

# Episodic Memory for Companion Dialogue

**Gregor Sieber**

OFAI

Vienna, Austria

gregor.sieber@ofai.at

**Brigitte Krenn**

OFAI

Vienna, Austria

brigitte.krenn@ofai.at

## Abstract

We present an episodic memory component for enhancing the dialogue of artificial companions with the capability to refer to, take up and comment on past interactions with the user, and to take into account in the dialogue long-term user preferences and interests. The proposed episodic memory is based on RDF representations of the agent's experiences and is linked to the agent's semantic memory containing the agent's knowledge base of ontological data and information about the user's interests.

## 1 Introduction

Recently, research on artificial companions has come more and more in focus. They are artificial agents (virtual or robotic) that are intended to support the human user in aspects of everyday life. They may range from virtual agents that assist their users in accessing information from the Internet in accordance with the users' interests, preferences and needs (Skowron et al., 2008), up to assistive robots in home environments that support elderly in mastering their life at home (Graf et al., 2009). In the long run when developing companions, the goal is to model and implement artificial "caring developing helpers" (Sloman, 2007) that learn and develop over time to be of long-term benefit for the user.

In order to come closer to the vision of artificial companions a number of research issues need to be addressed such as: action-perception and learning capabilities suitable to function with imperfect sensors in dynamically changing environments which can only be partially modelled; the development of affect sensing capabilities that extend over the detection of basic emotions such as joy, anger, fear, disgust etc. (Ekman, 1992); user

models that account for and adapt to the users' interests, preferences, affective states, needs and handicaps; approaches to multimodal dialogue that allow the agent's mental models and memories to be connected to its expressive behaviour (Castellano et al., 2008), and where natural language dialogue is semantically grounded (Benyon and Mival, 2008). Companions need to be aware of their own history and past interactions with their individual users, so that the single user can believe that her/his companion knows "what it is talking about". This is particularly important for creating acceptable long-term interactions.

To account for this kind of requirements, we propose a communication component for companions where autobiographic episodic memory, semantic memory and dialogue are closely connected. In our approach, input analysis is performed using information extraction techniques, that yield RDF triples describing the content of a user utterance in terms of the knowledge base (semantic memory) of the companion, and an utterance class describing the type of message (greeting, question, agreement, rejection, etc.). Short term memory holds the current user utterance and a set of pointers to currently important and thus activated parts of the companion's knowledge. We distinguish two parts of the long term memory: *Semantic memory* is composed of a knowledge base containing ontological data and a user model encoding e.g. elements of the ontology which the user is especially interested in. *Episodic memory* is based on RDF representations of the agent's experiences. It contains utterances of the user and the companion, and representations of the companion's actions and their evaluation (for the cases where it is known). The dialogue manager consists of a set of parallel, independent components for the different queries on the episodic memory described below and answer retrieval from the knowledge base. Which component is finally used

is decided by a scoring mechanism in connection with a rule set.

In the remainder of this contribution, we will concentrate on the interplay between episodic memory and dialogue. In particular, we describe how the episodic memory is represented (sec. 2), how episodes are retrieved (sec. 3), and how natural language output is generated from memory content (sec. 4).

## 2 Episodic Memory Representation

An episodic memory component for companion dialogue needs to provide adequate knowledge representation in connection with the cognitive model and the tasks of the agent. RDF-based<sup>1</sup> data stores are widely used for representing domain knowledge as well as common sense knowledge (e.g. the Open Mind Common Sense Database<sup>2</sup>, or ConceptNet<sup>3</sup>). Accordingly, we have developed an episodic memory component for artificial companions that stores episodes as RDF graphs. Since both memory, domain and common sense knowledge bases are composed of RDF triples, they are interoperable and can be easily extended. We use a Sesame<sup>4</sup> repository for hosting the data stores.

Episode encoding is automatic, since all user input and its analysis is immediately transferred from short-term memory to episodic memory. Thus the agent is able to recall the same data from an episode that was available at the time of the experience.

For episode retrieval, a similarity matching algorithm is required that can find memories based on similarity of the individuals and relations involved. Thus, our retrieval mechanism neither treats the RDF data as symbols in a similarity vector – such as for a nearest-neighbour search –, nor as a graph matching problem, which often is too slow for retrieval. Both of these approaches do not take advantage of the RDF encoding of the data, and as a consequence do not allow class or superclass information of individuals to be used for matching.

Our approach is to query the RDF repositories using a query language such as SeRQL and SPARQL. While these query languages do not allow a direct search for a similar graph, a set of

queries can be generated from a target episode making use of the full range of features of RDF and the query language. The episode most similar to the input episode is then selected from the result set by applying a heuristic.

### 2.1 Episodes

In our system, there are several types of episodes which share a set of basic parameters, each representing the different events and actions in the world of the agent.

The different sub-types of episodes are RDF subclasses of the basic episode concept and contain specialised parameters applicable to the type of action.

Basic properties stored with each episode are: a) *creation time* of the episode and b) an *episode ID* property which is used to trace back or forward through the episodes in (reverse) order of creation, to find the outcome and evaluation following an episode retrieved from memory. This is necessary, because triples in RDF are stored as graphs and not database entries like in a relational database which could easily be ordered by a primary key.

**Action episodes** are a subclass of episodes that represent the actions the agent is capable of. These are:

*Answer from domain knowledge* the agent maps the user's question to a SeRQL query and evaluates the query against its domain knowledge base. *Find similar interactions* represents deliberate remembering, i.e. actively searching for similar situations.

*Pattern search* allows the agent to check for a set of patterns in the behaviour of the user and its episodic memory which can be exploited for dialogue.

*Retrieve context* is employed by the agent when no other actions can be applied because parts of the utterance are missing. The companion then searches its memories to retrieve relevant context of the dialogue.

*Send message to the user*, which can either communicate the results of a query, memories of the agent, statements based on results from pattern search, or details about the situation of the agent, which includes reporting errors.

**Input episodes** store textual user input. They contain the analysis of the user input which is an RDF description of the entities, classes and prop-

<sup>1</sup><http://www.w3.org/RDF/>

<sup>2</sup><http://commons.media.mit.edu/en/>

<sup>3</sup><http://conceptnet.media.mit.edu/>

<sup>4</sup><http://www.openrdf.org/>

erties of the domain ontology contained in the utterance. For example, the question “When was Charlie Parker born?” is classified as utterance class WH-Question, and its analysis is an RDF triple with the ontology individual of class *Artist* representing Charlie Parker, the property *birth-Date*, and a variable as the object since it is this value the user wants to know.

**Evaluation episodes** can be either positive or negative. They are crucial for the agent to be able to learn from its past actions. If an evaluation is available, the agent can decide based on its memories whether a past solution should be repeated or not. Not all episodes have an evaluation. Evaluation values can either come from direct user feedback or internal feedback such as empty query results or failure to retrieve a query result.

In order to be able to find the right associations and memories, the agent also needs to have an internal notion of *relative time* that can be related to interactions with the user. As noted e.g. by Brom and Lukavský (2009) humans commonly do not use exact times, but instead refer to fuzzy categories. Thus, our (application specific) time model of the companion allows to differentiate between four coarse times of day – morning, noon, afternoon, evening. For events that are further in the past, the model contains the categories of: today, yesterday, this week, this month, this year, last year.

## 2.2 Episode Dynamics

Due to available computing hardware and scalable triple stores, the episodic memory component is technically able to store a large amount of memories. But when the episode base grows too big, it becomes increasingly difficult to retrieve episodes within an acceptable time limit due to the growing number of search and comparison operations required. Thus the companion needs a mechanism of reducing the number of episodes in the memory. Generally, there are two approaches to this: episode blending and forgetting.

Episode blending refers to a mechanism that groups similar experiences into one episode. Less important parameters of the memories are lost, and the similarities strengthened. This would mean the agent can remember what happened, and that it happened more than once, but the exact situations are lost. Episode blending is an interesting aspect of episodic memory that will be pursued in our fu-

ture work.

Forgetting refers to the deletion of episodes. Ideally, the episodes with least utility to the companion should be deleted. Nuxoll (2007) provides a list of possible approaches regarding forgetting: 1) remove the oldest memory first, 2) remove the least activated memory, 3) remove the most redundant memory, 4) memory decay.

Approach 1) does not take the importance of episodes into account and may result in losing important information. Approaches 2) and 4) both depend on assigning activation values to episodes, and delete those with the least activation. The idea of 3) is to locate two memories that are very similar to each other and remove one of them.

Our initial strategy is to assign a time-stamp of last retrieval to each episode, since we currently do not use activation values. Episode removal can then be regularly performed by issuing a SeRQL delete statement for all episodes whose retrieval date is older than a certain time, depending on the growth rate of the memory.

Note that the removal process described above still bears the risk of losing important memories of situations that are very rarely encountered. For our dialogue application scenario, this risk might not seem too critical, yet it might be e.g. for an agent in an artificial life environment where seldomly occurring enemies need to be recognised. A possible remedy would be the connection of episodic memory with a model of emotion. This would allow the emotional intensity of a situation to be a factor in episode retrieval and deletion.

## 3 Retrieval of Episodic Memories

One of the important aspects of any episodic memory component is to retrieve the right memories.

Since our episodic memory is realised using RDF, a set of SeRQL queries is used for episode retrieval. Queries are processed in parallel. The construction of these queries depends on the type of episode represented by the input situation.

The following section describes our model for deliberate retrieval for dialogue situations. This means that the companion actively chooses to search its memory for episodes of relevance.

The current situation is characterised by a set of features, expressed in RDF data, that are extracted from short term memory: 1) the description of the user utterance in terms of domain data, 2) the current time, 3) a list of entities in the user utterance

that are among the user's preferred entities, if any.

A query is issued representing the input situation. This means, we search the memory to see if the exact same situation has been encountered previously. Alternatively, queries using combinations and subsets of the instance set and the set of relations present in the user utterance are issued. For instance, given a popular music gossiping scenario, if the user asks a question about Michael Jackson, Janet Jackson, and Tina Turner, the agent searches its memory for previous episodes involving the named artists and relations or subsets of those, in order to connect to and take up previous discussions. Moreover, the structure of the domain data is used for generating a query containing the classes of the individuals in the utterance. For example, an agent that has talked about the birthday of any guitar player before, could relate a user question about the birthday of Joe Satriani to the previous experience by knowing that he is a guitar player too, and use this knowledge in the ongoing dialogue.

Queries related to classes can be iterated by following up the superclass hierarchy until a result is found. The iteration stops either when there is no further superclass, or when the property under discussion is not a property of the superclass anymore. For example, talking about the birthday of an *Artist*, the companion looks for episodes about birthdays involving its superclass *Person*, but not episodes with its superclass *Entity*, since the class *Entity* has no birthday property.

The most similar episode is selected from the result set by a heuristic which ranks those episodes higher that resemble the input episode more closely, so for example an episode that contains the same entities and the same properties as the input episode is ranked higher than an episode that contains a matching entity with a different property, and so on.

These content-driven retrieval strategies can be used to support the selection of the next dialogue move, taking into account available evaluations of similar past episodes. Additionally to the content-driven mechanism of remembering, the companion can also search its memories for recency- and preference-driven patterns that can be used for dialogue, such as the following examples. In contrast to the mechanisms mentioned above, these operations are automatically performed without requiring the agent's initiative.

*Has the same question been discussed recently, or ever before?* The companion can make a comment to the user about this – either noting as trivia that the question has been asked a year ago, or reacting annoyed if the user asks the same question for the fifth time within ten minutes.

*Is there a property in a user utterance that is among the user's interests? Has this property been asked for in the last 15 interactions?* For example, the user is very interested in the birth places of artists. The companion can use this information in the following ways: a) for the next artist under discussion, automatically provide the birth place to the user; b) the companion can comment on the fact that the property is part of the user interests; c) the companion can ask the user whether she would like to know the birthplace of a randomly selected artist from her preference list (the companion would select an artist whose birth place has not been inquired in the recent past, by checking against its memories).

*In the last 15 interactions that related to a certain property, is there a strong tendency (currently, more than 66%) towards one specific value of that property?* The companion can then search for similar cases among the data, and check whether there is another artist – maybe even among the user's preferred artists – that shares this birth place.

Additionally, this type of information is stored in the user model and leads to automatic retrieval of episodes where appropriate. Continuing the example of the birth place from above: a day after being asked about artists born in New York, the companion might notice while talking about the albums recorded by Billy Joel that he was also born in New York, and communicate it to the user.

Building upon the user preferences stored in the user model, the remembering process additionally contains queries related to the most prevalent preferences of the user model. This is similar to finding strengthened links in a connectionist model. For example, if one of the currently high-ranked user preferences is *asking for information about artists born in New York*, a query is automatically generated from the user model to look for this information connected to the individuals in the input graph.

## 4 Output Generation

Since our companion "thinks" in RDF statements, it requires mechanisms to communicate their con-

tent to the user. We distinguish two classes of RDF statements from which to generate natural-language output. The first class is RDF data that describes content from the domain ontologies, e.g. that Duke Ellington was born in Washington, DC. The second class are statements that describe a certain type of communicative intent, such as telling the user that she just asked the same question as five minutes ago.

Our approach for the second case is that of template-based generation, where each communicative intent from the ontology corresponds to a different template. The templates are described using the Velocity<sup>5</sup> template language, and can thus be extended separately from the program code, while still offering the possibility to make use of memory contents for filling slots in the templates.

The first case is handled by directly generating a sentence structure from the subject – predicate – object structure of the RDF graph. Triples are sorted by subject; subjects that also appear as objects are inserted as relative clauses. Statements that share the same subject are connected by coordination or relative clauses, depending on the type of relation, and so forth. The input may contain negation markers, which are realised as negative polarity items.

The surface string of predicates is generated by using a set of templates and morphological processing (e.g. pluralisation). For subjects and objects, a query on the knowledge base is performed to retrieve an adequate natural language representation. For example, while the name of a person is in the *name* property of the *Person* class, the name of a music album is contained in the property *albumTitle*. A mapping for each class to such a property is stored in an annotation file.

## 5 Related Work

Catizone et al. (2008) use an extended version of GATE's ANNIE subsystem, combined with a set of gazetteers, to identify relationships in the input to their Senior Companion system. The focus of the Senior Companion is to use the data extracted from the user utterances to collect information about the user's life. While our input analysis system is similar, it uses regular expression patterns over annotations for the matching of relations between, and properties of, individuals

<sup>5</sup><http://velocity.apache.org/>

and classes. In terms of functionality, our system focuses on being able to answer user requests and provide continued dialogue by taking into account the previous interactions with the user.

Episodic memory has first been distinguished from other memory types by Tulving (1972). Implementations have for example been used in artificial life agents (Nuxoll, 2007; Ho et al., 2003), in storytelling agents (Ho et al., 2007; Ho and Dautenhahn, 2008), and for non-player characters in games (Brom et al., 2007; Brom and Lukavský, 2009). Since our memory component is realised as an RDF graph, nearest-neighbour search as in the memory model proposed by Nuxoll (2007) does not directly apply.

Brom and Lukavský (2009) summarise important aspects of episodic memory and propose a more detailed concept of time categories than ours. In contrast to their work, our memory is not concerned with remembering locations, but with finding items relevant for current dialogue in the episodic memory of the agent, and thus stores different data.

Both the adaptive mind agent by Krenn et al. (2009) and Gossip Galore (Xu et al., 2009) describe companion systems able to answer questions on domain data. Both agents only have limited knowledge of their own past and do not use it for dialogue. Thus they cannot ground dialogue in their own experiences, and are unable to leverage knowledge about user preferences for providing more interesting dialogue.

Cavazza et al. (2008) describe a companion system for helping users plan a healthier lifestyle. Dialogue can be driven by the companion or by the user, but revolves around agreeing upon a daily exercise plan or negotiating re-planning in case of plan failure. Our system aims at a more open kind of dialogue which does not revolve around a plan model. Instead, the user is able to ask different kinds of questions on all the domain data available, which leaves the companion in a situation where much less expectations can be made towards the next user utterance.

## 6 Conclusion

We have presented a model of a companion that uses an RDF-based episodic memory component for enhancing dialogue with the user and grounding domain knowledge in interaction experiences interconnected with the agent's knowledge base.

The full implementation of the model is currently work in progress.

Retrieval of episodes is accomplished by using a set of competing SeRQL queries. Our model shows how the contents of past interactions and their relation to current dialogue can be employed by a companion for selecting the next dialogue move and generating dialogue content.

## Acknowledgements

The work presented is supported by the Austrian Ministry for Transport, Innovation and Technology (BMVIT) under the programme “FEMtech Women in Research and Technology” grant nr. 821855, project C4U. The Austrian Research Institute for Artificial Intelligence (OFAI) is supported by the Austrian ministries BMVIT and BMWF.

## References

- David Benyon and Oli Mival. 2008. Scenarios for companions. In *Austrian Artificial Intelligence Workshop*, September.
- Cyril Brom and Jirí Lukavský. 2009. Towards Virtual Characters with a Full Episodic Memory II: The Episodic Memory Strikes Back. In *Proc. Empathic Agents, AAMAS workshop*, pages 1–9.
- Cyril Brom, Klára Pesková, and Jirí Lukavský. 2007. Towards characters with a full episodic memory. In Catherine Pelachaud, Jean-Claude Martin, Elisabeth André, Gérard Chollet, Kostas Karpouzis, and Danielle Pelé, editors, *IVA*, volume 4722 of *Lecture Notes in Computer Science*, pages 360–361. Springer.
- G. Castellano, R. Aylett, K. Dautenhahn, A. Paiva, P. W. McOwan, and S. Ho. 2008. Long-Term Affect Sensitive and Socially Interactive Companions. In *Proceedings of the 4th International Workshop on Human-Computer Conversation*.
- Roberta Catizone, Alexiei Dingli, Hugo Pinto, and Yorick Wilks. 2008. Information extraction tools and methods for understanding dialogue in a companion. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*.
- Marc Cavazza, Cameron Smith, Daniel Charlton, Li Zhang, Markku Turunen, and Jaakko Hakulinen. 2008. A ‘companion’ ECA with planning and activity modelling. In *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, pages 1281–1284.
- Paul Ekman. 1992. An argument for basic emotions. *Cognition and Emotion*, 6:169–200.
- Birgit Graf, Ulrich Reiser, Martin Hägele, Kathrin Mauz, and Peter Klein. 2009. Robotic home assistant care-o-bot 3 - product vision and innovation platform. In *IEEE / Robotics and Automation Society: IEEE Workshop on Advanced Robotics and its Social Impacts - ARSO 2009*, pages 139–144, New York, NY, USA. Piscataway.
- Wan Ching Ho and Kerstin Dautenhahn. 2008. Towards a narrative mind: The creation of coherent life stories for believable virtual agents. In *IVA '08: Proceedings of the 8th international conference on Intelligent Virtual Agents*, pages 59–72, Berlin, Heidelberg. Springer.
- Wan Ching Ho, Kerstin Dautenhahn, and Christopher L. Nehaniv. 2003. Comparing different control architectures for autobiographic agents in static virtual environments. In Thomas Rist, Ruth Aylett, Daniel Ballin, and Jeff Rickel, editors, *IVA*, volume 2792 of *Lecture Notes in Computer Science*, pages 182–191. Springer.
- Wan Ching Ho, João Dias, Rui Figueiredo, and Ana Paiva. 2007. Agents that remember can tell stories: integrating autobiographic memory into emotional agents. In Edmund H. Durfee, Makoto Yokoo, Michael N. Huhns, and Onn Shehory, editors, *AA-MAS*, page 10. IFAAMAS.
- Brigitte Krenn, Marcin Skowron, Gregor Sieber, Erich Gstrein, and Jörg Irran. 2009. Adaptive mind agent. In *IVA '09: Proceedings of the 9th International Conference on Intelligent Virtual Agents*, pages 519–520, Berlin, Heidelberg. Springer.
- Andrew Nuxoll. 2007. *Enhancing Intelligent Agents with Episodic Memory*. Ph.D. thesis, Univ. of Michigan, Ann Arbor.
- Marcin Skowron, Jörg Irran, and Brigitte Krenn. 2008. Computational framework for and the realization of cognitive agents providing intelligent assistance capabilities. In *18th European Conference on Artificial Intelligence, Cognitive Robotics Workshop*, pages 88–96.
- Aaron Sloman. 2007. Requirements for Digital Companions It’s harder than you think. In *Position Paper for Workshop on Artificial Companions in Society: Perspectives on the Present and Future*. Oxford Internet Institute.
- Endel Tulving. 1972. Episodic and semantic memory. In E. Tulving and W. Donaldson, editors, *Organization of Memory*, pages 381–403. Academic Press, New York.
- Feiyu Xu, Peter Adolphs, Hans Uszkoreit, Xiwen Cheng, and Hong Li. 2009. Gossip galore: A conversational web agent for collecting and sharing pop trivia. In Joaquim Filipe, Ana L. N. Fred, and Bernadette Sharp, editors, *ICAART*, pages 115–122. INSTICC Press.