DyBaNeM: Bayesian Episodic Memory Framework for Intelligent Virtual Agents

Rudolf Kadlec and Cyril Brom

Faculty of Mathematics and Physics
Charles University in Prague, Czech Republic
rudolf.kadlec@gmail.com
http://amis.mff.cuni.cz

Abstract. Episodic Memory (EM) abilities are important for many types of intelligent virtual agents (IVAs). However, the few IVA–EM systems implemented to date utilize indexed logs of events as the underlying memory representation, which makes it hard to model some crucial facets of human memory, including hierarchical organization of episodes, reconstructive memory retrieval, and encoding of episodes with respect to previously learnt schemata. Here, we present a new general framework for EM modeling, DyBaNeM, which capitalizes on bayesian representation and, consequently, enables modeling these (and other) features easily. By means of a proof-of-concept implementation, we demonstrate that our approach to EM modeling is promising, at least for domains of moderate complexity.

Keywords: Episodic Memory, Dynamic Bayes Networks, Dialog.

1 Introduction

Episodic memory (EM) [27] represents personal history of an entity. Episodic memories are related to particular places and moments, and are connected to subjective feelings and current goals. In the context of agent-based systems, episodic memory has been studied as a tool enhancing an agent's performance in simulated environments [23,26,13]. EM abilities can also increase *believability* of intelligent virtual agents (IVAs) in many applications, including role-playing games, serious games, interactive storytelling systems, and tutoring applications. Agents for a serious anti-bullying game were equipped by a simple EM for the purpose of debriefing [9]. Virtual guide Max uses EM to modify museum tours based on Max's previous experience [19,24]. Generic EM for virtual characters was proposed in [6]. Similarly EM can be used in cognitive robots [10]. At the same time, studies investigating how humans perceive IVAs with EM abilities started to be conducted. For instance, several results suggest that humans tend to prefer IVAs with imperfect memory [5,22]. Increased interest of users interacting with an agent presenting background stories possibly stored in EM in first person was shown in [3].[1]

[1] Some authors prefer the term autobiographic memory. In cognitive psychology, the meaning of the two terms differs but for the purpose of the present paper, we use them as synonyms. For more detailed review of EM agents, see [21].

R. Aylett et al. (Eds.): IVA 2013, LNAI 8108, pp. 15–28, 2013.

Here we present a new EM modelling framework, DyBaNeM. It has been created specifically with IVAs' needs in mind, that is, with the needs to model human EM in a more believable manner than in the past, for instance, to respect fallibility of human memory. The framework brings several key innovations. First, to our knowledge, none of the abovementioned systems enables *reconstruction* of hierarchical episode structure (i.e., episode — subepisode relationship) in cases where an observer IVA, let us call him Bob, equipped with the EM observes another IVA, say, Alice. Bob can see Alice's atomic actions but he has to reconstruct her high level goals if he wants to remember them. Only the model presented in [6] makes it possible for Bob to remember his own hierarchy of episodes but not that of Alice's. Second, our framework enables probabilistic reconstructive retrieval process that can result in reconstruction of events that have not happened at all, but they are sensible given the other stored memories. Third, the model remembers only some of the most salient events as opposed to most of the current models that use data structure resembling plain log of events such as [6,9]. While some models use emotions as a measure of saliency, e.g. [8,24], we use mathematically better rooted deviation from a statistical schema. Fourth, current models cannot express degree of belief in the recalled memory, they usually either return an episode or nothing. Our framework removes this restriction, recall in DyBaNeM results in multiple, more or less probable, possibilities. Fifth, the framework uses IVA's personal experience in encoding of the episodes. Two IVAs may remember the same episode in differently.

From the psychological point of view the framework is inspired by the Fuzzy-Trace Theory (FTT) [11]. FTT hypothesizes two parallel mechanisms that encode incoming information: *verbatim* and *gist*. While verbatim encodes the surface-form of the information in detail, gist encodes the meaning in a coarse-grained way [11], capitalizing on previously learnt *schemata* [2] of episodes and parts of episodes. From the computational point of view the framework uses Dynamic Bayesian Network (DBN) [18] as the underlying probabilistic model.

We illustrate possible use-cases of the framework on the following example. Imagine a MMORPG inhabited by hundreds or even thousands of non player characters (NPCs). Each NPC can be interviewed by a human player that may ask two basic types of questions: 1) "What were you doing yesterday?"; 2) "What was the player X doing yesterday?" The first question asks about the NPC's recollection of its own actions, the second asks about the NPC's memory for actions of a human controlled avatar (or a different NPC). It is clear that the NPC has to be equipped with an EM model to answer both of these questions. However, the second question also requires the model to be able to *interpret* the players' (NPCs') actions and infer his/her high level goal that are not directly observable in the environment. Our framework can do that. In addition, the model should be generic and applicable to different NPCs with minimal effort. In DyBaNeM, the model's parameters can be automatically adjusted for each type of NPC in the game by means of learning schemata of episodes by standard byesian methods.

Finally, while our IVAs are not equipped with a dialog generating system, our framework enables, in principle, the following features in the dialog between the player and a NPC :

1. The NPC can provide a high level summarization of an activity. For instance, when the player (P) asks: "What were you doing yesterday?", the NPC (N) equipped with our model can answer: "After getting up I went to work, in the afternoon, I visited my friends and then I returned home." instead of inadequately detailed "I got up, then I did my morning hygiene. I had a breakfast, I get dressed and ..."
2. The player can ask further clarifying questions. E.g., P: "How did you get to your friends?"; N: "I walked there."
3. The NPC can express degree of certainty for each recalled event. E.g., N: "Maybe I went there by a car. I'm not sure."
4. The NPC can make mistakes that are believable given the context. E.g., N: "I went to work by public transport." (Even though the NPC used a car.)
5. The memory weights interestingness of the events, thus it can highlight the most unusual memories. P: "What were you doing yesterday?"; N: "I saw a foreign army marching through the town, the rest of the day was as usual."
6. Personal experience can influence interpretation of others' activity. A worker NPC may think that the observed player is just visiting a museum, whereas a thief NPC may reveal that the player is preparing a theft.

We now detail these six above mentioned use cases and we sketch how DyBaNeM can be used to implement them. Then we describe DyBaNeM's core. In the end we present a prototype IVA simulated in a 3D environment equipped with DyBaNeM and experiments demonstrating applicability of the EM model.

2 How DyBaNeM Supports Rich Dialog with IVAs

First we will briefly summarize functions and processes of DyBaNeM EM, then we will detail how these can support the user's interaction with the IVA.

DyBaNeM uses *episodic schemata* learnt a priori to segment sequences of observations into meaningful *episodes* that can have hierarchical structure. Episodic schemata are parameters of the underlying probabilistic model used in several stages of DyBaNeM's working cycle. The probabilistic model is implemented by a DBN. First the schemata has to be specified by hand or learnt from labeled data. This happens "offline" before the model is deployed to the IVA. Later, in *encoding*, the model is presented with a sequence of observations to be stored. DyBaNeM deduces hierarchy of episodes represented by the observations and picks the most interesting facts called *mems* that will be stored (persisted in a long term store). These mems will become the internal representation of the observations, e.g. one day of the agent's activity. Interestingness is measured with the use of the episodic schemata. During *storage* some of the mems may be forgotten. In *retrieval* the mems, that is the exact memory of fragments of the past events, together with episodic schemata, are used to reconstruct the whole original episodes. This process may be imperfect, the more mems remain stored the better will be the match between the original and the recalled episodes. We may perceive this process as a lossy compression of the episodes.

Now we will show how the functions listed in Introduction are enabled by DyBaNeM. We will use an example dialog where a human user interviews IVA Bob about his observation of IVA Alice. Technical details of these six functions will be discussed later.

1. High level summarization is enabled by hierarchical nature of DyBaNeM. The recalled sequences contain not only atomic actions but also high level episodes that can be used as summarization of the sequence of actions. Thus if Bob has DyBaNeM with two levels of abstraction, the values reconstructed by the bayesian inference on the highest level can be used to provide a summary of the day. Fig. 1 shows an example of such situation.

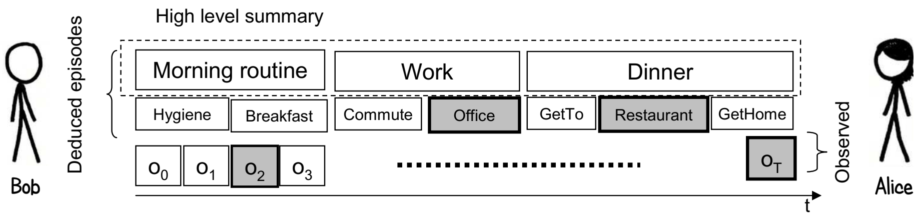

Fig. 1. Summarization example. Bob is a detective who monitors Alice. First, let Bob observe Alice's atomic actions $o_{0:T}$. Second, Bob can deduce the higher level episodes from this observation. Third, Bob encodes the whole day by the most salient events — these are the mems computed in the encoding algorithm. Mems are marked as the shaded boxes. When Bob is asked to summarize what Alice did yesterday he recalls the mems and reconstructs the rest with the use of the episodic schemata. In the end he responds by episodes in the highest level: "Morning routine, work and dinner."

2. Possibility of further clarifying questions is another useful feature of the hierarchical memory organization. When the user asks for details of an episode, Bob can reply by its sub-episodes as illustrated on Fig. 2a.

3. Expressing degree of certainty for recalled events is enabled by probabilistic nature of the framework. Each action/episode is represented by at least one random variable in the DBN. During reconstructive recall we obtain a probability mass function (PMF) for each variable that encodes probability of every action/episode at this point in time. When the probability of the most probable outcome dominates the other outcomes, we can say that the IVA is sure. However if there are two competing alternatives, the IVA can reflect this in the dialog. See Fig. 2b for an example.

4. Believable mistakes in recall can emerge as interplay of forgetting and reconstructive retrieval. When only a few mems remain stored then during the recall the forgotten events are reconstructed from the episodic schema. It can happen that the schema predicts an event that had not actually happen but it fits

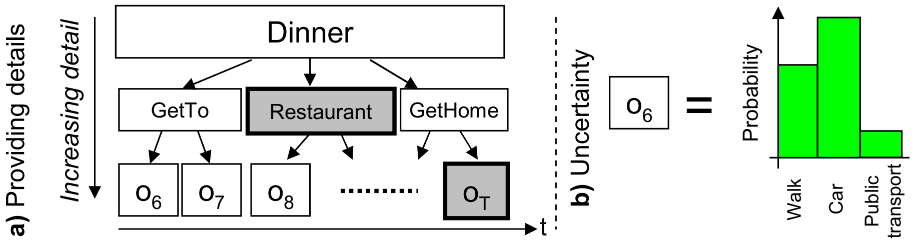

Fig. 2. **a)** When Bob is asked to say more about Alice's dinner, he will reply: "She left from work and went to the restaurant, she ate there and then she went back home." Shaded boxes represent mems, white represent reconstructed events. **b)** Further question can be: "How did she get to the restaurant?" which asks about recall of atomic actions represented by observations o_6 and o_7. In case of o_6 the associated PMF computed in recall assigns similar probability to both $Walk$ and Car. Thus Bob is not sure and he can reflect this in his answer: "She went by car or she just walked, I am not sure, sorry."

well to the way the episode usually unfolds. Different approach to this so called false memories phenomenon is discussed in our previous work [28]. Continuing in the example from Fig. 2, it may be the case that Alice used *Public transport* that day, but Bob does not remember this in a mem and his schema favors other options.

5. Measuring interestingness of events can be achieved by comparing the actual events to prediction from the schema. Imagine that 95 percents of days start by a sequence: *Get up, Brush teeth, Have a shower, Have a breakfast.* If the schema is expressive enough to capture this sequence, those events will become completely uninteresting. They are predictable, thus they do not distinguish one day from other. However meeting foreign soldiers marching through one's home town is much less probable. Thus it is the event that deserves more attention in the dialog than brushing teeth every morning again and again. The general notion is the lower the probability of an observed event given schemata the higher the surprise of observing it. We use Kullback-Leibler (KL) divergence [20] to measure how each observed event "diverges" from the prior prediction given solely by the schemata.

6. Influence of personal experience on interpretation of behavior of others is possible through a personalized set of episodic schemata for every IVA. Episodic schemata are parameters of the probabilistic model used in DyBaNeM, thus if Bob has a schema *theft preparation*, he may reveal that Alice was not visiting the gallery because of her interest in the new exhibition. Instead, he may conclude, she was examining the safety devices near the Da Vinci's painting. If the player asks IVA Cloe who does not have such schema, she would not know what Alice was planning.

3 DyBaNeM: Probabilistic EM Framework

We now describe DyBaNeM's computational core. We start with auxiliary definitions needed for description of the framework. Then we show how DBNs can be used for activity/episode recognition and how the episodic schemata are represented. We present the algorithms of encoding, storage and retrieval. Finally, we show how features 1-6 from Sec. 2 can be implemented in DyBaNeM. Additional details of DyBaNeM that are out of scope of this paper are available in [15,16]. DyBaNeM is available for download on its homepage[2].

Notation. Uppercase letters denote discrete random variables (e.g. X, Y) whereas lowercase letters denote their values (e.g. x, y). PMF of random variable X is denoted by $P(X)$. Domain of X is denoted as $D(X)$. Notation $X_{i:j}$ is a shorthand for sequence of variables $X_i, X_{i+1} \dots X_j$; analogically, $x_{i:j}$ is a sequence of values of those variables, the subscript denotes time. $\mathcal{M}$ will be a probabilistic model and $\mathcal{V}$ is a set of all random variables in the model.

Now we formalize representation of episodes and world state assumed by DyBaNeM.

Episode is a sequence (possibly of length 1) of observations or more fine-grained episodes (sub-episodes) that has a clear beginning and an end. Note that episodes may be hierarchically organized.

Episodic schema is a general pattern specifying how instances of episodes of the same class look like. For instance, an episodic schema (cf. the notion of script or memory organization packet [25]) might require every episode derivable from this schema to start by event a, then go either to event b or c and end by d.

Episodic trace $\epsilon_t^{0:n}$ is a tuple $\langle e_t^0, e_t^1 \dots e_t^n \rangle$ representing a hierarchy of episodes at time t; e_t^0 is the currently active lowest level episode, e_t^1 is its direct parent episode and e_t^n is the root episode in the hierarchy of depth n. Example of an episodic trace can be $\epsilon_0^{0:n} = \langle WALK, COMMUTE \rangle$ and $\epsilon_1^{0:n} = \langle GO\_BY\_BUS, COMMUTE \rangle$. The notation of episodic trace reflects hierarchical nature of agent behavior.

Our framework uses probabilistic representation, hence even if there is only one objectively valid episodic trace at each time step, input of the EM model will be a probability distribution. Let E_t^i denotes a random variable representing a belief about an episode on level i at time t. While the true value of E_t^i is, say, e_t^i, the PMF enables us to cope with possible uncertainty in perception and recall.

Probabilistic episodic trace $E_t^{0:n}$ is a tuple of random variables $\langle E_t^0 \dots E_t^n \rangle$ representing an agent's belief about what happened at time t. Analogically $E_{0:t}^{0:n}$ denotes probabilistic episodic trace over multiple time steps. The following data structure represents an agent's true perception of the environment state. Let ρ_t denotes **observable environmental properties** at time t.

For instance, ρ can hold atomic actions executed by an observed agent (and possibly other things too), e.g. $\rho_0 = STAND\_STILL$, $\rho_1 = GET\_TO\_BUS$. Analogically to $E_t^{0:n}$ and $\epsilon_t^{0:n}$, O_t is a random variable representing belief about observation ρ_t.

[2] DyBaNeM's homepage: `https://code.google.com/p/dybanem/`

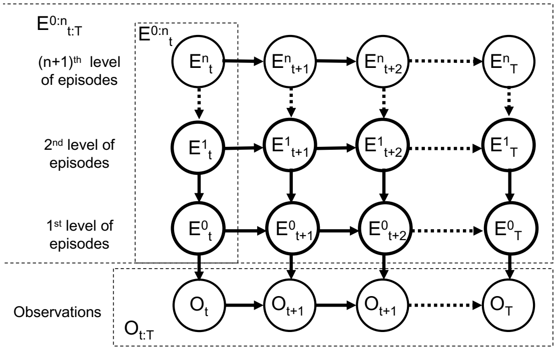

Fig. 3. An example of a DBN's structure called CHMM [4] together with our notation

Fig. 3 shows how these definitions translate to an example DBN structure used in this paper called Cascading Hidden Markov Model (CHMM) [4].

Surprise. In encoding, the framework works with quantity measuring difference between the expected real state of a random variable and its expected state given the remembered facts. We call this quantity surprise. In Bayesian framework surprise can be defined as "difference" between prior and posterior probability distributions. We adopt approach of [14] who propose to use KL divergence [20] to measure surprise. **KL divergence** of two PMFs $P(X)$ and $P(Y)$, where $D(X) = D(Y)$ is defined as:

$$KL(P(X) \to P(Y)) = \sum_{x \in D(X)} P(X = x) ln \frac{P(X=x)}{P(Y=x)}.$$

We use notation with $\to$ to stress directionality of KL divergence; note that it is not symmetrical.

Learning Schemata. Episodic schemata are represented by parameters $\hat{\theta}$ of a DBN. Expressiveness of schemata depends on the structure of a model at hand. We suppose that the DBN's topology is fixed. Thus learning schemata will reduce to well known parameter learning methods. Topologies without unobserved nodes including CHMM, are learnt by counting the sufficient statistics [18]. In our case examples of episodes that we want to use for schemata learning will be denoted by $\mathcal{D} = \{d_1, d_2 \dots d_n\}$ where each d_i is one day of an agent's life; d_i itself is a sequence of examples c_t, that is, $d_i = \{c^i_0, c^i_1 \dots c^i_{T_i}\}$. Each c^i_t is a tuple $\langle \epsilon^{0:n}_t, \rho_t \rangle$, it contains an episodic trace and observable state of the environment.

DBN Architecture. For computing probabilities, our framework makes it possible to use various DBN architectures. In this paper we use a CHMM [4] architecture which is a hierarchical extensions of a well known Midden Markov Model (HMM) (see Fig. 3). However more complex models, better suited for activity representation, like Abstract Hidden Markov Memory Model (AHMEM) [7], can be used. Downside of the AHMEM is its higher computational cost, thus we use simpler, but still sufficient CHMM. Experiments comparing AHMEM with CHMM in Dy-

BaNeM are presented in [15]. The schemata are represented by parameter $\hat{\theta}$, that is, by all conditional probability mass functions (CPMFs) of the DBN's nodes. Expressiveness of the schemata depends on the structure of DBN. In CHMM episodic schemata encode probability of an episode given previous episode on the same level in the hierarchy and also given its parent episode ($P(E_t^i|E_{t-1}^i, E_t^{i+1})$).

Encoding. The encoding algorithm computes a list of *mems* on the basis of the agent's perception, $Per_{0:T}$, of the situation to be remembered. $Per_{0:T}$ is a tuple of PMFs such that $Per_{0:T} = \{f_X : X \in \mathit{Observable}\}$, where f_X is PMF for each variable X of interest. In a case when Bob is going to encode Alice's activity (see Fig. 1), $\mathit{Observable} = O_{0:T}$. Alice's ϵ^{Alice} is hidden to Bob, nevertheless Bob perceives Alice's atomic actions that are contained in ρ^{Alice}.

Algorithm 1 is a skeleton of the encoding procedure. The input of the algorithm is $Per_{0:T}$, where the time window $0 : T$ is arbitrary. In our work we use time window of one day. The output is a list of mems encoding this interval.

Algorithm 1 General schema of encoding algorithm

Require: $Per_{0:T}$ — PMFs representing the agent's perception of the situation (i.e. smoothed observations)
Require: $\mathcal{M}$ — probabilistic model representing learned schemata
procedure ENCODING($Per_{0:T}, \mathcal{M}$)
    $mems \leftarrow empty$ ▷ List of mems is empty
    while $EncodingIsNotGoodEnough$ **do**
        $X \leftarrow GetMem(\mathcal{M}, Per_{0:T}, mems)$
        $x_{max} \leftarrow MLO_{P_{\mathcal{M}}}(X|mems)$
        $mems.add(X = x_{max})$
    end while
    return $mems$
end procedure

The algorithm terminates once the $EncodingIsNotGoodEnough$ function is false. We use stopping criterion $|mems| < K$ because this models limited memory for each day. In each cycle, the $GetMem$ function returns the variable X_t^i that will be remembered. The MLO function (most likely outcome) is defined as: $MLO_{P_{\mathcal{M}}}(X|evidence) \equiv \arg\max_{x \in D(X)} P(X = x|evidence)$. We get the most probable value for X and add this assignment to the list of mems. The $GetMem$ function is implemented in the following way. The idea is to look for a variable whose observed PMF and PMF in the constructed memory differs the most. This variable has the highest surprise and hence it should be useful to remember it. This memory creation strategy is retrospective, it assumes that the agent has all observations in a short term memory, and, e.g., at the end of the day, he retrospectively encodes the whole experience. The strategy memorizes the value of variable X such that: $X \leftarrow \arg\max_{Y \in VOI} KL\left(P_{\mathcal{M}}(Y|Per_{0:T}) \to P_{\mathcal{M}}(Y|mems)\right)$, where $P(Y|Per_{0:T}) \equiv P(Y|X = f_X : f_X \in Per_{0:T})$; we condition the probability on all observations. $VOI \subseteq \mathcal{V}$ is a set of random *variables of interest* whose value can be remembered by the model. In our implementation $VOI = E^{0:n}_{0:T} \cup O_{0:T}$. Note that we remember only the most probable value, including the time index.

Storage and Forgetting. During storage, the mems can undergo optional time decayed forgetting. The following equation shows relation between age t of the mem m, its initial strength S and its retention R: $R(m) = e^{-\frac{t}{S}}$ [1]. The initial strength S of mem m can be derived from the value of KL divergence computed in $GetMem$. Once $R(m)$ decreases under the threshold β_{forget}, m will be deleted from the list of mems and will not contribute to recall any more.

Retrieval. Retrieval is a simple process of combining the schemata with mems. We obtain the list of mems for search cue k, which can be, e.g., a given day. Then we use assignments in the mems list as an evidence for the probabilistic model. The resulting PMFs for all variables of interest are returned as a reconstructed memory for the cue k. Retrieval can be formalized as computing $P_{\mathcal{M}}(Y|mems)$ for each $Y \in VOI$.

Now we show how DyBaNeM's dialog supporting features are implemented.

1. High level summarization and 2. Further clarifying questions are possible because of the hierarchical structure of DBN used in both encoding and retrieval. Values of variables $E^n_{0:T}$ (see Fig. 3) can be used for summarization. If the user asks for details of time interval $\langle t_1, t_2 \rangle$, values of $E^{n-1}_{t_1:t_2}$ can be used to construct the answer (or $O_{t_1:t_2}$ when $n = 0$).

3. Expressing degree of certainty of recall is implemented by computing entropy of random variable corresponding to the action/episode. Entropy $H(X)$ of random variable X is defined as $H(X) = -\sum_{x \in D(X)} P(x) log_2 P(x)$. The higher the entropy is, the more uniform the PMF over X is. Thus there is more uncertainty since all outcomes of X seem similarly probable. On the other hand when entropy is close to zero there is only a little uncertainty about X's value.

4. Believable mistakes in recall result from forgetting and the inference process in retrieval. It can happen that there was an action a at time t' and during storage the mem for t' was forgotten. Later in retrieval, that is when computing PMF $f_{t'} = P_{\mathcal{M}}(O_{t'}|mems)$, the value had to be deduced from remembered mems and the probabilistic model $\mathcal{M}$ that includes the episodic schemata. If action b is more probable under this assumption ($P_{\mathcal{M}}(O_{t'} = b|mems) > P_{\mathcal{M}}(O_{t'} = a|mems)$), b will be recalled instead of a. There is no specific process for this feature, it is DyBaNeM's emergent property.

5. Interestingness of events is measured by KL divergence in the same way it is done by the encoding algorithm. The more different is a PMF predicted by the schemata from the recalled PMF the higher is the value of KL divergence. The first mem picked by the encoding algorithm is the one that deviates most from the prediction from schema. Subsequent mems contain less and less information. Thus if an IVA wants to communicate the interesting events first it can start with the first mem followed by the second and so on. If both the IVA and the human player have the same episodic schemata they will be able to reconstruct the same episodes. This is similar to function of lossy compression algorithms. DyBaNeM gets observed episode on the input, then it transforms the episode into a list of mems that is shorter than the original episode. With the use of the episodic schemata the mems can be used to reconstruct the episode. However some details of the episode might be changed due to forgetting and imperfect schemata. The difficulty in interpreting DBN*EM* as a compression algorithm is that not only mems but also the episodic schemata θ has to be stored (or transmitted). Since storage of θ requires far more space than storage of one mem this approach is feasible only if large number of episodes will have to be stored. On the other hand θ does not have to be transmitted if both parties, i.e. Bob and Alice, have already the same schemata.

6. Influence of personal experience follows from a different set of episodic schemata of each IVA. When Bob's schemata are trained on a different corpus of examples than Cloe's, the resulting probabilistic models will be also different. Thus inferences from these models may give different mems.

4 Prototype DyBaNeM Connection to an IVA

To demonstrate DyBaNeM's applicability to the domain of IVAs we connected it to an IVA whose behavior resembles a background character from a MMORPG. We show 1) that DyBaNeM can learn the schemata, store and recall one day of IVA's activity and 2) that it can support the dialog enhancing features discussed in Sec. 2. We also show 3) that the method has reasonable computational time requirements given domains of moderate complexity, even though the problem of exact inference in Bayesian Network is exponential in the network's treewidth.

Activity Dataset. As input for the EM model, we generated hierarchical activity dataset simulating 23 "days" of an IVA's life. The IVA was controlled by hierarchical decision making system (DMS) based on AND-OR trees formalism. An AND-OR tree describes decomposition of IVA behavior into goals and subgoals with possible alternatives of accomplishing each goal. The IVA's nondeterministic scheduling algorithm together with nondeterminism originating from the 3D simulation result in irregularities of stream of actions produced for each day. In our previous work we compared various statistical properties of the generated behavior to datasets of human behavior with reasonable match [17]. Our IVA is connected to a 3D virtual environment of Unreal Tournament 2004[3]. The agent was implemented in Java and the Pogamut platform [12] was used as a middleware for interfacing the IVA with the environment.

Every simulated day has a similar structure, the IVA gets up at home, he brushes teeth, washes face, goes to the toilet; then he usually goes to work; in the evening he may go to a theater or to a pub. He may also do shopping, clean the house and other activities resembling a normal life. In total the simulation contains 37 different types of atomic actions and 19 types of first level episodes. The generated stream of actions contains more levels of episodes but for this evaluation we use only the first level of episodes which is sufficient for

[3] Epic Games, 2004, [7.4.2013], `http://web.archive.org/web/20060615184746/http://www.unrealtournament2003.com/`

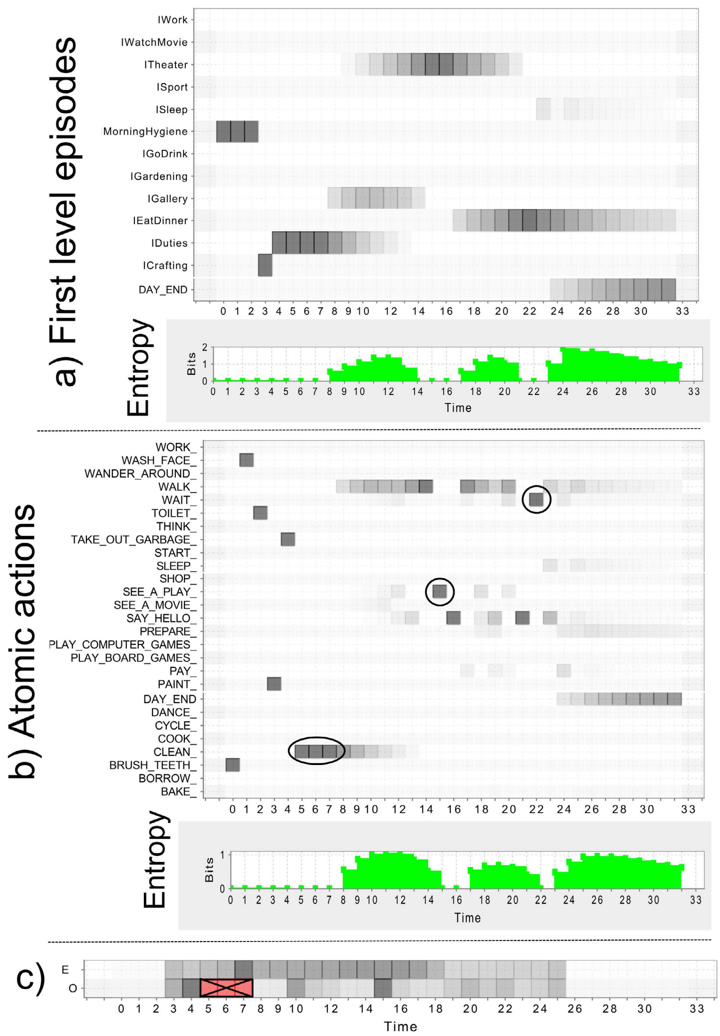

Fig. 4. Recall of the stored day when three mems are used for reconstruction. The mems are atomic actions (subfig. b), in ellipses). Subfig. **a)** shows selected high level episodes recalled for the day of interest. Level of gray indicates probability of each atomic action/episode at that time step. The darker the color is the more probable the action/episode is. This corresponds to the feature **1**. Entropy shows how certain the IVA is about his memory for those events (feature **3**). The more alternative memories are there the higher the entropy is. Subfig. **b)** shows probability and entropy of selected atomic actions. This is the second level of hierarchy that allows for clarifying questions (feature **2**). Subfig. **c)** shows KL divergence of all episodes (first line) and actions (second line) in the day compared to the prior episodic schema (feature **5**). The most interesting actions are marked by a cross (gray coding as in the case of probability). Most of the interesting actions/events become mems. Feature **4** is demonstrated by "fuzzy" transition around time 10: the model is not sure when exactly happened the switch from household duties to a gallery visit.

demonstrating all the features. There are different plans for working days and for weekends, that increases variability of the the IVA's episodic schemata. Not all days contained the same number of the atomic actions, the longest one has 33 actions. To make all days equal in size we added a sufficient number of padding actions *DAY_END* to the end of each day. Details of IVA's DMS are provided in [17] (sec 3.2).

Method. Twenty-two days were used for learning episodic schemata. The underlying probabilistic model CHMM was learned by counting the sufficient statistics [18]. The 23rd day was stored in DyBaNeM to demonstrate its abilities. The model was presented with the atomic actions from that day and it had to deduce high level episodes and store the observations. The result of the encoding process were 3 mems. This way we model Bob's EM for Alice's activity. For belief propagation in DyBaNeM's DBNs, SMILE[4] reasoning engine for graphical probabilistic models was used.

Results. When using only one mem to reconstruct the whole day, 52% of atomic actions were correctly recalled, with two mems it was 64%, and with three mems 73%. This means that when all three mems were used the most probable action in 73% of time steps matched the real action previously encoded. Recall of the stored day when all three mems were used is shown on Fig. 4. Learning the episodic schemata took 2 seconds, computing the first 3 mems for the stored day took 1.3 second on one core of P8600 2.4GHz, 1.5GB RAM.

Discussion. The evaluation indicates that computational cost is reasonable. Learning the schemata is done only once off-line and time necessary for encoding (1.3s) is also acceptable, though the domain is of moderate complexity only. Fig. 4 illustrates all the features 1-5. To demonstrate feature 6 we would need a second IVA with a different lifestyle that could be used to learn another set of episodic schemata. This extension is trivial but we omit it for space restrictions. The 73% recall accuracy is a reasonable starting point: it can be increased with more mems stored, and a user study, a future work, will indicate what accuracy is most welcomed by users.

Extending the IVA with ByDaNeM is a simple task that requires a developer only to: a) get logs of IVA's behavior that were used for episodic schemata learning, b) decide when to store episodes (e.g. at the end of the day) and c) decide when to recall the episode. Thus no advanced knowledge of DyBaNeM's internals is needed by the IVA developer.

5 Conclusion

We have demonstrated that bayesian approach to IVA—EM modeling, exemplified on our new DyBaNeM framework, is promising and it can be considered by developers of IVAs with EM abilities as a possible development method. To investigate scalability of this approach, we are presently experimenting also with

[4] SMILE was developed by the Decision Systems Laboratory of the University of Pittsburgh and is available at `http://genie.sis.pitt.edu/`.

larger domains, including human corpora, and different underlying DB representations. Our most recent evaluation data actually must be omitted for space limitations. A key future step is a user evaluation of the framework.

Acknowledgments. This research was partially supported by SVV project number 267 314 and by grant GACR P103/10/1287. We thank to xkcd.org for drawings of Alice and Bob.

References

1. Anderson, J.: A Spreading Activation Theory of Memory. Journal of Verbal Learning and Verbal Behavior 22, 261–295 (1983)
2. Bartlett, F.: Remembering: A Study in Experimental and Social Psychology. Cambridge University Press, Cambridge (1932)
3. Bickmore, T., Schulman, D., Yin, L.: Engagement vs. Deceit: Virtual Humans with Human Autobiographies. In: Ruttkay, Z., Kipp, M., Nijholt, A., Vilhjálmsson, H.H. (eds.) IVA 2009. LNCS, vol. 5773, pp. 6–19. Springer, Heidelberg (2009)
4. Blaylock, N., Allen, J.: Fast Hierarchical Goal Schema Recognition. In: Proceedings of the National Conference on Artificial Intelligence (AAAI 2006), pp. 796–801 (2006)
5. Brom, C., Burkert, O., Kadlec, R.: Timing in Episodic Memory for Virtual Characters. In: Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games, pp. 305–312 (2010)
6. Brom, C., Pešková, K., Lukavský, J.: What Does Your Actor Remember? Towards Characters with a Full Episodic Memory. In: Cavazza, M., Donikian, S. (eds.) ICVS 2007. LNCS, vol. 4871, pp. 89–101. Springer, Heidelberg (2007)
7. Bui, H.: A general model for online probabilistic plan recognition. In: International Joint Conference on Artificial Intelligence, pp. 1309–1318 (2003)
8. Deutsch, T., Gruber, A., Lang, R., Velik, R.: Episodic memory for autonomous agents. In: Human System Interactions (HSI 2008), pp. 621–626 (2008)
9. Dias, J., Ho, W.C., Vogt, T., Beeckman, N., Paiva, A., André, E.: I Know What I Did Last Summer: Autobiographic Memory in Synthetic Characters. In: Paiva, A., Prada, R., Picard, R.W. (eds.) ACII 2007. LNCS, vol. 4738, pp. 606–617. Springer, Heidelberg (2007)
10. Dodd, W., Gutierrez, R.: The Role of Episodic Memory and Emotion in a Cognitive Robot. In: IEEE International Workshop on Robot and Human Interactive Communication, ROMAN 2005, pp. 692–697 (2005)
11. Gallo, D.: Associative illusions of memory: False memory research in DRM and related tasks. Psychology Press (2006)
12. Gemrot, J., Kadlec, R., Bída, M., Burkert, O., Píbil, R., Havlíček, J., Zemčák, L., Šimlovič, J., Vansa, R., Štolba, M., Plch, T., Brom, C.: Pogamut 3 Can Assist Developers in Building AI (Not Only) for Their Videogame Agents. In: Dignum, F., Bradshaw, J., Silverman, B., van Doesburg, W. (eds.) Agents for Games and Simulations. LNCS, vol. 5920, pp. 1–15. Springer, Heidelberg (2009)
13. Ho, W.C., Dautenhahn, K., Nehaniv, C.: Computational Memory Architectures for Autobiographic Agents Interacting in a Complex Virtual Environment: A Working Model. Connection Science 20(1), 21–65 (2008)

14. Itti, L., Baldi, P.: Bayesian surprise attracts human attention. Vision Research 49(10), 1295–1306 (2009)
15. Kadlec, R., Brom, C.: DyBaNeM: Bayesian Framework for Episodic Memory Modelling. In: The 12th International Conference on Cognitive Modelling, ICCM (in press, 2013)
16. Kadlec, R., Brom, C.: Unifying episodic memory models for artificial agents with activity recognition problem and compression algorithms: review of recent work and prospects. Tech. rep. (2013)
17. Kadlec, R., Čermák, M., Behan, Z., Brom, C.: Generating Corpora of Activities of Daily Living and Towards Measuring the Corpora's Complexity. In: Dignum, F., Brom, C., Hindriks, K., Beer, M., Richards, D. (eds.) CAVE 2012. LNCS, vol. 7764, pp. 149–166. Springer, Heidelberg (2013)
18. Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. The MIT Press (2009)
19. Kopp, S., Gesellensetter, L., Krämer, N.C., Wachsmuth, I.: A Conversational Agent as Museum Guide – Design and Evaluation of a Real-World Application. In: Panayiotopoulos, T., Gratch, J., Aylett, R.S., Ballin, D., Olivier, P., Rist, T. (eds.) IVA 2005. LNCS (LNAI), vol. 3661, pp. 329–343. Springer, Heidelberg (2005)
20. Kullback, S.: Statistics and information theory (1959)
21. Lim, M.Y.: Memory models for intelligent social companions. In: Zacarias, M., de Oliveira, J.V. (eds.) Human-Computer Interaction. SCI, vol. 396, pp. 241–262. Springer, Heidelberg (2012)
22. Lim, M., Aylett, R., Enz, S., Ho, W.: Forgetting Through Generalisation — A Companion with Selective Memory. In: AAMAS 2011, pp. 1119–1120 (2011)
23. Nuxoll, A.M., Laird, J.E.: Enhancing intelligent agents with episodic memory. Cognitive Systems Research 17-18, 34–48 (2012)
24. Rabe, F., Wachsmuth, I.: Cognitively Motivated Episodic Memory for a Virtual Guide. In: ICAART, pp. 524–527 (2012)
25. Schank, R.C.: Dynamic memory revisited. Cambridge University Press (1999)
26. Subagdja, B., Wang, W., Tan, A.H., Tan, Y.S., Teow, L.N.: Memory Formation, Consolidation, and Forgetting in Learning Agents. In: AAMAS 2012, pp. 1007–1014 (2012)
27. Tulving, E.: Elements of Episodic Memory. Clarendon Press, Oxford (1983)
28. Čermák, M., Kadlec, R., Brom, C.: Towards modeling false memory using virtual characters: a position paper. In: Symposium on Human Memory for Artificial Agents, AISB, pp. 25–29 (2011)