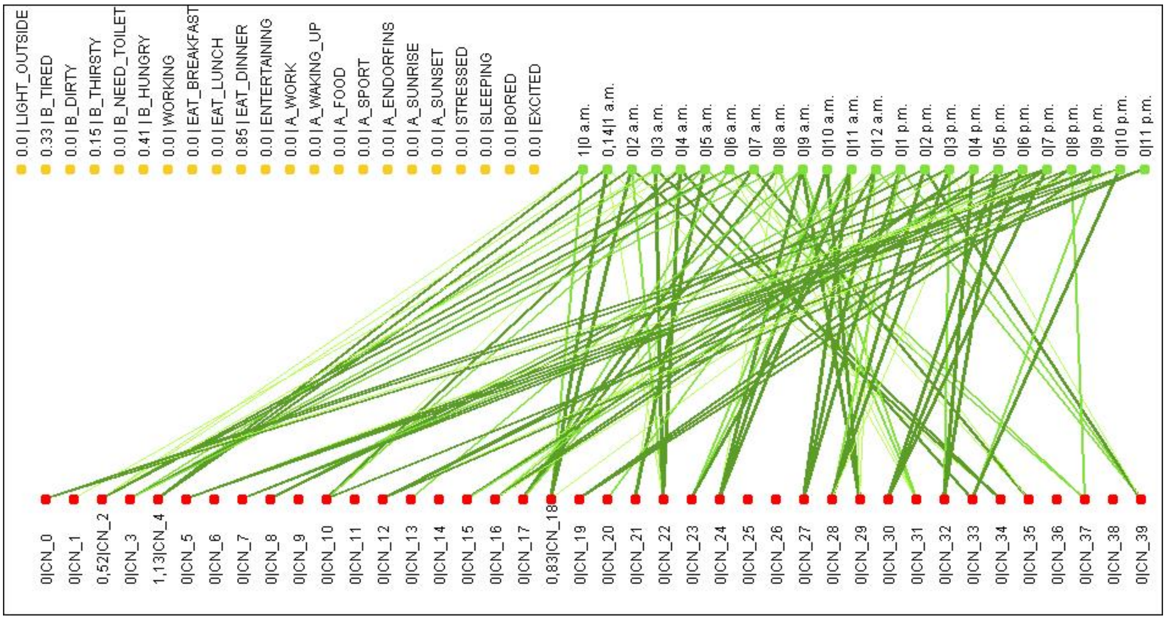


Fig. 29. The time network for experiment with context nodes turned off.

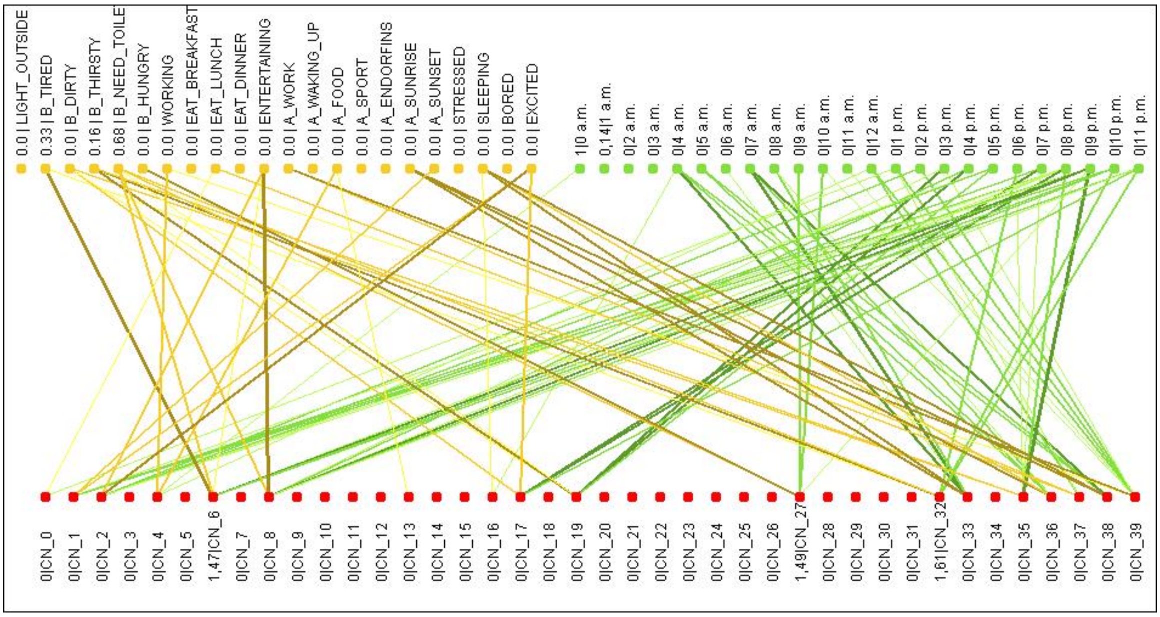


Fig. 30. The time network for experiment with doubled activation of context nodes.

Discussion.

The results of experiments suggest that it might be possible to learn concept nodes only on the basis of cartesian nodes. But that will have two consequences. First, the time nodes would be very similar and will not reflect the lifestyle – many concept nodes for the morning etc. Second, we will loose another source of information for the reconstruction process of recall. The contextual links can provide us with fast answers to questions like “When are you usually hungry, working, sleeping...?”. The lesson learned from the experiment is that activations of context nodes should be rather inferior to the activation of cartesian nodes and it is worth to keep them in the model.

7.5 Labeling of Time Concepts

Motivation.

The goal of the sixth experiment was to determine whether the resulting time concepts have any relation with socially agreed time patterns. Larson et al.[21] showed that people create for themselves about six temporal concepts for parts of a day. Thus we have tried to label concept nodes by six following labels: morning, noon, afternoon, evening, late evening and night. Each of these (apart from night) represented three or four consecutive hours (tab. 5) which is in compliance with the study of Larson et al. [21].

Label	Hours
Morning	7, 8, 9, 10
Noon	11, 12, 13
Afternoon	14, 15, 16
Evening	17, 18, 19, 20
Late evening	21, 22, 23, 0
Night	1, 2, 3, 4, 5, 6

Tab. 5. List of hours for corresponding time-concept labels.

The assignment can be either automatic or handmade. We have exploited the property of links between cartesian nodes and concept nodes. We inject an activity to hours of a part of a day to determine the average activity for each part of a day for each concept node. The most active part of a day then labels the concept node. The algorithm is following:

```
Double maximum = 0, sumForPartOfADay;
for (PartOfADay dayPart : PartOfADay.values()) {
       sumForPartOfADay = 0;
       for (Hour hour : dayPart.hours()) {
              sumForPartOfADay += getActivityForHour();
       }
       if (sumForPartOfADay > maximum) {
              maximum = sumForPartOfADay;
              conceptNode.automaticName = dayPart.getName();
       }
}
```

Results.

The labeling of concept nodes was used in all performed experiments. More specifically, it was used to determine corresponding concept nodes when querying the memory with questions which contained part of a day. We list here an illustrative example of the result of labeling (fig. 30) for one learned time network (fig. 31).

```
CN_1 LATE_EVENING 0,3
CN_3 NIGHT 0,4
CN_4 NIGHT 0,34
CN_7 MORNING 0,22
CN_9 NOON 0,67
CN_10 MORNING 0,52
CN_11 NOON 0,6
CN_13 LATE_EVENING 0,52
CN_14 MORNING 0,47
CN_16 AFTERNOON 0,42
CN_18 EVENING 0,47
CN_19 LATE_EVENING 0,21
CN_20 AFTERNOON 0,24
CN_24 LATE_EVENING 0,36
CN_25 NOON 0,22
CN_26 LATE_EVENING 0,42
CN_27 NIGHT 0,5
CN_28 NOON 0,51
CN_29 EVENING 0,51
CN_33 AFTERNOON 0,24
CN_35 NOON 0,21
CN_37 NIGHT 0,39
CN_39 EVENING 0,41
```

Fig. 30. Corresponding labels of time concepts for millionaire lifestyle. The notation is: anonymous name of the node – assigned name of part of a day – credibility of the assignment.

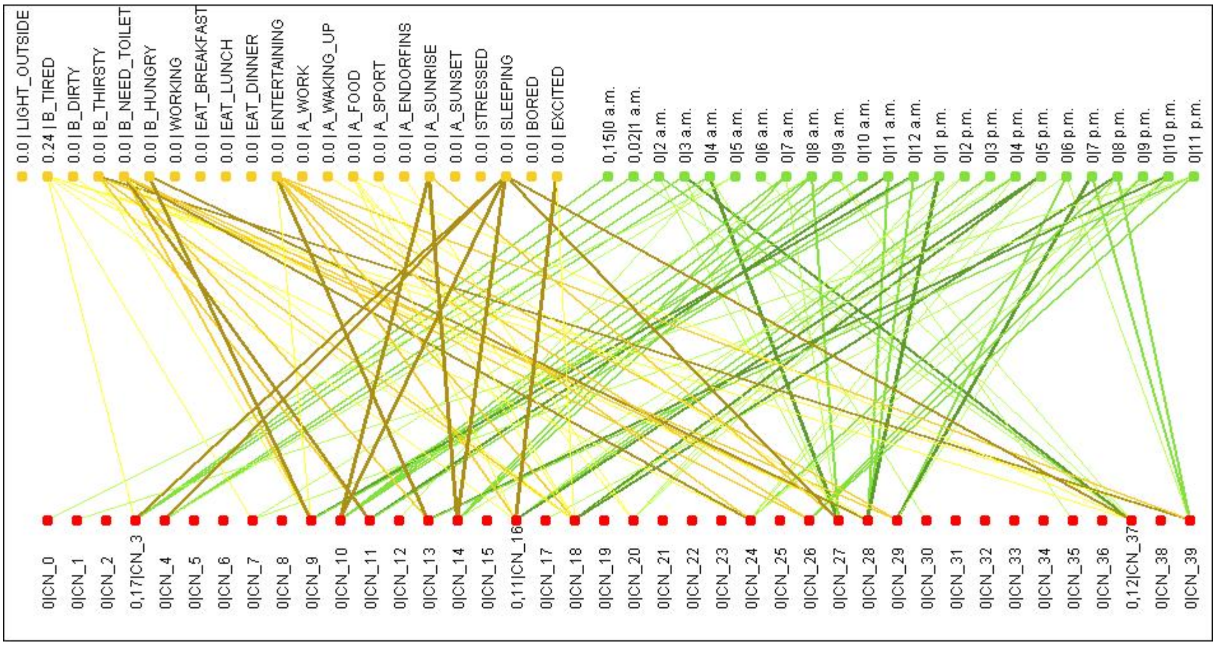


Fig. 31. The time network for millionaire lifestyle which corresponds to the label assignment from fig. 30.

Discussion.

The result of this experiment is disputable. On one hand, we can conclude that we are able to assign reasonable labels to obtained time concepts. Thanks to the algorithm we get them along with a level of credibility and it seems that set of nodes with same label represents the part of a day faithfully. But there is one aspect we should not neglect. People do not create the same parts of a day. What is morning for someone can still be night for someone else. Humans' time concepts are culturally and personally dependent. Thus we cannot claim that the concept nodes with a given label really represent the part of a day. What we can say is that there exists a mapping from concept nodes to ordinary sociologically and biologically defined parts of a day which can be further used for querying purposes.

7.6 Evaluation of the Memory Requirements

Motivation.

One of the fundamental characteristics of the memory is its space demand. Thus it is vital to provide a comparison of the proposed model with another options of episode storing. We will review requirements of our model and compare it with the abstraction of logging-based model.

Settings.

We have tested the demand for memory on the agent who ran according to the student lifestyle. The cartesian nodes were overlapping. The lifespan of the agent was 30 days. We have run the simulation twice for each mode of storage (single structure, multiple structures). We have stored everything in the memory – every task, every desire, every item, every concept node involved in the episode. Thus we have stored, for instance, *Want* desires which are responsible for the localization of resources, etc. The forgetting coefficient for links between concept nodes, actions and items was set to 0.9. The decrease of day budget was set to 0.85. The bias when a link was discarded was set to 0.3.

Results.

We have counted every line from every node – either concept nodes, actions, items, days. The average sum of nodes contained in the memory is depicted in the fig. 32. The figure shows the evolution of the demands over the lifespan of the agent.

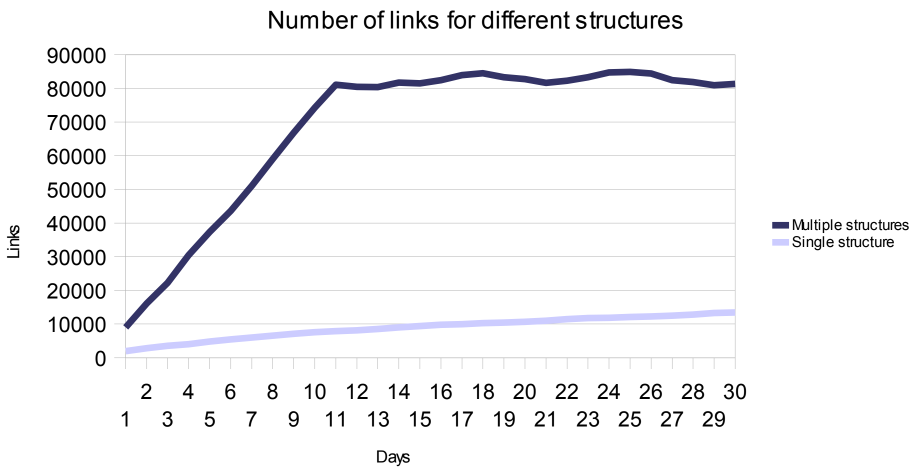


Fig. 32. Average number of links for a memory over 30 days of simulation.

Discussion.

The experiment shows that the demand for memory does not grow beyond control. It depends a lot on the settings of forgetting functions and mechanisms. Implemented mechanisms lead to the oblivion of detailed time information after less then two weeks and day information after about three weeks. Thus the memory demands stabilize after two weeks. But the psychological evidence suggests that people can retain a lot more than last two weeks. It suggests also that they retain mostly important or emotionally significant memories which adverts to the problem of the rating mechanism for the interestingness of events.

The number of links for multiple structures is immense. We will make a comparison with a log, which is logging every action agent takes, its time, its parents (to reconstruct the episode). Hence either every line of the log contains a lots of redundant information – like the name of the intention, action – either the log operates over some more complicated structure and notes every change in the situation by adding a note somewhere. That is exactly what we are doing in the multiple structure mechanism. Every time there is a change in the situation we add a link. Moreover, the link is lightweight. It is a triple *(node, weight, day)* where node is a pointer to concept node, action or resource, day is a pointer to the particular day and weight is a real number. Hence the link can be as small as 10B.

The most important difference between the proposed model and the log-based memory is the forgetting mechanism. It can be quite difficult to model forgetting in the log-based model in other way than by deleting oldest records. On the other hand, the connectionist model can profit from weighted links and model human-like forgetting mechanism via extinction as well as the interference. Moreover, if we use some mechanism to define the importance of an event, the interesting events will eventually persist longer in the memory.

Another advantage of the proposed model over the log is the recall mechanism. Our memory is indexed in multiple dimensions. We can use practically any cue in the query from time over actions to resources keeping down the complexity of the query. The number of operations of a query always depends only on the number of input points (cues) and the number of neighbors of each node.

Moreover, the total number of links can be further decreased by (a) discarding immediately links for search tasks and by (b) using the general day structure for retention of frequent events and activities thus not being forced to store them for separate days.

7.7 Episode Blending

Motivation.

We have outlined three core problems of previous memory model of episodic memory. One of them was the episode blending. The episode blending means, that two similar episodes are merged into one while loosing some details (like the exact time of execution). The model accounts for that by introduction of multi-days – e. g. weeks or “two-days”. Then if the same activity is executed in the afternoon, for instance, five times a week agent should recall that he was doing the activity that week afternoons. He should also remember that he was doing it the five separate days, but this information would deteriorate faster as the link to week was strengthened five times. Eventually the only trace left would be for afternoons of the week.

Settings.

We proposed an experiment which tests this hypothesis. The agent had a modified student plan which contained one activity for afternoon on four consecutive days and another on the remaining three days and two alternate activities for evenings. The simulation then run for three weeks. Every midnight the agent executed a query which retrieved actions for the part of a day (afternoon and evening) for last seven days plus for the last week. Then we examined the results and compared them with the log of agent's activities.

Results.

The experiment was run five times and the results were examined. We will present here one illustrative example of the work of the recall for weeks followed by the explanation of the result. We will explain the results of the query for the second week (fig. 33). The listing shows seven most active nodes of the connectionist network for the input from evening and corresponding day or week. There are several interesting aspects to note:

1. The agent is alternating going to the theater with going to the cinema during the week. He starts with a theater thus he goes to the theater four times and to the cinema three times. That results in the "last week" answer which prefers theater over cinema.
2. The agent played computer games three times that week evenings. The answer for the week contains playing computer games while none of the days considered it worth noting. The reason for this behavior is that the activity did not overlap enough with the evenings of single days, nonetheless, the activity summed up over the week to emerge in the week result.
3. Results for days contain four times eating. The dinner occurs around 18 o'clock. If it took him long time to satisfy the goal for eating, the agent remembers it more than going to the theater or the cinema. The problem of the evening activities is that they usually start in the evening and overlap to the late evening which renders them weaker – they are connected to fewer concept nodes which are defining evening. That is one of the deficiencies of the socially defined hard-wired definitions of parts of a day.
4. The eating happened every day. It was noted in most of the days, but it has only one record in the week answer. That is because the agent chose few different ways to fulfill the desire to eat. Thus the root desire links were strengthened every day and *IEat* occurs in the week result, but the rest of the details is lost during the recall.
5. The activities for swimming and studying were not scheduled for the evening but they simply overlapped to the evening.

```
last week:   [ISeeAPlay, 1,451], [ASeeAPlay, 1,451], [IEat, 1,333],
             [ASeeAMovie, 1,163], [ISeeAMovie, 1,163],
             [IEntertainment, 1,135], [APlayComputerGames, 1,133]

1 day ago:   [IEat, 1,174], [ACookSomething, 1,174], [AShopFood, 1,046],
             [IEquipUp, 1,046], [SHOP, 0,95], [Carrefour, 0,95],
             [ADrink, 0,792]

2 days ago:  [IEat, 1,172], [AEatAtResto, 1,172], [AFindAResto, 1,143],
             [IFindAResto, 1,143], [ISeeAMovie, 0,832],
             [ASeeAMovie, 0,832], [ADrink, 0,813]

3 days ago:  [ISeeAPlay, 1,118], [ASeeAPlay, 1,118], [ADrink, 0,98],
             [IDrink, 0,98], [AEatAtResto, 0,943], [IEat, 0,943],
             [AFindAResto, 0,886]

4 days ago:  [ISwimmingTraining, 1,175], [AGoSwimming, 1,175],
             [AEatAtResto, 1,086], [IEat, 1,086], [Swimming pool, 0,96],
             [SWIM, 0,959], [EAT, 0,883]

5 days ago:  [IEat, 1,181], [AEatAtResto, 1,174], [ADrink, 1,174],
             [IDrink, 1,174], [AFindAResto, 1,089],
             [IFindAResto, 1,089], [Tap, 0,934]

6 days ago:  [IEat, 1,292], [ACookSomething, 1,292], [ADrink, 1,153],
             [IDrink, 1,153], [AShopFood, 1,019], [IEquipUp, 1,019],
             [IGoToToilet, 0,966]

7 days ago:  [IStudyFix, 1,178], [AStudyAtSchool, 1,178],
             [School, 0,999], [SIT_AT_THE_LECTURE, 0,999],
             [ISitAtTheLecture, 0,999], [ASitAtTheLecture, 0,999],
             [IEat, 0,803]
```

Fig. 33. Example of result of query for evenings (17-20 o'clock) for last week and corresponding days. The explanation is given in the text.

Discussion.

The experiments showed that agent can recall the most frequent activities over the less frequent when asked for the last week. The traces to the week are much stronger than traces for each day which in long term should result in the oblivion of the traces for days whilst keeping the trace for the week → episode blending. It also accents the necessity to define the interestingness of events which would diminish recalls of desires like *IEat*, *IDrink*, *IToilet*.

We can assess that agent is able to give reasonable answers to questions which contain week as a cue and that the memory is able to blend some episodes in the long-term perspective.

7.8 Findings Summary

Before we proceed to the future works and conclusion we would like to sum up findings of performed experiments. We have showed that agent is able to adapt to new timezones using his time network to synchronize the internal time with the real time. The ability to learn time shifts is not impaired by the difference between lifestyle.

The accuracy of the recall of episodic memory has fulfill our expectations. The accuracy of single structure becomes impaired in the long-term perspective as the interference affects the storage. On the other hand multi structure mode of storing shows solid results.

Then we tried to determine the suitable settings of cartesian and context nodes which would form better concept nodes – e.g. with better accuracy of recalls. We found that the activation of cartesian nodes should be overlapping because the resulting concept nodes are then connected to cartesian nodes in closer vicinity.

The activation of context nodes has also an influence on the formation of concept nodes. We tried experiments which explored extreme settings – either high activation or none. The results suggest that context nodes are important for higher diversification of concept nodes as well as their correspondence to the particular lifestyle but high activation produces undesirable modifications to the learning process.

Afterwards we tried to label concept nodes with human understandable names. We used a simple algorithm which exploited features of the time network. The assignment produced reasonable names for concept nodes. Nevertheless, it is disputable if such an assignment can reflect the internal time patterns as for instance people communicate using the sociologically agreed time patterns whilst having their own and making this transition on the fly.

Consequently we have put under scrutiny space requirements of the episodic memory. We have argued that the memory can be more efficient than the log in the long-term perspective and it allows for easy and fast execution of variety of different queries.

The last experiment engaged the problem of episode blending. We have showed that agent can faithfully reply to questions which contain a week as a cue and that the forgetting can result in the episode blending. For instance, the agent will remember that he was eating in the evenings of one week, but he will forget the details – like which day he went to the restaurant and which day he cooked something at home.

In the end we can assess that we have successfully implemented the prototype of an agent who lives in a complex environment and who features autobiographic episodic memory extended by the time perception. It is obvious that the current design and implementation can be further improved in many ways (as will be discussed in the next chapter). Nevertheless, we believe that the project has high potential and shows an interesting course for the future research. The accuracy of the memory is compelling and if properly subsidized by a more complex forgetting with a mechanism for the interestingness of an event it can be even more faithful, efficient and less space demanding.

8 Future Works

We have ventured into the vast and mostly undiscovered area of modeling episodic memory for a virtual agents using a connectionist model. Presented results of the prototype are very promising but as it is only a prototype it leaves multiple research paths unexplored. We will try to summarize a variety of possible improvements, experiments and scenarios in the following text.

Starting from the very bottom of the implemented agent we can enrich agent's world with a deal of items, places, other agents and human players creating more realistic background for the simulation. Along with this extension goes enlargement of the set of desires as well as the depth and width of corresponding AND-OR trees. All those improvements should be easily achieved in the new version of Pogamut 2 which should provide a better support for storytelling.

Following time-perception model can be enhanced by the hierarchy of cartesian nodes starting from minutes from which will evolve automatically created nodes for hours or even parts of a day. Hence they will not be sharp as the hours or half-an-hours we use in the model.

The second part of the input of concept nodes is the context nodes layer. We believe that context nodes can be created automatically if we have some mechanism which monitors agents state and the state of environment and then some algorithm which translates output of the monitor into context node activations. The biological needs model is an example of how it should work.

As we climb up along the flow of activation through the model, we arrive at concept nodes. We have made some preliminary tests to determine if the time network which has learned one lifestyle can adjust to another lifestyle. For instance, if it can account for the difference between weekend and week days (Huttenlocher et al. [33]). Tests showed that the network is quite rigid and does not adjust much. We presume that this problem can be solved by adding a new set of concept nodes for each new lifestyle. It would require a mechanism which would determine a critical level of difference between two lifestyles and, if there were more learned, the one the most suitable for the lifestyle in use.

Last but not the least are connections between trees, concept nodes, days, etc. Many episodic memory theories propose that the persistence of a record – e.g. the time it lasts, the level of details stored along – depends a lot on the emotional state of the individual during the event. It would be tempting to use our model along with an emotional model.

The retention of important episodes can be further enhanced by introduction of mechanisms which would strengthen its traces whenever the episode is retold, revised or invoked during a night via dreams.

We can put under scrutiny the content of the memory. There is an abundance of links which interconnect resources and actions which are not used at the moment. For instance, if the agent performs an action which requires a nail and the hammer, it would create a semantic link between the two. If we want to determine if the agent did something at least once, it suffice that it is present in the memory even though there is not a single link to it anymore.

Other improvements can be done with the testing framework. The agent can be adapted to the Pogamut GRID [34] which enables use of multiple machines to run various experiments. Then the parameters of the model can be optimized using, for instance, evolutionary techniques.

9 Conclusion

We have designed a module for episodic memory for virtual humans. We have successfully implemented a prototype in the complex continuous environment of Unreal Tournament 2004 using the platform Pogamut 2. We have performed a series of tests and experiments to evaluate its performance and properties.

The former model of episodic memory sustained following problems: excessive accuracy of responses, no error proneness and the inability to perform episode blending. The proposed model untangled these challenges by introduction of unique system for time perception and the connectionist model for episode storage.

The time perception is achieved through automatically formed time concepts. Time concepts are inspired by the psychological research on the human time perception and the related time estimates for past events. Time concepts are clustering time together with contextual information. The quality of obtained patterns was verified by a series of experiments. We have proposed a mechanism which enables agent to adapt to timezone changes and used it for the time concept verification. We have showed that it is possible to assign socially valid labels (widely agreed time patterns) to learned concepts. Moreover, we have explored the role of key variables which have impact on their formation.

Time concepts are a basis for dating events in the connectionist memory. The connectionist memory stores episodes using a number of links which interconnect events, resources and time information which were active at the same time. The links are then used for the episode retrieval. Consequently, the agent can answer vaguely specified time-cued questions and he can use socially agreed temporal patterns in his answers.

We have carried out several experiments concerning the evaluation of the resulting episodic memory. We have showed that it is able to retrieve correct data from the past with a decent accuracy. The key elements for the space demand of the memory are the settings of forgetting and the definition of importance of an event for agent. Nevertheless, we still cannot account for the phenomenal human memory which can date more than 20 years old memories with remarkable precision. We believe that our mode can help in the pursuit of believability of intelligent virtual agents.

Literature:

[1] World of Warcraft. http://www.worldofwarcraft.com/, [10.4.2009]

[2] Aylett R.S., Louchart S., Dias J., Paiva A., Vala M. (2005): “FearNot! – An Experiment in Emergent Narrative.” In Proceedings of Intelligent Virtual Agents, LNAI 3661, Springer, 305–316.

[3] Hodges, L.F., Anderson, P., Burdea, G.C., Hoffman, H.G., Rothbaum, B. O. (2001): “Treating Psychological and Physical Disorders with VR.“ In Proceedings of the 2001 of Computer Graphics and Applications, IEEE 2001, 25-33.

[4] Cavazza, M., Charles, F., Mead, S.J. (2004): “Developing Re-usable Interactive Storytelling Technologies.“ In Proceedings of the 2004 of IFIP World Computer Congress (Toulouse, France).

[5] Loyall, A. B. (1997): Tech report CMU-CS-97-123. Ph.D. dissertation, Carnegie Mellon University.

[6] Nuxoll, A. (2007): Enhancing Intelligent Agents with Episodic Memory. Ph.D. diss. The University of Michigan.

[7] Tulving, E., Donaldson, W. (1972): Organization of Memory. New York: Academic Press.

[8] Peskova, K. (2006): Model paměti pro animata. Bachelor thesis. The Charles University. Prague (in Czech).

[9] Gibson, J. J. (1979): The Ecological Approach to Visual Perception. Boston: Houghton Muffin.

[10] Bratman M. E. (1987): Intention, Plans, and Practical Reason. Cambridge. Mass: Harvard University Press.

[11] Squire, L. R., Zola-Morgan, S. (1991): The Medial Temporal :Lobe Memory System. In Science 253: 1380-1386.

[12] Kolodner, J.K. (1983): Reconstructive Memory: A Computer Model. Cognitive Science, 7 (4), 281-328.

[13] Bahrick H. P. (1984): Semantic Memory Content in Permastore: Fifty Years of Memory for Spanish Learned in School. Journal of Experimental Psychology: General, 113, 1 – 29.

[14] Roediger, H.L., McDermott, K.B. (1995): Creating False Memories: Remembering Words not Presented in Lists. Journal of Experimental Psychology: Learning, Memory, and Cognition, 21, 803-814.

[15] Sternberg R. J. (2002): Cognitive Psychology. Portál, Prague.

[16] Bartlett, P. F. (1932): Remembering: A study in Experimental and Social Psychology. Cambridge, England: Cambridge University Press.

[17] Godden, D.R., Baddeley, A.D. (1975): Context-dependent Memory in Two Natural Environments: On Land and Underwater. British Journal of Psychology, 66, 325-331.

[18] Flaherty, M. G. (1999): A Watched Pot: How We Experience Time. New York: New York University Press, 230 pages, paperback, ISBN: 0814726879.

[19] Friedman, W.J. (1993): Memory for the Time of Past Events. In Psychol Bull 113(1): 44 – 66.

[20] Friedman, W.J, (1987): A follow up to "Scale Effects in Memory for The Time of Events": The Earthquake Study. Memory and Cognition, 15, 518-520.

[21] Larsen, S.F., Thompson, C.P., Hansen T. (1995): Time in Autobiographical Memory, In: Remembering Our Past: Studies in Autobiographical Memory, New York: Cambridge University Press, 129-156.

[22] Artificial Minds for Intelligent Systems: http://artemis.ms.mff.cuni.cz/, [1. 4. 2009]

[23] Brom, C. (2009): Towards Virtual Characters with a Full Episodic Memory II: The Episodic Memory Strikes Back. In: Proc. of Workshop on Empathic Agents. AAMAS 2009.

[24] Dodd, W. (2005): The Design of Procedural, Semantic, and Episodic Memory Systems for a Cognitive Robot. Master thesis. Vanderbilt University, Nashville, Tennessee.

[25] Johnson, W. L. (1994): Agents That Learn to Explain Themselves. In Proc. of the 12th Nat. Conf. on Artificial Intelligence, 1257-1263. AAAI Press.

[26] Dayan, P., Abbott, L.F. (2000): Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. Cambridge: MIT Press, c2001, Chapter 8.

[27] Pogamut 2: http://artemis.ms.mff.cuni.cz/pogamut/, [1. 4. 2009]

[28] Burkert, O., Kadlec, R., Gemrot, J., Bída, M., Havlíček, J., Dörfler, M., Brom, C. (2007): Towards fast prototyping of IVAs behavior: Pogamut 2. In: Proceedings of 7th International Conference on Inteligent Virtual Humans, LNCS Vol. 4722. Paris, France. Springer-Verlag, Berlin.

[29] Epic Games: Unreal Tournament 2004: http://www.unrealtournament.com, [1. 4. 2009]

[30] Netbeans IDE: http://www.netbeans.org/downloads/, [1. 4. 2009]

[31] Kadlec, R., Gemrot, J., Burkert, O., Bída, M., Havlíček, Brom, C. (2007): POGAMUT 2 - A platform for Fast Development of Virtual Agents' Behavior. In: Proceedings of CGAMES 07, La Rochelle, France.

[32] Pöppel, E. (1997): A hierarchical model of temporal perception. In: Trends in Cognitive Sciences. 1(2), 56-61.

[33] Huttenlocher, J., Hedges, L. Prohaska, V. (1988): Hierarchical Organization in Ordered Domains: Estimating the Dates of Events. Psychological Review, 95, 417-484.

[34] Pogamut GRID: http://artemis.ms.mff.cuni.cz/pogamut/, Sub-projects. [10. 4. 2009]

[35] Ho, W., Dautenhahn, K., Nehaniv, C. (2008): Computational Memory Architectures for Autobiographic Agents Interacting in a Complex Virtual Environment. Connection Science. In press.

Appendix A – Attachments

1. The image of connectionistic memory structure – the higher resolution with a complete view of the episodic memory structure for one day.
2. CD with:
   - Documentation:
     - Installation guide.
     - Programming documentation.
     - Javadoc – the automatically generated documentation of the code
   - Episodic bot – the source code and configuration files (lifestyles, behaviors).
   - Pogamut 2 – installation file
   - Questionnaire – the questionnaire (in Czech and in English) accompanied with the evaluation of filled out questionnaires.
   - The master thesis in the PDF format.
   - Videos:
     - The progress of the timezone learning.
     - The evolution of the time network during the learning phase.
     - The demonstration of an agent who is living in the environment.
   - README.TXT – step by step instructions for installation.

Appendix B – List of Context Nodes

The complete list of context nodes used during the simulation (see section 4.3.2 for more details).

- External context:
    - Light outside
    - Sunrise
    - Sunset
- Biological needs:
    - Tired
    - Dirty
    - Thirsty
    - Need a toilet
    - Hungry
- Internal context:
    - During the activity:
        - Working
        - Eating (breakfast, lunch, dinner)
        - Entertaining
        - Sleeping
    - After the activity:
        - After work
        - After food
        - After sport
        - After waking up
    - Other:
        - Stressed
        - Excited
        - After endorphins