Diplomarbeit

Neuro-Psychoanalytically Inspired Episodic Memory for Autonomous Agents

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Diplom-Ingenieurs unter der Leitung von

o.Univ.Prof. Dipl.-Ing. Dr.techn. Dietmar Dietrich
und
Dipl.-Ing. Tobias Deutsch
als verantwortlich mitwirkendem Assistenten am
Institutsnummer: 384
Institut für Computertechnik

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

von

Andreas Gruber
Matr.Nr. 0025482
Jörgerstraße 39/9, 1170 Wien

Wien, am 15.5.2007 ____________________________

Kurzfassung

Autonome Agenten nehmen ihre Umgebung wahr und müssen Entscheidungen selbstständig treffen. Das „Artificial Recognition System“ (ARS) ist ein bionischer Ansatz eines autonomen Systems zur Informationsverarbeitung und Entscheidungsfindung. Ziel dieses Projektes ist, auf unvorhergesehene Situationen zuverlässig und flexibel zu reagieren.

Das episodische Gedächtnis umfasst die Autobiographie eines Agenten und ist für die Erinnerung persönlicher Erfahrungen zuständig. Ein intelligentes System, das diese Fähigkeit besitzt, kann Episoden wieder erinnern, zukünftige Situationen antizipieren oder die Auswirkung von bestimmten Verhaltensweisen vorhersehen. Unerwünschtes Verhalten kann vermieden oder erwünschte Ziele durch die Rekonstruktion von Episoden erreicht werden.

Die vorliegende Arbeit beschreibt den Entwurf eines episodischen Gedächtnismodells und dessen Integration in das ARS System. Das vorgestellte Modell basiert auf Konzepten der Neuropsychoanalyse und Psychologie und behandelt die wesentlichen Funktionalitäten in der Verarbeitung von episodischen Erinnerungen—die Kodierung, die Speicherung und den Abruf. Die Kodierung ist für den Erwerb neuer Informationen zuständig. Die Erfahrungen des Systems werden bei relevanten Ereignissen erfasst. Die Speicherung verwaltet die abgelegten Erinnerungen. Die Episoden sind entsprechend dem Kontext, in dem sie sich ereignet haben, gespeichert und mit einer emotionalen Bewertung versehen. Ein Vergessensmechanismus beschreibt den unterschiedlichen Zugriff auf gespeicherte Erfahrungen. Die Aufgabe des Abrufs ist, bei vorhandenen Reizen passende Erinnerungen zu aktivieren. Dabei wird sowohl beabsichtigter als auch spontaner Abruf berücksichtigt. Spontane Erinnerungen tauchen ohne explizite Bemühungen durch das System auf.

Die Implementierung der Software wird in einer Simulationsumgebung evaluiert. Autonome Agenten navigieren in einer virtuellen Umgebung um Energiequellen zu finden und diese zu konsumieren. Die Ergebnisse zeigen, dass die Agenten fähig sind, Episoden in einem ähnlichen emotionalen Zustand wieder in Erinnerung zu rufen und mögliche zukünftige Situationen zu antizipieren. Damit sind die Agenten in der Lage, die emotionale Auswirkung von bestimmten Verhaltensweisen zu analysieren und vorherzusehen. Gestützt auf ihren Erfahrungen können sie somit ihre Entscheidungsfindung verbessern.

Abstract

Based on perception of environmental data, the task of an autonomous agent is to make decisions without external help. The "Artificial Recognition System" (ARS) is a bionic approach to build an autonomous system for information processing and decision making. Its objective is to deal with unforeseen situations in a reliable and flexible way.
The concept of an episodic memory system is concerned with memorizing due to personal experiences. Episodic memory inherits the history of an agent and is crucial for remembering these personal incidents. The capability of remembering past experiences is essential for an intelligent system to recognize similar episodes, to anticipate future situations, or to predict the impact of certain behaviors. Equipped with that functionality, the autonomous system shall be able to avoid unwanted behavior or reach a particular desired goal by reconstructing episodes.
The architecture of the computational episodic memory model designed and integrated into the ARS system is inspired by concepts of human memory research from neuro-psychoanalysis and psychology. The presented framework describes the fundamental stages in episodic memory processing—encoding, storage, and retrieval. Encoding is concerned with the acquisition of new information. An event based approach is proposed to capture the experiences of the system. Storage covers all processes that maintain the stored experiences. Episodes are stored according to the context in which they happened attached with an emotional evaluation. The proposed forgetting mechanism describes the crucial maintenance functions to improve the accessibility of essential episodes. The task of retrieval is to recall an appropriate experience given a certain stimulus. Both, deliberate and spontaneous retrieval are considered. Spontaneous recollections characterize the memories that pop up without the explicit effort of the system.
The software implementation is evaluated in a simulation environment. Autonomous agents navigate in a virtual environment in order to find and consume energy sources. The results show that past episodes happening in a similar emotional state can be successfully recognized. Furthermore, possible future situations can be reliably foreseen. The agents are able to analyze and predict the emotional impact of certain behaviors which improves their decision making.

Acknowledgements

First of all, i want to thank professor Dietmar Dietrich, head of the Institute of Computer Technology, for the opportunity to write this thesis and to contribute to his work. Special thanks go to my supervisor Tobias Deutsch for his excellent support. He has taken special effort to advise me through all of this work. Further special thanks go to my “second” supervisor Roland Lang for his great support to my thesis. My gratitude also belongs to the whole ARS project team for advising me and for their fruitful and interesting discussions. Last but not least I want to thank my parents for the opportunity to study and their support through all of my students career.

Contents

Chapter 1

Introduction

In future, automation systems will be equipped with more and more sensors to control functionalities [Rus03]. The main reasons for this trend can be found in the progress of sensor technology, miniaturization, and communication technologies. Most of the existing control systems are not able to meet the requirement the demand of processing this huge amount of data. A new approach that deals with this emerging challenge is the "Artificial Recognition System" (ARS)[1] [Pra06]. The goal of this ongoing project is to develop an intelligent autonomous system for ambient or robotic automation systems. Therefore ARS deals with situation recognition in a reliable and flexible way. The more sensors are integrated in an automation system, the better a perceived situation can be analyzed, and the more sophisticated a system might react. To process such vast amount of data in an efficient way, the approach of the ARS system is inspired by the human mind. The way the human mind handles huge amount of information is unique in its technique. The objective of the ARS project is the creation of a general control architecture for an autonomous, emotionally supported reasoning system using concepts of neuro-psychoanalytic research. While neurosciences describes the brain bottom-up, psychoanalysis describes it top-down. Based on perception of information (bottom-up), the task of the autonomous system is to make decisions (top-down) without external help.

An episodic memory improves decision making of the system providing essential information on the basis of past experiences. The episodic memory is that kind of memory, that covers all experiences an entity has made in its "life". It captures the autobiographical history of ones past. In contrast to semantic memory, which contains general knowledge like "Vienna is the capital of Austria", the episodic memory deposits all personal incidents. A memory that reminds you of a trip to a certain town would be for example episodic. The episodic memory creates the autobiographical history of an individual.

To remember experiences is an essential capability of an intelligent, autonomous system. The episodic memory gives such a system a personal history that remembers it of its personal past. An autonomous system that is able to access its history, may be aided in decision making in various ways. It can be used to anticipate future situations happening in a similar context or to predict the outcome of certain behaviors. This ability allows the autonomous system to evaluate alternative behaviors. Certain unwanted impacts of possible behaviors may be avoided as well as some desired goals may be reached by reconstructing experienced

[1] `http://ars.ict.tuwien.ac.at/`

past episodes. Furthermore, a system incorporating an episodic memory may deliberately reflect about its past which can be used for learning. Behaviors can be improved by learning from own mistakes or from rewarding excellent behaviors.

The task of this work is thus to develop an episodic memory inspired by concepts of neuro-psychoanalysis to enhance the ARS system in evaluating perceived situations and to aid in decision making. The intention of this work is not to reproduce the complete structural architecture of a human episodic memory for implementation in a technical system, but to implement the most important characteristics of this research area in a computational model. The generic episodic memory model provides general functionalities of human episodic memory processing. However, the focus of the design is also its usage for autonomous agents applied in ambient or robotic automation systems.

1.1 State of the Art

Episodic memory (EM) has been researched comprehensively by various neurologists, psychologists and psychoanalysts. There do not exist a lot of computational models of an EM that implement the complex and profound findings of these scientists in a technical system in a sophisticated way. This section presents a selection of previous computational approaches of building an EM model in a system.

One of them is the computational EM model developed by Nuxoll and Laird ([NL02], [NL04]) with the purpose to integrate it in the Soar[2] [LLR06] architecture. It tries to model human episodic memory as well as to build a task-independent and architectural model for artificial intelligence agents. The framework of Nuxoll is based on the theories of episodic memory processing stated by the psychologist Endel Tulving [Tul83] and decomposed into the functional stages encoding, storage and retrieval. The EM can be used to aid decision making by evaluating alternative actions and to deliberately reflect about a task. Encoding is initiated each time an action is taken. Among the stored episodes is no organization. The episodes are stored attached with an activation level [NLJ04], which does not decay. For deliberate retrieval, a cue is compared to all episodes that contain at least one element from the cue and the episode with the best match is selected. A partial matching algorithm has been introduced for retrieval. The EM module is evaluated in a "Pacman"[3] like game where agents navigate in a grid world—which consists of cells that are either empty or contain points—in order to reach a high score. The possible actions of an agent is to select a certain direction (east, west, south, or north). Agents acting on the basis of stored episodes outperformed agents that randomly select their actions. Although this implementation achieves a pretty good performance, it has as well some limitations. Forgetting and spontaneous retrieval is not implemented as well as the retrieval phase may be slow due to the linear search through all stored episodes. Even though the author stated as one intent of this work to model human episodic memory, emotions in combination with memory have not been considered.

Similar to the work of Nuxoll, the EM model of Tecuci [Tec05] is based on Tulving's theory of EM processing [Tul83] too. The goal of Tecuci's work is to design a generic EM module which can be used for a wide number of applications. The work addresses the structures and organization of an episodic memory, the maintenance due to changes and the retrieval of

[2]Soar—States, Operators and Reasoning—is one of the most widespread cognitive architectures [LLR05].
[3]Pacman is a video game (`http://en.wikipedia.org/wiki/Pac-Man`).

the memories. The memory system is driven by expectations—new episodes are interpreted on the basis of the expectations the system has of what will happen. A generic episode is composed by a triple of context, content, and outcome. Semantic memory is used to encode specific episodes, that will provide general knowledge about the application domain like performable actions, effects and goals that can be perceived. Episodes that are similar are grouped by containers, which are called *memory organization packets* (MOP). The MOPs are organized in a similarity network [Tec05]. Memories are encoded including all known parameters, like actions and goals. Retrieval is both deliberate and spontaneous. Spontaneous retrieval is a by-product of the storage of a memory—if there is a similar episode next to the appropriate storage location, this episode will be retrieved (this effect is called *reminding*). Deliberate retrieval is a search process that traverses the memory and returns the memory that best matches the query. A great advantage of this approach of an EM module is that it is not domain dependent and thus applicable to a wide range of applications. It integrates semantic memories and other aspects of human memory like spontaneous and voluntary retrieval are considered. Beside this, it does not implement a forgetting mechanism as well as other issues like the tight coupling of episodic memories to emotions.

An episodic memory system with emotions is designed and integrated in a cognitive humanoid robot called ISAC[4] [DG05]. ISAC is developed to work safely with humans. It interacts with the world through air-powered actuators, a computer display system, voice, and multiple sensors. The EM system is designed to retrieve the correct episode from the memory using emotions and statistics. Past experiences are recorded in the EM system in a time-indexed format. An episode consists of various semantic memory (SM) units. One main characteristic of the EM retrieval process is that the SM units of the retrieved memory should be similar to the current situation. Furthermore, recently accessed episodes should score higher, and new aspects are more important than common ones. Emotions in the cognitive robot are used to give an experienced episode emotional significance. The impact of emotions is implemented in a history component. It contains a time-dependent exponential decay function which is indirect proportional to the emotional salience and increased each time a memory is accessed. The author states that the emotional salience is very important to select the correct memory units—the robot should remember highly salient episodes very well. When the history value of a memory decays under a certain cutoff-value, the memory will be removed. In various experiments [Dod05] has been demonstrated that ISAC performs well for information that is both novel and salient. This approach of an EM system has some features in common with aspects of human memory. Firstly, it realizes the tight relationship between the EM system and the SM system. Furthermore, it integrates emotions which give the system a humanoid behavior. Forgetting is implemented as a kind of activation (the history component), which level depends on the emotional level and decays with an exponential rate. A limitation of the model is that memories are removed if their activation gets below a certain cut-off value.

The aim of the work of Dautenhahn and Ho ([HDNB04], [HDN05]) is to build a generic adaptive memory control architecture inspired by human memory for autonomous agents to survive in a virtual environment. The virtual environment consists of different landforms (oasis, desert, mountains,...) each with their unique resources and features. The agents are equipped with 90 degrees wide opened object and landform sensors and have four physiological variables—*glucose, moisture, energy, and body temperature*—which have to be kept in certain range to keep alive. The agents incorporate event specific knowledge which is a long list of their own experienced history. Each memory record represents a specific situation containing

[4]ISAC—Intelligent Soft Arm Control

the surrounding objects, the landform, and the internal physiological variables. If one of the agent's internal variables gets out of its bound, an event reconstruction process is triggered. Therefore the whole memory is traversed to retrieve a relevant event. An event is a series of situations. It is searched for various groups of events happening in different periods of time depending on a key record. The key record contains the information for satisfying the current internal needs of the agent. The process of finding an appropriate situation to the current one can be done backwards for a *redo* event—repeating a situation by executing the actions in the original order—or forward for an *undo* event—executing the previous actions reversely to reach the desired situation. In an experiment, agents equipped with such an EM structure outperformed agents that act purely reactive. Although this approach has implemented a very flexible event reconstruction mechanism (events as series of situations are dynamically reconstructed), there are some limitations that have to be noted. First of all, this model is very domain dependent. It is only applicable for navigation problems, where agents have to find a "best path" in an environment. No forgetting mechanism is implemented, and further important features of human memory like emotions or semantic memories have not been considered.

1.2 Problem Statement

The bionic approach to decision making elaborated by the ARS system [Pra06] should be extended by an episodic memory system. The ability to access past experiences allows a more appropriate and accurate form of decision making. The EM module has to capture, maintain, and appropriately return the happenings the autonomous system has made in its "life". An effectively designed EM system has to be able to recognize past experiences for a given situation to be used for future situation evaluation. Furthermore, future situations should be anticipated whenever possible and their impact predicted to select the "best" behavior between various alternatives. Emotions are used in the ARS system for the evaluation of perceived situations. On the basis of this biologically inspired evaluation system, episodes as longer sequences of situations should be stored with an emotional rating.

The plan for designing such a model originates from concepts that are known from researches in human episodic memory. Theories from neuro-psychoanalysis and psychology have to be explored to derive a computational memory model that implements the most important features from these research areas. The concept of the EM model for the ARS system should be applicable for a wide range of applications. The design of the ARS system generic—it models functionalities of the human brain in a technical system instead of developing for a particular problem domain—the target applications are from the fields of ambient or robotic automation systems, like building automation.

To evaluate the ARS system, a simulation environment where autonomous agents navigate in order to find and consume energy sources serves as test environment. The agents interact with the environment by using various sensors and choosing various behaviors based on results of the situation evaluation and decision making stages of the embedded ARS system. This architecture has to be used for the development of the EM module. The preprocessed perceptions (both external and internal) of the autonomous agents serve as basis for the EM module. The main focus of this thesis is to determine, how the stored experiences are maintained in memory and how the experiences are retrieved to be appropriately used, when an agent records new experiences.

Chapter 2

Background

In the first section of this chapter some findings of cognitive scientists about human episodic memory are summarized to be used as inspiration for the later implementation of an computational model of episodic memory. The basic distinction of human long-term memory into semantic, procedural and episodic memory is elucidated followed by an explanation of the fundamental functional stages of episodic memory processing. The section is concluded with the presentation of a conceptual framework for episodic memory.

In the second section, the ARS-project ("Artificial Recognition System"), for which the episodic memory system is designed, is introduced. The ARS project deals with situation recognition for robotic or ambient automation systems. ARS researches for a psychoanalytical model based on Sigmund Freud's ego-superego-id model. This model is extended by the episodic memory concerned in this thesis. Beside a brief introduction into the ARS-project, the environment where the episodic memory module will be embedded and the interface to it are depicted.

2.1 Human Episodic Memory

Human memory is a topic, which has been explored by several cognitive scientists from various different perspectives. For the objective of this thesis—to build an artificial episodic memory (EM) system for autonomous agents—theories about human memory and human EM of the scientists Mark Solms (neuro-psychoanalyst) [ST02], Alan Baddeley (psychologist) [Bad97], Endel Tulving (psychologist) [Tul83], and Daniel Schacter (psychologist) ([Sch96], [ST94]) are summarized in this section. The intent of this summary is not to describe human EM completely and in detail, but to highlight some aspects that are important for a later implementation in a computational model. In the first part of this section, EM as a part of human long-term memory is depicted. One possibility for a functional classification of long-term memory into semantic, procedural and episodic memory is outlined followed by a more detailed distinction between the episodic and semantic memory system. In the second part, the three fundamental processing stages of memories—encoding, storage and retrieval—are elucidated with special considerations on episodic memory processing. The concept of consolidation is further presented. In the final part of this section, Tulving's general abstract processing system [Tul83], a conceptual framework for the study of episodic memory is briefly considered. It describes the act-of-remembering: an act starting from the encoding of an experience leading to its recall and mental replay.

2.1.1 Episodic Memory: A Part Of Long-Term Memory

Human memory characterizes various mental functions. Sometimes the term "memory" is used for the "act of remembering"—the retrieval of a previously stored experience or fact. Occasionally it is not used as a synonym for the reactivation of our knowledge, but as a representation of our knowledge itself. In this sense, the term "memory" is used to denote the part of the mind, that contains the traces of influence from the past [ST02, p. 140].The storage of human memories can be divided into two components: short-term memory (STM) and long-term memory (LTM). STM contains memories of facts or events that are present in mind in this moment, and refer to happenings that arose a few seconds or hours ago. LTM in contrast contains both recent and older memories [ST02, p. 143]. Human LTM can further be categorized into sub-subsystem: semantic, procedural and episodic memory. A general consensus has emerged among neuropsychologists, neurobiologists and other cognitive psychologists that some kind of classification of memory is fundamental [ST94, p. 1-2]. Memory represents a number of separate but interacting systems. All these systems make possible the utilization of acquired and retained knowledge. Memory systems are mainly defined in terms of the kind of information they process and the principles of operations [ST94, p. 13].

> **Semantic memory (SM):** "Is a network of associations and concepts that underlies our basic knowledge of the world—word meanings, categories, facts and propositions, and the like" [ST02, p. 150].

SM contains objective information about the world and how it works e.g. cats have four legs. The content is general and does not have any personal character in the sense of representing experiences. We are speaking of SM when we say "I know".

> **Procedural memory (PM):** "Is a kind of bodily memory. It is memory for habitual motor skills, or, more generally, perceptuomotor or ideomotor skills" [ST02, p. 156].

PM represents knowledge like how to run, how to play guitar, and so on. It covers acquired skills through lots of training runs, that we can recall whenever we need them. The third memory sub-system in the classification is episodic memory.

> **Episodic Memory (EM):** "Involves the literal reexperiencing of past events—the bringing back to awareness of previous experiential episodes" [ST02, p. 160].

EM is the memory we are speaking of when we say "I remember". It allows us to recall the personal incidents that uniquely define our lives. Memories are subjective and come along with the reliving of past moments of experience.

These functionally categorized memory systems interact in a certain extent with each other. In literature, there can be found an additional distinction between procedural and propositional memories. Propositional memory is composed of EM and SM [Tul83, p. 33]. This composition of the EM and the SM system into the category propositional memory points out the fact that is widely agreed, that these two memory systems are closely interacting with each other.

Episodic Memory versus Semantic Memory

Tulving states that EM and SM should be regarded as separate, albeit closely interacting systems which have some similarities but as well some differences [Tul83, p. 32]. As mentioned above, the distinction into memory systems is arbitrary. The definition of an EM or an SM is done due to its functionalities. As both are memory systems, they are concerned with the acquisition, retention and utilization of information and knowledge. Informations are acquired through senses. The retention of the stored information is done as a result of mental activity. The individual is not aware of the knowledge it possesses. The utilization (retrieval) is initiated by stimuli or cues. The presence of a retrieval cue only activates a small part of the available information (retrieval is selective).

Feature	Episodic	Semantic
Informations		
Source	sensation	comprehension
Unit	events; episodes	facts, scripts,...
Organization	temporal	conceptual
Operations		
Registration	sensation	comprehension
Affect	more important	less important
Inferential capability	limited	rich
Context dependency	more pronounced	less pronounced
Access	deliberate	immediate
Retrieval queries	time? place?	what?
Retrieval consequences	system changed	system unchanged
Recollection	personal (remember)	impersonal (know)

Table 2.1: Summary of differences between episodic and semantic memories [Tul83, p. 35]

Beside these similarities, there are lots of differences between EM and SM. Some features that distinguish episodic and semantic memories are summarized in table 2.1. According to differences in information, events in EM are recorded upon mere sensation (e.g. due to the noise of a passed by car). To store knowledge in SM, happenings have to be comprehended and understood. “Both, the episodic and semantic memory systems register only change; if some information already exists in the system, the same information is not entered again” [Tul83, p. 37]. The prototypical unit of information in EM is an episode, or as a synonym for it, an event. An event is something that occurs in a particular situation and always has a beginning and an end in time [Tul83, p. 142]. Tulving further distinguishes between simple events and complex events. While simple events like seeing a flash arise due to a particular change in the individual’s environment, complex events are lasting longer in time, like a trip over several weeks. Complex events further require interpretation of their components in terms of SM. Events are ordered in time—one precedes another—and may be nested or overlapped in time. The encoding of an event further may include reference to other related or similar events. The prototypical units of information in SM are categories, facts, scripts, propositions and the like. The knowledge in EM is organized temporally and relatively loose—initially precisely recorded information about an event can be easily lost. Details of complex events are better remembered than details of simple events. In SM, knowledge is organized conceptual—by a wide variety of relations between the SM units.

According to differences in operations, episodic memories are acquired more rapidly, while semantic memories are gradually learned. While the EM system mainly registers perceptible properties, semantic memories are given to us in symbolic form containing higher level knowledge. Episodic memories have affective components, while affects in SM are less important. "Personal experiences are often emotional, or take place while the person is in a particular mood, and information about the state of the rememberer may be recorded as part of the memory trace of an event" [Tul83, p. 43]. This affective component of an encoded event can later be retrieved and relived. SM possess a rich inferential capability—the capability to extract more information from an input than provided—due to its tightly coupled conceptual knowledge structures. This capability is very limited in EM. Events are what they are, they occur when they occur, knowledge about their content has to be deduced from another knowledge base, e.g. from SM. Operations in EM are more context dependent—the processing of an unit of information in EM may be influenced by other units. It is proposed that a principal characteristic of EM lies in its handling of contextualized knowledge [Tul83, p. 44].

Access to EM depends on the subjects mental state. A perceptual change in the current environment—a change in the current situation or the appearance of a stimulus object—is interpreted immediately by SM. In contrast, to retrieve a memory from EM due to a perceptual change, the subject must be in *retrieval mode*. For example, if we hear a word, its meaning is immediately interpreted by SM. But we do not necessarily remember the last event when we heard this particular word. The general form of retrieval queries in the episodic system is typically built up in a way like: *What did you do at time T in place P?* It usually consists of temporal, spatial, and behavioral information. It contrast, a query for SM could be: *What is X?* X might be an object, property, and the like. The retrieval of an event in EM tends to change the stored information and makes it further more likely to be retrieved upon a similar stimulus. SM remains more the less unchanged due to a retrieval. Remembered episodes tend to have an affective tone unique for each subject, they are personal experiences that belong to the individual. In contrast, retrieved semantic memories—as we term "I know..."—represent impersonal experiences.

Tulving regards the episodic and semantic systems as two functionally different memory systems. That does not mean that the systems are completely separate and have nothing to do with each other. He means, that each system *can* operate independently from the other, although not as efficiently as with the support of the other system. Despite Tulving's distinction, there exist a general agreement that episodes have semantic contents and that the two systems are closely interdependent and interact with each other all the time. Tulving further cites that evidence exists "for the important role that the semantic system plays in storage and retrieval of episodic memory information" [Tul83, p. 65]. He particularly points out the important role of the semantic system when retrieving episodic information in its ecphory theory. Ecphory means, that the retrieval of episodic memories is a synergistic process in which the cue information (semantic) and the episodic information are combined (see section 2.1.2). The tight relationship between EM and SM can be illustrated by the following example [Tul83, p. 64]. Some of the knowledge in SM is stored in the form of scripts. These scripts may be learned through the repeatedly personal interaction with the world. For example, the script *going for dinner in a specific town* is learned by frequently executing this. A script may be stored in SM that contains all relevant informations about such dinners. If you are faced another time to a particular situation in a restaurant in that town (e.g. ordering dinner), you might *know*—by retrieving that script from SM—how to behave in that specific situation. In contrast, if you have only been once in a restaurant in that town, you would try to *remember* that particular episode from EM.

2.1.2 Functions Of Human Episodic Memory

The processing of human memories can generally be classified into three fundamental functional stages. These are encoding, storage, and retrieval. Encoding is the process responsible for acquiring new information. Storage deals with the retention of this information in memory. Retrieval is responsible for bringing the memory back into mind [ST02, p. 140] (figure 2.1). This functional classification is complemented with the concept of consolidation—a

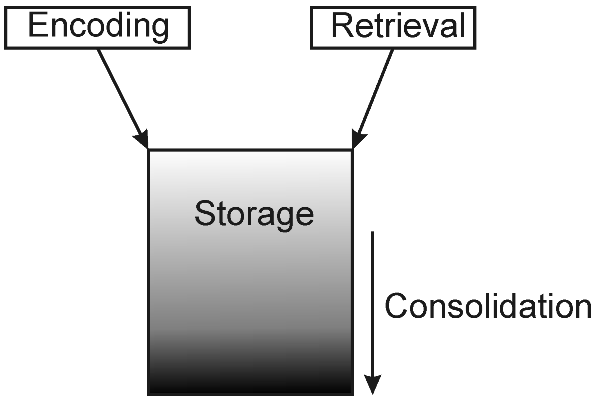

Figure 2.1: Functions of human memory [ST02, p. 141]

concept that takes an important part in human memory research. The process of laying down memories into deeper and deeper levels of memories is called consolidation [ST02, p. 142]. The concept of consolidation can be considered as a storage process. The fundamental functionalities of human memory including the concept of consolidation are further discussed in this section under special considerations of the EM system.

Encoding

The encoding process is a procedure for transforming a perception into a memory [Sch96, p. 42]. Encoding in EM describes the act of codifying a real episode into a memory trace or an *engram.* It begins with the perception of an episode and ends with an encoded engram, which may later on be recoded [Tul83, p. 142]. As addressed in section 2.1.2, Tulving uses the term event as a close synonym for an episode and defines an event as:

> **Event:** "Something that occurs in a particular situation" [Tul83, p. 37].

It is something that occurs in a certain place at a particular time. One characteristic is, that it always has a beginning and an end in time. The beginning and end may be so close together, that the event happens instantaneously. Events are temporally related with each other, one precedes another, they may overlap, or be nested. They also can be described in terms of some actions executed by an actor. Tulving further notes, that the term *episode* usually carries with it the connotation of an event that occurs in an ongoing series of events. To define features of an event, Tulving decomposes an event into a *setting* and a *focal element*. The setting refers to the place and time in which the event occurs while the focal element describes the salient happening within the setting. The focal element can be thought of as the personal significance of an event, i.e. a picture against a background. Events never repeat, they are unique [Tul83, p. 143].

Encoding is a central concept in the theory of memory. One of the most crucial parts is the time when it occurs. Tulving states that the rememberer is not aware of the encoding process the most time and it always occurs when information about a perceived event is stored in memory [Tul83, p. 151]. An event is recorded in terms of its salient happening. Encoding depends a lot on mental activity. A large part of what is perceived is influenced by the current state of the mind. The result of the encoding process is the engram. It can be seen as the difference of the states of the memory system before the encoding of an event and after it. An engram consists of a bundle of features that not necessarily correspond to the features of the original event. The point of view that an engram is composed of a bundle of features makes it possible to specify similarity relations among events, their engrams, and recollective experiences. Further it provides a basis for retrieval. The features of an engram can decay at different rates, implying that the composition of a memory can change over time. Engrams of different events are similar or different depending on the extent of shared or distinctive features they possess.

The processes which describe all kinds of changes of an engram are called *recoding*. Recoding happens when an event is repeated. As mentioned before, every event is unique and therefore cannot occur a second time. But in situations with a very limited scope, two events may be very similar. There exist two possibilities: should the original engram be modified or should the engram of the repeated event be stored separately? Tulving discusses that it is not quite clear what is the right choice, but he argues that the engram should be modified. Events influence each other and the original memory trace should therefore be adapted. “When an event occurs, its encoding may include reference to other similar or related events, and the corresponding information is recorded as part of the trace of the event” [Tul83, p. 42].

The encoding of an event ends with the formation of the engram and an optional change of this engram due to a repeating event and the recoding process. Once, an engram is formed, it exists in a latent form in the memory.

Storage

Storage covers all processes relevant for the retention of the stored information. Memories are stored in several places and in different manners. This highly redundant storage of human memory is represented by “filing cabinets” [ST02, p. 150]. As considered in section 2.1.1, the storage part of figure 2.1 can be functionally separated into the memory sub-systems SM, PM, and EM, which represent these filing cabinets. The concept of consolidation—which is relevant for each memory sub-system—plays an important role in the storage stage. Not all perceptions from STM are handed over to LTM. The consolidation mechanism can as well be seen as a process of sifting. It does not only lay down memories in deeper and deeper levels, it also sorts out memories that should not be retained. The physiology of consolidation can be seen as a process of wiring: "cells that fire together, wire together" [ST02, p. 146]. Thus the probability for a further activation rises. Circuits that are no more used, die off ("use it, or loose it").

Anatomically, episodic memories are mainly encoded in the *hippocampus*[1]. It is coupled tightly with a group of structures that are called the *limbic system*. The network of structures that builds up the limbic system was originally not discovered in combination with the memory, but with the emotions[2]. This underlines the feature, that episodic memories are not only stored, but are relived as well—they are in its nature emotional [ST02, p. 163].

Forgetting is a very fundamental part of human memory. "Freud is supposed to have said that once a memory is laid down, it can never be forgotten" [ST02, p. 149]. With this statement, Freud wanted to emphasize the persistence of memory. A similar point of view has Tulving with his differentiation between "availability" and "accessibility" of a stored memory. To Tulving, forgetting arises to a great part due to the inaccessibility of still available information in memory. The term availability refers to the hypothetical presence of a memory, whereas accessibility refers to that part of the available memory that can be recalled [Tul83, p. 203]. There exist various experiments in psychoanalytic literature that test the ability of remembering autobiographical events. Most of them deal with so called *diary experiments*. The experimenter records its own diary, and tests himself after a specific period of time whether he is able to remember the events.

So did Willem Wagenaar, a Dutch professor of psychology (the experiment is detailed in [Bad97, p. 218-219]). He recorded one or two incidents each day. He scaled each incident for its salience— whether it was unusual or something that happened frequently—assessed the degree of emotional involvement he was feeling, and the pleasantness of this involvement. After a period of time he tested himself if he could recall the incidents. He cued himself with different combinations of stimuli referring to the features of the event (who?, what?, where?, when?). The results concluded from this experiment (figure 2.2) are that the ability of successfully retrieving an event, or reversely, the rate of forgetting is influenced by the salience of an event, the emotional involvement of the subject, and the richness of the cues provided. The more salient an incident is, and the higher the emotional involvement, the better its probability for recall. Further he noted an influence of the richness of the retrieval cue. The more features are used to cue, the better an event could be recalled. If all retrieval cues where used, he could remember nearly every occasion.

Despite these profound determinants on the rate of forgetting, there have been observed some more. First of all there is a general tendency for older memories to be recalled less frequently. Lengthening the delay between encoding and retrieval dramatically increases forgetting—memories become less accessible with the passage of time. The rate of forgetting further slows down by the passage of time [Sch96, p. 73]. There has been observed a relation between the percentage of correctly recalled memories and the retention time, that is exponentially dependent on time [Bad97, p. 220].

Furthermore, the ability of remembering an event increases the more frequently the stored memory has been rehearsed by subsequent recollections. This can be referred to the term

[1] "The hippocampus is a folded piece of primitive cortex that lies on the inner surface of the fore brain within the temporal lobe" [ST02, p. 163].

[2] An emotion is an internally directed sensory modality. It is a state dependent function and reflects bodily changes [ST02, p. 106].

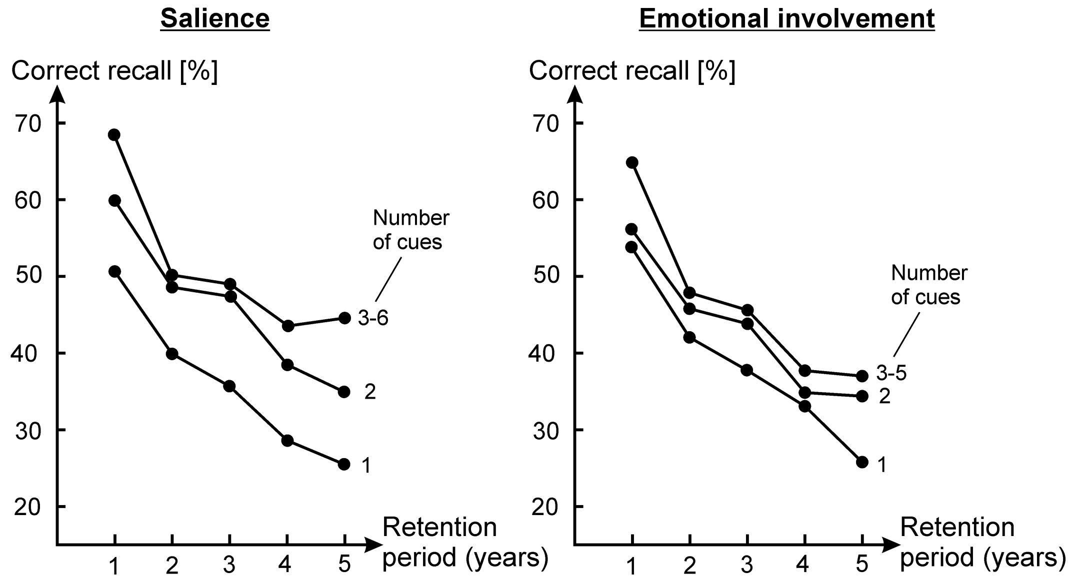

Figure 2.2: Influence of the salience and the emotional involvement of an event on the rate of forgetting [Bad97, p. 219]

learning. Baddeley states, that beside others the term learning can be legitimately used for remembering a personal incident. Therefore a certain amount of practice is necessary [Bad97, p. 108]. This is known as the *retrieval practice effect*: "the act of successfully recalling an item increases the chance that the item will be remembered" [Bad97, p. 112]. Schacter further describes the strengthening of memories by frequently recalling an incident as an effect of consolidation. "Once, an experience has been repeatedly retrieved, it becomes consolidated" [Sch96, p. 87]. Another popular explanation of forgetting is described with the *interference theory*. The key issue is, that forgetting arises due to competition between memory traces. An earlier learned memory might be over learned by a more recently event and therefore the probability for retrieval of the earlier image decreases. The less frequent a memory trace is recalled, the less the chance of further recalling the event (retrieval practice effect) which leads to forgetting [Bad97, p. 176].

There exist great evidence of the power of emotions on memory. Whereas very strong emotions, for example in panic situations can produce massive disturbances of memory (repression), there is a general tendency that events recorded with high emotional arousal lead to enhanced later recall [Bad97, p. 276]. Both rehearsal and emotion are the most relevant features to understand why certain memories stay with us for much of our lives [Sch96, p. 209]. A further emotional effect on memory is the influence of mood. Mood may act as a powerful associated context. Anything experienced in a specific mood will tend to be remembered more easily when the mood is reinstated (*mood state-dependency*) [Bad97, p. 281]. Hence, sad events are better recalled when one is feeling sad whereas happy events are better recalled when one is feeling happy.

Retrieval

A very high number of engrams representing encoded events can exist in the memory. The task of the retrieval process is to recall specific engrams for *recollective experience*. Tulving uses the term recollective experience to describe the mental experience of remembering [Tul83, p. 185]. A synonym for this term could be *to have a memory image*: remembering a past event concludes that the rememberer has a memory image of it and is aware of it in the form of a mental replay. To obtain the objective of the retrieval process—recollective experience—two necessary conditions must be fulfilled. Firstly, the system must be in *retrieval mode* and secondly an appropriate *retrieval cue* must be formed.

To be in retrieval mode means that the subject must be aware for retrieval. As considered in section 2.1.1, access to episodic information depends a lot on the subjects mental state. Retrieval can be triggered spontaneously or deliberately. Since retrieval can occur spontaneously and memories may come unbidden into mind, we are not aware how we entered into retrieval mode. So its very difficult to define how to get into retrieval mode and when to be in it [Tul83, p. 169]. When the memory system is in retrieval mode, a retrieval is initiated by the formation of a *retrieval cue*. A cue can be imagined as a search key in a memory—e.g. when searching in the telephone book for a number. It is like a stimuli to get the correct information. Retrieval is always cued. Tulving tries to describe the effectiveness of an retrieval cue with its *encoding specificity principle*: "recollection of an event, or a certain aspect of it, occurs if and only if properties of the trace of the event are sufficiently similar to the properties of the retrieval information" [Tul83, p. 223]. Tulving emphasis on the interaction of the stored information (the engram) and the retrieval information (the cue). There must exist a *compatibility relation* between the engram and the retrieval cue. This compatibility relation could be matching of features, associative relatedness, informational overlap, and the like [Tul83, p. 224]. But not all experiences are potentially recallable. Sometimes they are not accessible because the right cues are not available or because the relevant engrams have weakened [Sch96, p. 64].

The retrieval process has to find a match in episodic memory between the retrieval cue and a stored engram. As depicted before, the essence of Tulving's encoding specificity principle is that the probability of recalling an event depends directly on the similarity of the engram and the retrieval cue. The result of this process leads to recollective experience. What the subject remembers heavily depends on the kind and the amount of information that is stored in memory and what is found by the retrieval mechanism. Tulving hypothesizes that recollective experience not only derives from the stored engram in EM, but also from the retrieval cue. This synergistic effect of retrieval information and stored information Tulving calls *ecphory* [Tul83, p. 180]. Ecphory is a process that combines information from the engram and the cue. The retrieval cue not only selects features or components from an engram, it complements the engram. So this is an activity that makes use of components from EM (the engram) and from SM (the cue). Schacter also accepts the idea that a memory is an emergent property of the cue and the engram. Anyway, to which extent the retrieval cue contributes to the subjective experience is controversial among scientists who study memory [Sch96, p. 71].

2.1.3 Tulving's General Abstract Processing System

The general abstract processing system (GAPS) of episodic memory is a conceptual framework for the study and understanding of EM. It is *general* as it applies remembering of all sorts of events, *abstract* as the specific nature of the components is unspecified and it is a *processing system* as it is not a structure of memory but it has to do more with activity and function [Tul83, p. 131]. The GAPS describes the basic unit for the analysis of episodic memory—the *act of remembering*—which Tulving defines as:

> **Act of remembering:** "An act that begins with an event and that is perceived by the rememberer and ends either with recollective experience, the rememberer's private awareness of the event on a subsequent occasion, or with memory performance, the overt expression of the recollective experience" [Tul83, p. 134].

The GAPS is decomposed into an arbitrary number of thirteen elements, which are termed as the elements of episodic memory (figure 2.3). The elements are grouped by three groups: observables (observable entities), processes (hypothetical processes), and states. Each element has a connection to another element. Such a relation can be called "has an effect on", "influences" or "brings about".

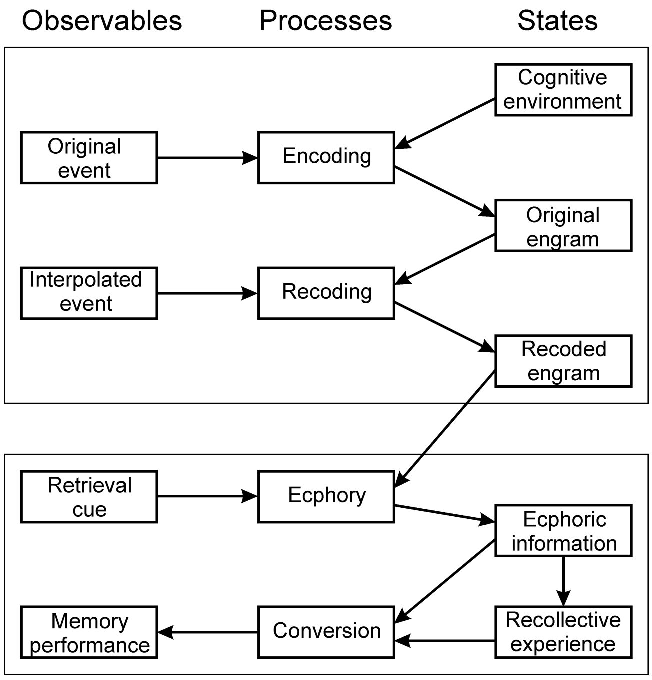

Figure 2.3: Elements of Episodic Memory [Tul83, p. 135]

The upper box in figure 2.3 represents the elements of *encoding*, the lower box represents the elements of *retrieval*. Encoding in the GAPS is described by the two processes "encoding" and "recoding". The "original event" is encoded—influenced by the cognitive environment—to an "original engram". This original engram can be recoded due to the occurrence of an "interpolated event" (a similar event to the original one). This recoding yields to an adapted "recoded engram". Retrieval plays a crucial role in modeling the act-of-remembering. It consists of the processes "ecphory" and optionally "conversion". As depicted in section 2.1.2, ecphory is a synergistic process that combines information from the "retrieval cue" (the query) and the original or recoded engram and results in "ecphoric information". Ecphoric information need not necessarily correspond to the information of the original or recoded event. This information can lead to "recollective experience" (a mental replay of the past event). The act-of-remembering which started with the perception and encoding of an original event usually ends with the retrieval and recollective experience. Optionally there may be another element of the retrieval process: "conversion". Conversion of information may lead to observable "memory performance" expressed by direct behavior: A person might take an umbrella to a trip because the last time the weather changed very fast and it started to rain.

The performance of the memory system heavily depends on the semantic knowledge represented by the formation of the retrieval cue and the kind and amount of stored engrams in the EM. The crucial part for the performance is the interaction between the engram and retrieval factors like the cue. The GAPS framework is not meant to be complete. It is a concept that attempts to decompose the act-of-remembering of a specific event. Other possible concepts might be restricted to only a sub-set of the mentioned elements. In all the imaginable schemes the two key processes encoding and retrieval play a crucial role.

2.1.4 Summary

Theories from human memory research of various neuro-psychoanalysts and psychologists are depicted in this section. Human memory is classified into short-term memory and long-term memory. The latter can further be categorized into the memory sub-systems semantic, procedural, and episodic memory. The distinction of the closely interacting episodic and semantic memory systems is regarded in more detail considering its similarities and differences. Characteristics of the episodic memory system are that it mainly registers perceptible changes, has affective components, and is context dependent. The prototypical units of information are events, which are stored temporally ordered and may further include reference to other related events. Human memory processing can be classified into the fundamental functionalities encoding, storage, and retrieval. Encoding is a procedure for transforming a perception into a memory. Storage describes the processes relevant for the retention of the stored information, like forgetting. The main determinants for the rate of forgetting are the time delay since the encoding of the event, the emotional involvement, the salience, and the frequency of rehearsals of the event. Retrieval determines, which memory will be remembered given a certain cue. A compatibility relation must exist between the retrieval cue and the stored event. The section is concluded with a conceptual framework for the study and understanding of episodic memory.

2.2 The ARS Project

The ongoing "Artificial Recognition System" (ARS) project at the Institute of Computer Technology in Vienna deals with situation recognition for ambient or robotic automation systems [Pra06]. The long-term goal is to build an intelligent autonomous system applicable for building automation that can deal with unforeseen situations in a reliable and flexible way. In future, more and more sensors will be integrated in building automation systems to control functionalities, to detect risks or to supervise the environment. Intelligent systems that are able to perceive the overall situation will be needed. Existing control mechanisms will not be capable to manage the demand of complexity in building automation. Most of them only act reflective on changes of sensor values and do not capture and act on the overall situation. The way, the human brain processes huge amounts of information is unique in its technique. To model human intelligence in a technical system, an approach has to be found that goes apart from classical control techniques. To do so, concepts of neurology and psychoanalysis are used.

The ARS-project is subdivided into two sub-projects: ARS-PC (Perception) and ARS-PA (Psychoanalysis). ARS-PC models human perception in a technical system. Assuming that in future building automation systems a huge amount of different sensors will be integrated, there is a need for new modeling of data processing. Nodes in sensor networks are capable to integrate sensors in large scale like light barriers, cameras, and microphones. What stands out is to process these data in an efficient way. Due to this large number of different sensors, some information is redundant. If a person enters a room, it may be captured at the same time by a camera, a light barrier, and a microphone. This huge amount of sensory information has to be processed in an efficient and reliable way, separating important from unimportant data. The human brain is one biological system that is able to cope with these demands. Inspired by human perception, perception in the ARS system is done by symbolization of information. A symbol is a piece of information that has a meaning and that can be processed by a system. The symbolization process is done in multiple levels where information is condensed more and more from the lower levels to the higher [Rus03] (figure 2.4). Low-level symbols (micro symbols) are concentrated sensor information whereas symbols at the highest levels (representation symbols) have incorporated world knowledge. So the main task of the perception module is to filter out redundant information and to reduce the amount of data in quantity having higher quality [Fue03].

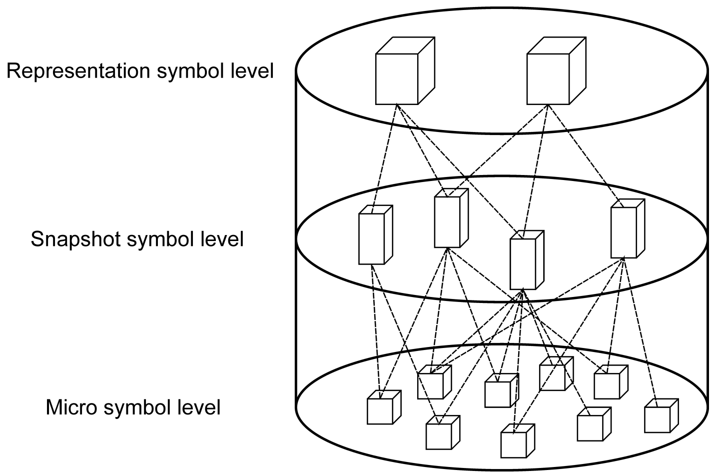

Figure 2.4: Symbol levels in the ARS system [DLP+06]

ARS-PA deals with high-level reasoning. It is a bionic approach to autonomous decision making [DLP+06]. Grounding on the task of perceiving information, the task of such an autonomous system is to make decisions based upon the perceived situations without external help. Beside the perception via sensors, actuators are needed to interact with the environment. These requirements to the system come very close to the definition of an autonomous agent:

> **Autonomous agent:** "An autonomous agent is a system situated within and a part of an environment and acts on it, over time, in pursuit of its agenda and so as to effect what it senses in the future" [FG97].

Thus, the ARS-PA system can be termed as an autonomous agent that is embedded within its environment. It senses the environment by various sensors and acts on it by a number of actuators. To select an appropriate behavior within a certain number of possible behaviors, the system has to be able to evaluate the currently perceived situation together with the executable actions. Therefore, an evaluation system has to be introduced. In biology, emotions can be seen as such an evaluation system. Humans rate perceived situations and their behavior with emotions; in its simplest way whether it is good or bad. The main task of the psychoanalysis part of ARS is therefore to implement a concept of emotions together with other concepts of psychoanalysis and to develop a technical system that aids an autonomous system in decision making. ARS-PA researches for a psychoanalytical model that is based on Sigmund Freud's Ego-Superego-Id personality model.

The psychoanalytical model is detailed in the first part of this section. In the progress this work, the psychoanalytical model has been enhanced with an episodic memory, that captures all experiences that the system has lived so far. With this EM the system is aided in decision making to act in a more intelligent manner. The interface to the EM is defined in the second part of this section. To decouple the preprocessing of the sensor values from the PA model, a simulation environment is used for developing and testing. This simulation environment is introduced in the third part of the section.

2.2.1 The Psychoanalytical Model

The central part of building an autonomous system is the creation of a general control architecture. The approach used in the ARS project is inspired by Neuro-Psychoanalysis. While neurosciences deals with the explanation of the brain bottom-up, psychoanalysis does so top-down. A fruitful exchange between the two research areas has started in recent years. The work of Solms [ST02], which represents an approach to combine neurosciences with psychoanalytical concepts, is the basis for the design of the model of emotional situation evaluation. The control architecture is hierarchical and includes both low-level and high-level forms of reasoning. The psychoanalytical model of the human mind integrates the combination of the various levels [RLVF06]. There are two sources of information for situation evaluation:

External stimuli — They cover all the sensory information that is streaming from outside into the system. In order to be manageable, the vast amount of raw sensor data has to be reduced dramatically. This is done in a multi-level symbolization process, where information is condensed in a hierarchically organized matrix of symbols. Symbols of the lowest level represent fragments of perceptions (edges,...) whereas higher level symbols represent more complex features which are derived from various channels associated with each other. As a result, the sensory information is transformed into "images" which represent the outside world.

Internal stimuli — They indicate the actual physiological state as given for example by internal variables like the energy level or the health state and the current levels of "drives" and "basic emotions". Drives represent the internal state. If there is an imbalance—e.g. an internal variable gets out of its range—the drives are affected and may further initiate active search behavior. (i.e. the energy level falls below a certain threshold, the drive hunger is affected which further leads to a *search-for-food* behavior). The most important purpose of the basic emotions is to evaluate situations whether they are desirable or harmful. Beside this, basic emotions like fear or anger are also linked to specific bodily reactions and provide fast, stereotype responses to certain categories of input situations.

The architecture that tries to combine the neuro-psychoanalytical view for autonomous, emotionally supported reasoning systems is presented in figure 2.5. It is a symbol processing decision unit based on Sigmund Freud's Ego-Superego-Id personality model, drives and emotions. The functional units in figure 2.5 are illustrated as rectangular blocks, the arrows indicate

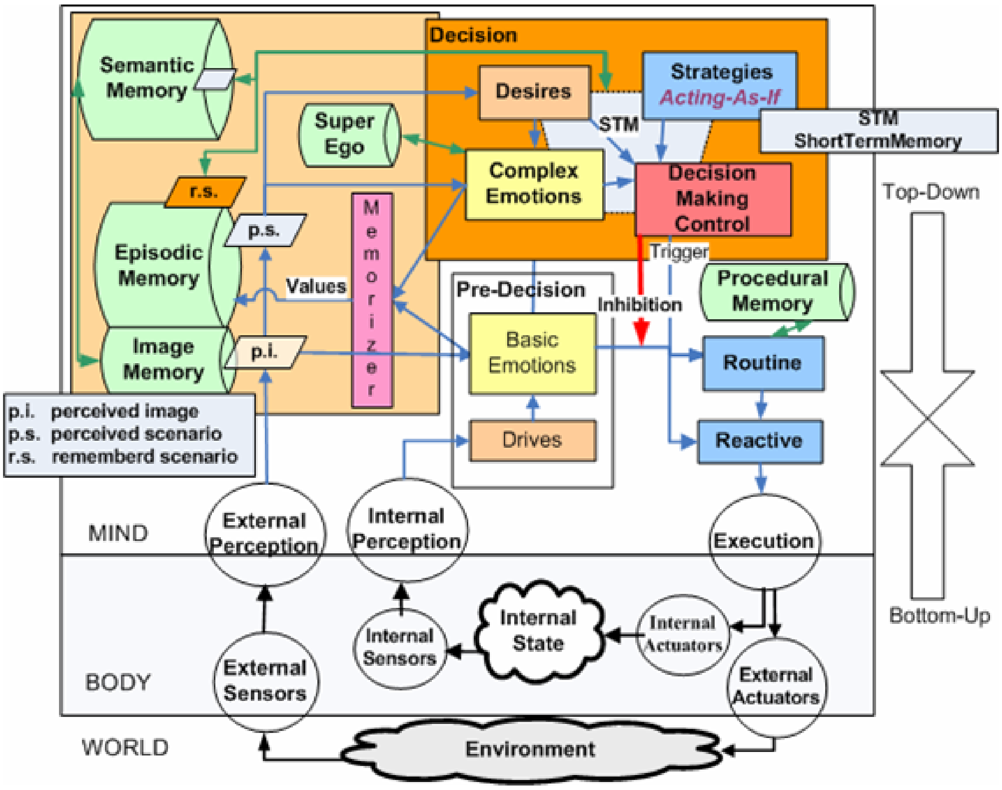

Figure 2.5: Functional blocks of the ARS-PA model [DLP+06]

informational and/or control flows. The functional modules make use of different memory systems to fulfill their functions. The key ideas of the neuro-psychoanalytical picture implemented by that architecture are that human intelligence is based on a mixture of low-level and high-level mechanisms and the usage of emotions as evaluation mechanism on all levels of the architecture [RLVF06]. Low-level reasoning is quick and makes the agents available with a basic mode of functioning. It provides reactive, relatively pre-defined responses but it may not always be accurate. It is based on the actual levels of the drives and basic emotions. High-level emotional decision making needs strategic planning and leads to behaviors that are more sophisticated. Emotions enable the agents to evaluate all the information they acquire both on the low-level (basic emotions) and on the high-level of the architecture (complex emotions). The episodic memory containing emotionally rated previous experiences is an important feature for evaluating new situations.

The behavior selection process runs as a loop and starts with the perception of a situation. Internal perceptions originating from the internal state of the agent are handed over to the drives module. External perceptions which originate from stimulations of the environment are extracted in the end of the symbolization process into symbols using knowledge stored in the image and semantic memory. The created images represent the currently perceived situation. This representation is passed on to the basic emotions unit. The basic emotions are affected from external perceptions and from the drives which are further influenced by the internal perceptions. Low-level decision making is done in the pre-decision module which consists of the drives and the basic emotions modules. If a drive is very high, the system will try to bring the drive back into its balanced range (e.g. if the drive "hunger" is high, the system will set actions like search-for-food to satisfy the drive hunger). Something similar happens with the basic emotions—each basic emotion is connected with a specific behavior tendency like for example *fleeing* for the emotion fear. An important task is further to rate the performed behavior based on the perceived consequences (mainly on the internal state). For example, if the system does not successfully accomplish a certain behavior (e.g. it cannot find an energy source when it is searching for food), this behavior may be rated by an increasing level of the emotion anger.

Perceived situations are further handed over to the high-level decision making (decision module). The complex emotions unit performs a rating by matching the current situation with more social emotions like shame, compassion, and the like. Therefore it interacts with the episodic memory by searching for similar situations to the current one including their emotional rating. A behavior may be triggered directly by a complex emotion. Additionally, the strategies module may plan and decide the executed behavior. Finally, actions are prepared via the reactive or routine unit and are then executed. Reactive responses directly arise from the pre-decision unit whereas routines are longer sequences of actions stored in the procedural memory.

2.2.2 Simulation Environment

To evaluate the psychoanalytical model during development, a hard wired installation equipped with a large number of sensors making available a variety of possible behaviors would be too costly. Thus, a simulation environment called the "Bubble Family Game" (BFG) (figure 2.6) has been designed [RLVF06]. It contains robot like autonomous agents called "bubbles" that may influence a number of objects. Objects within the bounded environment are energy sources, obstacles and other agents. Some of the energy sources are consumable by one single bubble, others are too large to be consumed by one single bubble—the help of other bubbles is needed. If a single bubble wants to consume the latter type of energy source, it needs the help of another bubble (e.g. the large energy source represents an elephant that can only be hunted by more than one bubbles). Obstacles are objects within the environment where the bubbles cannot pass through. Each bubble is equipped with various sensors to perceive the environment (external perception) and its internal state (internal perception).

Vision — To visually detect the environment in front of the agent within a cone-shaped corridor

Smell — To "smell" the presence of other agents within a radius

Acoustics — To receive commandos from other bubbles

Bump — To detect collisions with other objects (other bubbles, obstacles,...)

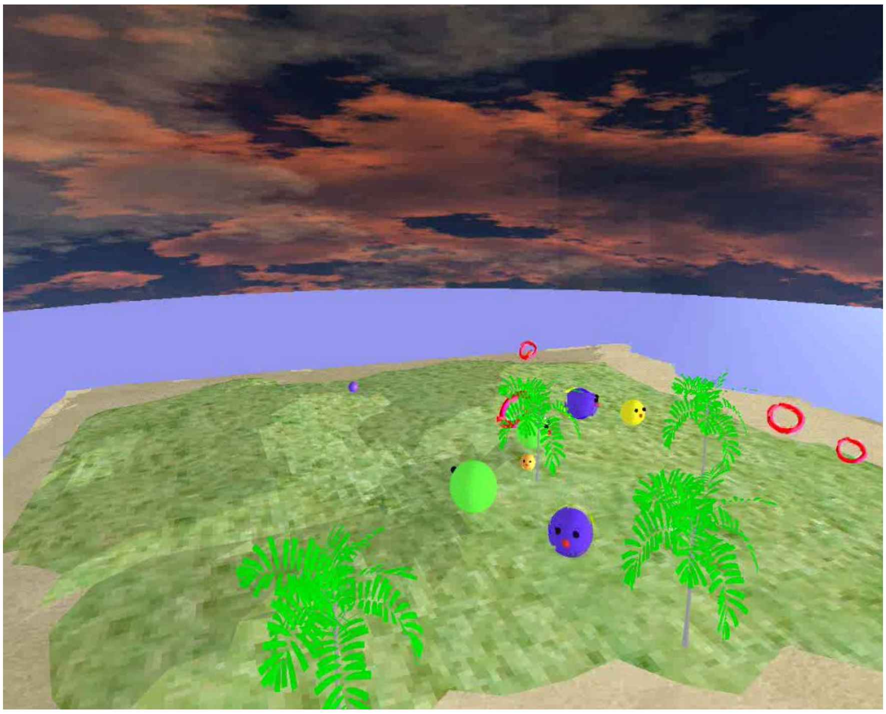

Figure 2.6: The Bubble Family Game

Odometry — To perceive its own relative position and movements

Health state — To detect its internal state

A bubble has one internal variable, the energy level. The task of each bubble is to keep its internal state, i.e. its energy level, within homeostasis. To stay "alive", the bubbles navigate in the environment in order to search for energy sources and consume it. If the energy level falls below a certain threshold, the bubble "dies". Different behaviors need different amounts of energy, so the level of consumptions depends on the level of activity. The total energy (bubbles and environment) is limited. If a bubble consumes an energy source, the level of the energy source decreases and is later reproduced with a certain rate. Possible behaviors might be *promenade, escape, search_for_food*, or *eat_food*. The individual bubbles have different values of emotional parameters, memory entries, action patterns, or different implemented behaviors. Bubbles may be facilitated with the emotional behavior arbitration model or with a rule-based expert system. The latter incorporates a few behavior schemes which are chosen by simple rules taking account the internal state (the energy level) and external influences. Two teams of bubbles act competitively to gain advantage over the energy sources. Each team has to cooperate with the other agents of its team to survive in the environment. The simulation ends when only one team has survived.

2.2.3 Interface Episodic Memory

The episodic memory module of the emotional behavior arbitration model captures, manages, and appropriately returns the experiences an agent has lived so far. It consists of all previously experienced episodes attached with an emotional rating. An episode is a sequence of situations

[RLVF06]. As depicted in figure 2.5 the functional units access the memory units via the memorizer module. Whereas the image memory is mainly used by the perception module, and the semantic memory—which contains knowledge describing the associations between the images—is mainly accessed during the symbolization process, the episodic memory is used by the decision unit. While the content of the semantic and image memory is general, the content of the EM is more agent specific. The experiences of the individual agents are specific to them self. Each bubble gathers its own autobiographical memories. The input to the EM includes all data referring to the currently perceived situation. The EM has to process the continuously delivered data and has to decide, when the agent has experienced something. The data handed over to the EM module are:

Drives — A list of the drives and their levels

Emotions — A list of the basic emotions and their levels

Template image matches — A list of all known template images and their match to the current situation

Actions — A list of the actions that are currently executed

Scenarios — A list of the momentary scenarios in progress

The drives represent the internal state. Drives are affected on the basis of thresholds. If an internal variable gets out of its range, the corresponding drive is influenced (e.g. if the energy level falls below a certain threshold, the drive hunger is affected). The drives in the current implementation are *hunger, seek*, and *play.* The basic emotions are influenced by the external perception and the drives. They reflect how the agent perceives the current situation. They represent an evaluation of the currently externally and internally perceived situation. The basic emotions in the current implementation are *fear, anger*, and *lust.* The template images (TI) represent extractions of perceived situations. They are predefined and stored in the image memory (figure 2.5). The composition of all known (stored) template images represent an implicit knowledge base of an agent about the objects in the environment. A single TI consists of condensed information from the external perception and may further include information from the internal perception (a specific object may always be perceived related with a particular emotion). When gathering a new perception via the sensors it is compared with each TI from the image memory and each TI gets a match corresponding to its similarity to the current perception. Consider the following example: Humans know how a fridge looks like. We have an internal image of a fridge. When we look at a real fridge, we process the visual perception (the perception of a fridge may further be influenced by a high level of the drive hunger), compare it to our internal (template) image of a fridge and might say: this object is to 90 percent a fridge. Thus, a gathered composition of all TI-matches represents a kind of snapshot of the current environment perceived by the agent.

A scenario is a stored template of a sequence of states. It consists of states and transitions (figure 2.7) and represents a specific pattern of activity (e.g. cooperating for food). Transitions are triggered by template images—i.e. to reach the next state, a specific TI match must fulfill a certain condition, for example exceeding a certain threshold. The scenario recognition processes are continually handed over to the EM. A scenario recognition process is a process, within which a scenario is in progress. It is indicated to the EM, when a new scenario recognition process is initialized (the first transition is triggered), recognized (the final

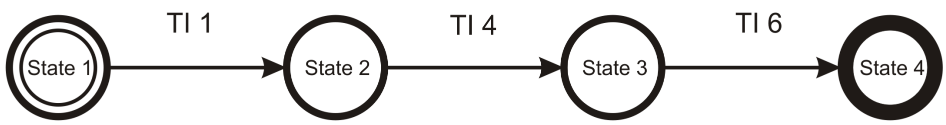

Figure 2.7: Scenario template

state is reached) or aborted (the next state is not reached due to specific abort conditions). Scenario recognition processes may run parallel—for instance, two or more scenarios of the same or different type may be in progress simultaneously. The list of actions contains all actions that are currently executed. An agent might execute two or more actions simultaneously. The action list represents the currently executed behavior. Possible actions might be: *promenade, move-to-food, eat-food, attack, flee, dance, and the like.* This data is handed over continuously to the EM module. The tasks of the EM module are:

- To detect and capture, when the agent has experienced something
- To manage the stored experiences
- To return an appropriate experience given a certain cue

While the input to the EM module covers all data about the currently perceived situation, the output of the EM module is an appropriate stored experience that is evoked either spontaneous or by stimulating externally with a certain cue (figure 2.8). Thus, memories may be recalled unbidden (see section 2.1.2) or as a result of a stimulus formed by the agent. When retrieving deliberate, the retrieval cue—the query to retrieve memories—can be formed arbitrary (e.g. cuing for a specific emotion and a particular executed action).

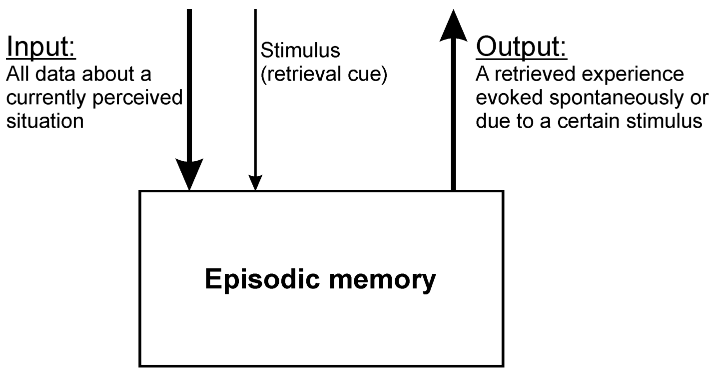

Figure 2.8: Interface episodic memory

2.2.4 Summary

The ARS project is a bionic approach to build an autonomous system that deals with situation recognition for ambient or robotic automation systems. While ARS-PC (Perception) models human perception in a technical system, ARS-PA (Psychoanalysis) develops a psychoanalytical model for decision making. The task of this project is the creation of a general control architecture for an autonomous, emotionally supported reasoning system. The episodic memory module presented in this work supports this system in decision making on the basis of past experiences. The ARS-PA system is evaluated in a simulation environment called the "Bubble Family Game", where autonomous agents navigate in a virtual environment in order to find and consume energy.

Chapter 3

Memory Model Architecture

The main task of the episodic memory (EM) module of the emotional arbitration model is to equip the self-acting agents with the ability of accessing past experiences. On the basis of these agent-specific memories, the agents are able to anticipate future situations and to evaluate the impact of certain behaviors. Inspired by theories of of human episodic memory research (section 2.1), an EM system is designed that fits into the psychoanalytical model of the ARS-project (section 2.2). The intent of the work is not to rebuild human EM, but to construct a technical system that coincides with the most important characteristics of human EM processing. The proposed memory model architecture is based on Tulving's General Abstract Processing System (section 2.1.3)—a conceptual framework for the study and understanding of human EM.

The chapter is outlined by a brief overview of the memory capabilities and the memory organization in the first section. This introduction follows the detailed description of the functional stages of human memory processing which can mainly be divided into encoding, storage, and retrieval. In the second section, aspects of encoding episodic memories are considered. Encoding refers to the processes that determine when an experience has to be encoded and what information has to be stored about an experience. The third section deals with the storage of episodic memories. Storage covers all processes that have to do with the maintenance of the stored experiences and how they change over time. Special considerations are given to the dynamics of memories, called "forgetting". Finally, episodic memories are retrieved and recalled. The processes describing the retrieval of an experience are issue of the final section of this chapter. The determination of when a retrieval is initiated, how a stimulus is constructed to query an experience, and which experience is recalled based on this stimulus are the tasks of retrieval.

3.1 Overview

The three fundamental functionalities of the EM system—as for each memory system—are encoding, storage, and retrieval. A memory has to be captured, it has to be stored and maintained, and it has to be returned given an appropriate retrieval cue. These functionalities have to be implemented in an efficient manner. The EM module has to be accurate—to return the right experience according a particular stimulus—and scalable—the performance should not decrease with an increasing number of stored memories.

In the first part of this section, some requirements of the EM module are addressed. The rudimentary capability is to be able to recognize and recall past situations. More sophisticated requirements of the EM module are to reconstruct longer sequences of situations (episodes) and to anticipate future situations and their outcome. In the second part of this section, the memory organization is outlined. On the basis of the definition of a situation, events are defined as synonyms for experiences. Furthermore, episodes are introduced as series of events that occur in a particular context (scenario).

3.1.1 Memory Capabilities

The rudimentary capability the EM module has to possess is to recollect previously experienced situations. The autonomous agent should be able to recognize similar stored situations to the currently perceived situation. Memories should pop up without the explicit effort of the agent. Furthermore, previously experienced situations have to be accessible at any moment for deliberate retrieval given a particular stimulus. The stimulus need not to be formed by the complete information of the respective situation—retrieval should be possible with only partial information indicated in the query. If the retrieved situation occurred in a specific context, the whole happening should further be accessible for recall. For example, if a past experience is recalled where a bottle of milk fell down on the floor and the retrieved situation occurred while making coffee, the entire episode when making coffee should also be retrievable.

Based on this rudimentary capability, sequences of situations that occurred after the retrieved situation should be recallable. Answers to questions like "what happened after that situation" should be provided. The agent should be capable to anticipate future situations. Possible impacts to which the currently executed behavior originating from the current situation may lead can be foreseen. With this anticipative capability, the agent is able to evaluate various behaviors between which it is indifferent to execute. The impact of each possible behavior is evaluated and decisions can be made based upon these evaluations. The behavior that leads to the desired goal state or that proposes the best result (e.g. evaluated by emotions) will be selected. For example, an agent stands in front of an energy source and wants to consume it. But to consume it, it has to fight against an agent from an opponent team, or has to flee and search for another energy source. Therefore it looks up in the memory and retrieves an experience where it fought against this bubble, lost, and felt very bad afterward. In this case it will decide to flee.

The episodic memory should further be used to reconstruct past episodes. For example, the agent wants to reach a specific goal originating from the current situation. Thus, it looks up in the EM for an episode that happened in a similar context like the current happening fulfilling a certain desired goal. The remembered episode with the best impact referring to the currently desired goal will then be reconstructed. For example, episode 1 and episode 2 in figure 3.1 represent two remembered episodes, which happened in the same context[1]. Episode 1 did not reach the desired goal whereas the goal of episode 2 was accomplished. Thus it is searched for a similar situation to the current one in the remembered episode in which the desired goal was successfully reached (episode 2). Originating from this situation in the remembered episode, the episode is reconstructed until the target situation with the

[1]Two episodes occur in a same context, if the happening was similar—e.g. in both episodes the agent was consuming food.

accomplishment of the goal is reached. An action plan including the actions that have been executed in the past episode is made, to trace back step-by-step the specific situations between the similar situation and the target situation in the past episode. The currently desired goal can thus be reached again.

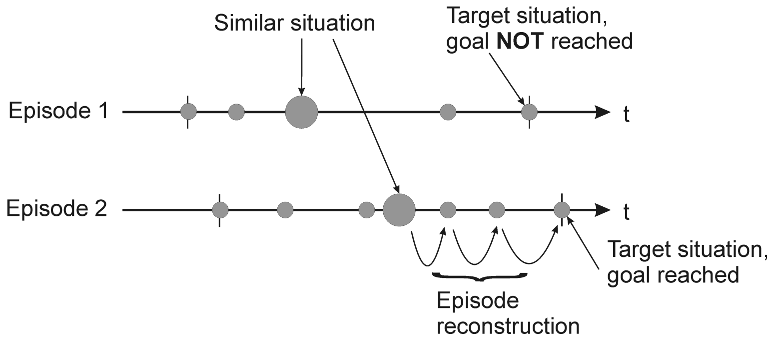

Figure 3.1: Episode reconstruction to reach a specific goal

Consider therefore the following example. You are very tired and want to make coffee, but the milk is empty. Your desired goal is to drink coffee with milk to become awake. Thus, you try to resolve that problem by remembering whether you were confronted to a similar situation in the past that you have resolved successfully. Two memories pop up that let you remember two episodes when making coffee and the milk was empty. Once you tried to go the supermarket and buy some milk, but it was closed and so you did not reach your goal. The other time you went to the neighbor and asked him if he gives you some milk. He handed over some milk and you reached your target of drinking coffee and being awake. Thus you try to reconstruct this specific previously experienced episode. You go again to the neighbor to ask for some milk and the desired goal may be reached again.

3.1.2 Memory Organization

According to section 2.2.3, the data continuously delivered to the EM module is composed by *drives, emotions, template image matches, actions*, and *scenarios*. On the basis of this data the EM module captures, stores, and returns experiences appropriately. It consists of emotionally rated episodes, which are sequences of situations. First of all it has to be declared on the basis of the gathered information how a situation is composed of. A situation in the EM module is defined as

> **Situation:** A composition of all information that characterize the momentary state of an agent.

The momentary state of an agent is made up by the currently perceived internal state, the currently perceived external environment, and the influence the agent has on the environment. According to this definition a situation is composed of:

Drives — Representing the internal perception

Template image-matches — Representing the external perception

Emotions — Representing an evaluation of the external and internal perception

Actions — Representing the current behavior of the agent

These four items make up a situation and are further denoted as the *features* of a situation. A situation is a representation of the agent's state at a particular moment in time. Theoretically, the time gap between two situations is infinitely small—situations are continuously streaming to the EM module. As the EM covers all experiences an agent has lived so far, a situation as defined here cannot be identical with an experience. Tulving states, that the prototypical unit in EM is an *event*. As regarded in section 2.1.2, Tulving uses an event as a close synonym for an episode, and defines an event as "something that occurs in a particular situation" [Tul83, p. 37]. It always has a beginning and an end in time, which may be so close to one another that we think of the event as "instantaneous" [Tul83, p. 142]. Tulving further distinguishes between *simple events* and *complex events* (section 2.1.1). While simple events are particular changes in the individuals perceptual environment, like seeing a flash—complex events usually last longer in time. EM further only register changes. If some information already exists, it is not stored again (section 2.1.1). Regarding to these facts, an event in the EM module is defined as:

> **Event:** A happening, that arises at a particular time when something in a situation changes significantly.

An event arises at a particular time. The beginning and the end of an event come together and an event as regarded here occurs simultaneously. This definition refers to Tulving's simple events. An event is the prototypical form of an experience in the EM module. A situation becomes an event when something in the situation changes significantly. An event is an extended situation. Additionally to the static data of a situation, an event contains information like the salient happening within a situation. According to Tulving's decomposition of an event into a setting and a focal element (section 2.1.2), the static data of a situation refers to the setting of an event, whereas the additional information about the salient happening corresponds to the focal element. The EM module monitors the happenings within the stream of perceived situations and to detect significant changes in the features of a situation that cause the encoding of an experience. The emotional involvement of the agent is further attached to the event. Events represent the basic memory units and are stored chronologically ordered. Events may also have semantic contents. The template images—the implicit knowledge about objects of the world—provide some semantic knowledge, especially if there were relations among them. In the current implementation of the emotional arbitration model, semantic memories in the kind relations between template images are not supported. According to Tulving's statement that an episode can be connotated as a series of events (section 2.1.2), an episode in the emotional behavior arbitration model is defined as:

> **Episode:** A temporally ordered sequence of events.

This definition refers to Tulving's complex events (section 2.1.1). An episode in the EM module is an extended form of an experience and is lasting longer in time. An episode always comes along with a recognized scenario, which represents the context in which the episode happens. Scenario templates are a kind of semantic memories. They represent semantic knowledge, that is stored in the form of scripts (see section 2.1.1). These scripts that represent commonly social events such as make coffee or going to a restaurant might be learned with the repeatedly interaction with the world. Having such a basis of scripts, specific experiences are stored and remembered based upon these scripts [Bad97, p. 240-249].

An episode is a concrete realization of a specific, predefined scenario template. It covers the series of events that are captured from the initialization of a scenario (the triggering of the first transition) until the completion of the scenario. The states, by which a scenario template is characterized, are represented by the events, within which the corresponding transition was triggered. In figure 3.2 are illustrated two realizations of *scenario A* by *episode*

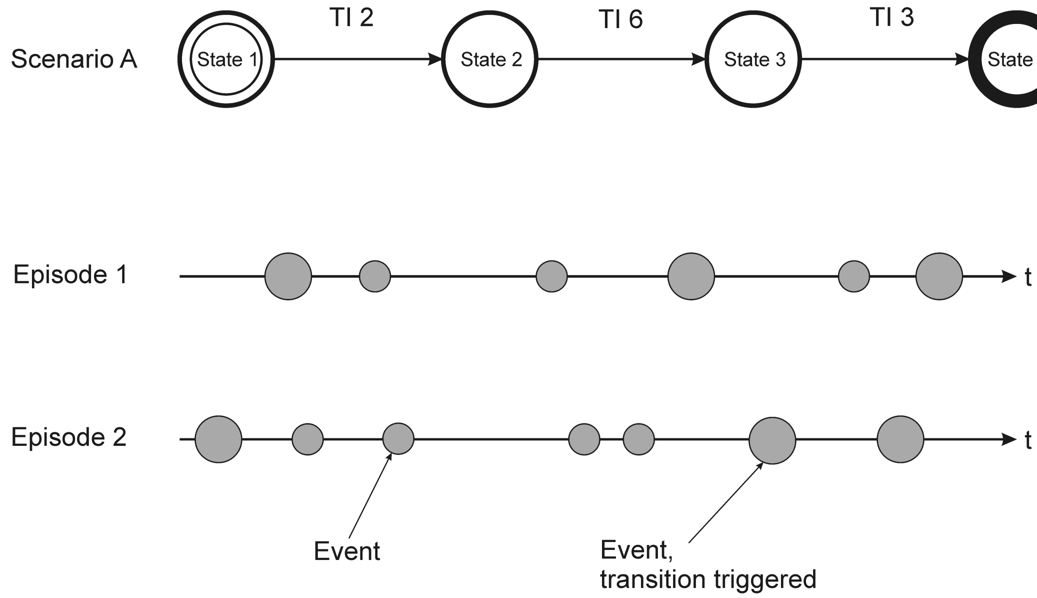

Figure 3.2: Episodes as realizations of a scenario template. *Episode 1* and *episode 2* both happen in the context of *scenario A*.

1 and *episode 2*. Scenario A consists of four states. The scenario get into the succeeding state, when the corresponding template image (TI) fulfills a certain condition (e.g. scenario A gets from state 2 into state 3 if the condition for TI 6 is fulfilled). The events within which a transition condition has triggered are illustrated by big circles. These events are similar in each episode of this scenario (corresponding to the particular TI), and indicate when the scenario has skipped into its next state. The small circles represent additional events that occur within the completion of the scenario. The occurrence of these events makes the individual episodes to one particular scenario template unique. An episode of a scenario template is thus characterized by its individual composition of events. While the scenario template an episode is a realization of represents the *context* of an episode, the sequence of events within the realization of an scenario is termed as the *content* of an episode. Additionally the emotional state at the beginning and the end of an episode is attached. This is termed as the *impact* of an episode. The impact represents the effect the episode caused to the agent. For each scenario might exist various realizations in the form of episodes. The realizations of episodes which happened in the same context (they are realizations of the same scenario template), differ in terms of the particular series of events, their emotional impact, and in the time they happened. Episodes of the same or different scenario template may overlap or be nested in time—they may occur parallel.

Consider the following example: we have a lot of scenario templates in our mind (e.g. the scenario *make coffee*). Today I have already realized this scenario two times: in the morning in my house and in the afternoon at the university. These episodes are different in terms of their concrete realization by the specific sequence of events, but both have in common the same characteristics defined in the scenario template (like enter the kitchen, fill in water and coffee, start the coffee machine, and the like). I might remember these episodes in the evening when I open the fridge and see the milk (which is an event of the episode make coffee). The episode *make coffee* realized in the afternoon may be further nested in time into the overall episode *go to university*— which is a concrete realization of the scenario *go to university*.

3.2 Encoding

Encoding describes the process of transforming an original event into a memory trace or an engram (section 2.1.2). The task of encoding is therefore to determine

- When is encoding initiated?
- What information is stored?
- What features make up an event?
- What happens when a similar event occurs?

One of the most crucial parts in the process of encoding is to determine the particular moment when it occurs. According to Tulving, the individual is not aware when the process is initiated. This issue is concerned in the first part of this section. Once the encoding process is triggered, it has to be declared what information is stored, what features make up an event. This topic is concerned in the second part of this section. The occurrence of a similar event to a still existing may include that the similar events get references to each other or that the existing one is modified. This issue is addressed in the final part of this section.

3.2.1 Triggering

The episodic memory consists of all experiences an entity has accumulated in his life. In section 3.1.2 an event has been declared as a prototypical experience in the EM module and was further defined as “a happening, that arises at a particular time, when something in a situation changes significantly”. The task of triggering the encoding of an event is to determine when an event should be recorded—i.e. when the agent has experienced something. According to Tulving, the individual is not aware, when encoding is initialized and it happens frequently (section 2.1.2). The system has to monitor all the time everything that happens to the agent and has to recognize, when an event occurs, i.e. when something in a situation changes significantly. A situation is composed of the features drives, emotions, template image-matches, and actions (section 3.1.2). The encoding of an event should be triggered when any of these features suffers a significant change, in other words, when something happened to the agent in a situation.

The concept to determine the moment, when something happened to the agent, is inspired by the regulation mechanisms of the internal environment—frequently termed as "homeostasis"—described by Damasio [Dam03]. The adoption of the levels of drives and emotions are the result of homeostatic processes. The brain responds with some regulatory reactions, when their levels get out of balance caused by a certain object or situation. The causative object for drives, like the drive *hunger*, is usually internal—a reduction in the availability of something vital, like energy. The reaction might be a *search for food* behavior to bring back the drive into a balanced level. The adoption of the level of an emotion depends on a complicated chain of events. It begins with the appearance of an emotionally competent stimulus—a certain object or situation actually present [Dam03, p. 57]. The objects that influence the emotional levels are usually external (e.g. kinds of physical stimuli in the environment). Despite that, emotional levels may also be affected due to interactions with drives. Thus, it can be deduced, that something happened to the agent, when the levels of drives or emotions get out of a certain range. The level of the respective drive or emotion is affected due to the appearance of an event, and the regulatory reactions try to bring them back into a balanced range (homeostasis) [Dam03, p. 49-51].

Beside the detection of an event through the drives and emotions via the internal perception, an event should further be recorded, when something significant changes in the external perception or due to a change of the agent's behavior. To summarize, something may happen to the agent

Internal — Through a change of the drives and basic emotions via the internal perception

External — Through a change of the template image-matches and basic emotions via the external perception

Behavioral — Through a change of the agent's currently executed actions

For example, an event should be recorded, when a specific action is set, or when the emotional activity is very high. In other words, a feature of a situation is activated, if it suffers a significant change. The definition of a significant change is different for each feature and is further considered.

Triggering on Emotions

As mentioned in the section 2.2.3, there exist three basic emotions in the emotional arbitration model: fear, anger and lust. To determine a significant change of a basic emotion, various approaches are summarized in table 3.1. The first approach triggers encoding, when the level of an emotion is beyond a certain threshold. For example, an event is encoded, when the agent is very angry (level of emotion anger is high). This may be plausible. Something happened that forced the emotion to get out of its balanced range. Further it is easy to be implemented. The level of the emotion has only to be checked to a certain value. A disadvantage of this approach is that it depends on a certain fixed threshold which is not variable. An event would be recorded if the level of an emotion is infinitely little above that threshold and it would not be recorded if the level is little below. Further, all emotional changes under the fixed threshold or above that threshold are not recognized, if the change does not cross the threshold. As emotions do not change abrupt but slowly, this might be a pretty poor approach for recognizing events.

The second approach triggers, if the change of an emotion to the level in the previous situation reaches a certain value. Something happens in a situation, that forces an emotion to a

Approach	Pros & Cons	
Trigger on threshold	Pro:	May be plausible Easy to implement
	Con:	Inaccurate due to fixed threshold Does not recognize anything when the threshold is not crossed
Trigger on changes to previous situation	Pro:	May be plausible
	Con:	Impact depends on current emotional level Refers to a situation unspecified in time Difficult to define change level
Trigger on changes to previous event	Pro:	Monitors change over longer time space
	Con:	A lot of time may be elapsed since last event

Table 3.1: Triggering on emotions

significant change. This happening comes very close to the definition of an event, so it would be a very plausible trigger mechanism. A disadvantage is that the impact of changes are treated independently on the absolute emotional level. A change when the emotional level is low has a different effect than the same change at a high emotional level. Furthermore, emotions change permanently, they rarely remain constant. So it is very different to define an absolute change value when an event should be triggered. Another disadvantage is that only changes to the previous situation are considered. As there is no definition of time within the emotional arbitration model, a change to an unspecified situation in time is not very meaningful. Slower changes lasting over longer time spaces are not considered.

The third approach triggers if the change to the previous event of that type (i.e. for the emotion anger the previous event where anger triggered the encoding of an event) is above a certain value. In contrast to the second approach, slower changes of an emotion over longer time spaces are monitored. This is both an advantage and a disadvantage—if the level of the corresponding emotion changes very slow, a lot of time may be elapsed since the previous occurrence of an event and triggering may therefore be nonsense.

The proposed concept is a combination of the three approaches. The encoding of an event should be triggered depending on both, the change to the previous situation and on the absolute level of the emotion. If the absolute level is very low, even if there is a change to the previous situation, the agent does not care the situation and would not record it. The higher the absolute emotional level and the higher the change of this emotion to the previous situation, the more likely that the current situation has an effect on the agent, and is thus recorded. On the other hand side, a high absolute emotional level and no change to the previous situation would not lead the the encoding of a situation. The situation has no impact to the agent and as the EM system only register changes, the same information is not entered again if it still exists [Tul83, p. 37]. The relevance of an emotion to the current situation depending on the absolute emotional level and the change is measured by a salience, which is considered in section 3.2.2. If the salience of an emotion to a certain situation is above a certain threshold, the situation will be encoded as an event. With this combination, the disadvantages referring to the triggering concepts depending merely on a fixed threshold or on a change to the previous situation are made weaker. Furthermore, slow emotional changes over longer periods of time should be recognized as well. This is done likewise as the third approach suggests, but with a small modification. Changes are as long tracked, as long as the change keeps strictly monotonically increasing. The situation is recorded as an event, depending again on the absolute emotional level and the change since the start of tracking, measured by a salience (see section 3.2.2).

These proposed concepts trigger encoding of an event, if any single emotion of the feature emotions suffers a significant change and is relevant to the current situation. This could be extended. The relevances of the individual emotions could be composed to a overall relevance of the feature emotions and an event is triggered based upon this overall relevance of the feature.

Triggering on Drives

Drives in the emotional arbitration model are hunger, seek and play (section 2.2.3). Drives represent the current internal state. They act on thresholds—i.e. if an internal variable like the energy level gets out its bound, the corresponding drive (hunger) is affected. If a drive gets out of a balanced range, some regulatory reactions are executed like a *search for food* behavior for the drive hunger, to bring the drive back into a balanced range [Dam03, p. 49]. Thus, it is proposed that drives are relevant both when they are very high or very low. Imagine the drive hunger. It is an unusual situation both when you are very hungry or if you eat too much and oversaturate. The relevance of a drive to situation again is measured by a salience (section 3.2.2). Based upon this dependency of drives upon thresholds, it is proposed that the encoding of an event should be released, when a drive exceeds a specific threshold. The concept might be compared with the approach considered when discussing triggering on thresholds of the feature emotions. Thus a significant change in the feature drives arises both when a drive *gets* very high or *gets* very low. Furthermore encoding should be triggered, when the drive *gets back* from this extraordinary range.

Triggering on Template Images

Template images (TI) are predefined images and represent a part of the knowledge base of the agent. Each perception is compared to the existent template images and each TI gets a match. So, the composition of the TI-matches represents a kind of snapshot of the perceived environment (the external perception) by the agent. To trigger an event, these matches have to be observed and a significant change in the composition of the individual TI-matches has to be detected. The concept of triggering an event on the basis of TI-matches is very similar to the concept proposed for emotions. The approaches are summarized in the table 3.2.

Triggering an event on a threshold of a TI-match encodes an event, if the match of a specific TI exceeds a certain threshold or goes below this threshold. This concept is also known as triggering "on change event". Lets assume, the agent has detected an object in the simulation environment, for example an energy source, the match for this specific TI exceeds a predefined threshold. This object appears in that particular situation and the agent has registered a significant change in its external perception. The encoding of an event is triggered. The

Approach	Pros & Cons	
Trigger on threshold	Pro:	Easy to implement
	Con:	Inaccurate due to fixed threshold Does not recognize anything when the threshold is not crossed
Trigger on change to previous situation	Pro:	Recognize changes in a situation
	Con:	Impact does not depend on absolute match level Refers to a situation unspecified in time
Trigger on change to previous event	Pro:	Monitors changes over longer time space
	Con:	A lot of time may be elapsed since last event
	Con:	May no be plausible for TI-matches

Table 3.2: Triggering on template images

event will be encoded as well, if the object disappears, i.e. the corresponding TI gets out of focus. This is very plausible: The composition of the highly activated TI-matches reflect the relevant features of a perceived situation. The disadvantage of this approach is similar to the one discussed with the emotions. The barrier, which decides whether an TI-match will be encoded or not is fixed. Once the threshold is crossed (upward or downward), changes will no more be recognized as long as the TI-match does not cross the barrier again.

The second proposed approach supposes to trigger an event if the difference of a TI-match to the match in the previous situation exceeds a certain value. The difference is calculated for each individual TI, and an event is triggered, if the change of any TI-match is high enough. With this concept, every change in a perception is detected. The disadvantages are that firstly it is very difficult to determine a fixed value, for which the triggering is forced. Further, this approach does not depend on the absolute TI-match. Even if the impact on the absolute TI-match is not that strong than with the emotions, it is supposed that the absolute level of an TI-match also has an impact on the relevance to the current situation. Secondly, the calculation of the difference to the previous situation refers to a situation that is unspecified in time in the model. The third approach triggers encoding, if the change to the previous event caused by this TI-match is above a certain value. Changes over longer time spaces could be monitored, but as TI-matches change more abrupt and not that smooth than emotions, monitoring the track of TI-matches over longer time spaces may not be meaningful.

Similar to the concept with emotions, the concept proposed as well is a combination of the absolute TI-match and the change of a match to the previous situation. The higher the absolute TI-match and the higher the change, the more relevant—in other words the more activated—the TI is to the current situation. No change at even a high absolute TI-match means that either the event might have been recorded before and still exists in the memory—the EM system only register changes—or that this TI has no relevance to the current situation. The relevance of a TI to the current situation again is measured by a salience (section 3.2.2).

These proposed concepts trigger encoding of an event, if any single TI-match of the feature TI-matches suffers a significant change and is relevant to the current situation. This could be extended, that the relevances of the individual TI-matches are composed to a overall relevance of the feature TI-matches.

Triggering on Actions

Actions represent the current behavior of the agent—the influence it actuates on its environment. The current behavior is composed by a set of actions, which are executed simultaneously at a particular moment. To trigger the encoding of an event on the basis of the agent's behavior, significant changes in the feature actions have to be detected. It is proposed that the encoding of an event is triggered, if the composition of the set of executed actions changes to the previous situation. That means, a change of the current behavior is recognized, if a new action is executed or a previously active action is no more executed. This is very plausible and easy to implement. Furthermore, encoding should also be triggered, if a specific action is accomplished. An action is accomplished, if it was performed successfully.

Episode Recognition

An episode is a sequence of events that represents a concrete realization of a scenario template (section 3.1.2). The scenario recognition processes that currently are in progress (initialized, aborted and recognized (see section 2.2.3)) are continuously handed over to the EM module. If a scenario is completely recognized, the series of events from the initialization of the scenario until its completion is recorded as an episode[2]. Each event that is part of the episode will be related to it. The series of events from the beginning to the end of the episode is attached as its *content*. The scenario template, the episode is a realization of, is associated to the episode as the *context* in which the episode happened. Scenario templates represent semantic memories—something that the system "knows". A memory is established durable, if it is associated meaningfully with knowledge that already exists. This refers to what Schacter terms as *elaborative encoding*: the integration of new information with something that is already known [Sch96]. The emotional state at the beginning and the end of the episode is further attached and denoted as the *impact* of the episode. The storage of the episodes is treated in more detail in section 3.3.

3.2.2 Salient Feature Selection

After discussing when an event should be encoded, the next step is to determine what features of a situation should be stored. Tulving states, that an event is recorded in terms of a salient happening. As stated in section 2.1.2, Tulving decomposes an event into a setting and a focal element. While the setting refers to the time and place in which the event occurs, the focal element describes the salient happening within the event (e.g. a picture against a background) [Tul83, p. 143]. Transferred to the EM module concerned in this work, the setting corresponds to the entire data of a situation, whereas the focal element refers to the part of the situation that is relevant. The task of the salient feature selection is thus to determine the features that are relevant in a particular situation.

Tulving further states that a cue for later retrieval of an encoded engram will only be effective, if its properties are similar to the properties of the trace of the encoded event (*encoding*

[2] The detailed mechanisms for recognizing episodes is considered in section 4.2.2

specificity principle, section 2.1.2)—there must be a compatibility relation between the retrieval cue and the engram. Retrieval aspects will be specified in more detail in section 3.4, but the importance of this compatibility relation should be highlighted and considered now. For what would an encoded event be for if it was not possible to retrieve it later? So it is very important to focus in the stage of encoding on making an engram as effective as possible for later retrieval. As Tulving suggests, there is a relationship between encoding and retrieval that has to be considered at the time of encoding. If a stimulus leads to the retrieval of an item, then it is assumed to have been encoded, whereas if not, then it is assumed not to have been encoded [Tul83, p. 264].

In contrast to Tulving's statement that not the complete event is recorded, the entire event with all features will be encoded in the EM module. This can be argued by the enormous capacity of the computer memory both in terms of quantity and detail. The size of an memory entry does not affect performance for later retrieval. Performance depends on the number of engrams, the retrieval cue, and the operations but not on the individual size of the memory entries. To come along with Tulving, that an event is recorded in terms of the salient happening (the focal element), a selection is made that highlights the relevance of specific features to the current situation with a salience value (the focal element of the event). The salience in the EM module is thus defined as:

> **Salience:** A measure for the relevance or the activation of a feature to a certain situation.

As depicted in section 3.1.2, a situation is composed by various features that represent the internal and external perception, its evaluation by the emotions and the currently executed behavior. The relevance of each feature to the current situation is measured by a salience value which is further the basis for the encoding of the situation as an event. The event based approaches for the triggering of encoding were discussed in section 3.2.1. Whether a situation should be recorded or not is determined by the salience of each feature which is further combined to the salience of an event. If the salience is high enough which means that there happened something significant in a situation, it will be recorded. Consider the following example. When you hear a very loud noise, the template image representing that perception would be very relevant in this situation, it is highly salient. Furthermore, you might be frightened due to this noise. Thus, the affected emotion is highly salient too. The situation will be recorded as an event and the corresponding template image and emotion are marked as the relevant features of this event that caused its encoding. Salient features are the significant, highly activated features, that make up an event. They represent the relevant part (the focal element) of an event. This concept realizes the implicit encoding of the cue information. Spontaneous retrieval—which is stimulated with the currently encoded event as cue (section 3.4)—will only check the salient part of an event to get a match.

The salience of a feature is deduced from the elements[3] of a feature. The relevance of each element of a feature is measured by a salience value at the moment of triggering encoding. So for each feature (drives, emotions, TI-matches, and actions) the salience is derived based upon the salience values of its consisting feature elements.

[3] Each feature is composed of various feature elements which represent the individual items the feature consists of. For example, the elements for the feature emotions are the individual emotions anger, fear and lust. Figure 3.4 details the composition of an event into features and feature elements.

Salience of emotions

The salience of an emotion measures the relevance of this emotion to the current situation. It evaluates, whether the emotion is part of the focal element of the encoded event. Perceived situations are evaluated by the emotions. As depicted in section 3.2.1, the level of an emotion may be influenced by the appearance of an emotionally competent stimulus—a certain object or situation actually present [Dam03, p. 57]. Thus, the change of the emotional level is one indicator to measure the relevance of an emotion to the current situation. The higher the change of the emotional level, the higher the effect the emotionally competent stimulus has on the emotion and thus the more relevant is this emotion to the happening in the current situation. Another indicator is the absolute level of the emotion. The higher the emotional level, the more the emotion is out of its balanced range, and thus the more the emotion is relevant in the current situation. Consider the emotion lust. Something happens—e.g. the agent consumes food—and the level of the emotion lust increases. The relevance of the emotion lust to the current situation is measured by its salience; the higher its change and the higher its absolute level, the higher the salience. If the salience exceeds a certain threshold, the situation will be recorded as an event.

The salience of an emotion i (e.g. emotion lust) S_{E_i} (equation 3.1) is determined by a function f dependent on the absolute level L_{E_i} of the emotion and a component representing the change of the emotional level to the previous situation C_{E_i}.

$$S_{E_i} = f(L_{E_i}, C_{E_i}) \tag{3.1}$$

with

$$0 \leq L_{E_i} \leq 1$$

$$0 \leq S_{E_i} \leq 1$$

Emotional levels (L_{E_i}) are defined in the psychoanalytical model to be in a range between zero and one. An emotion with the level one means that it is very high, whereas an emotional level close to zero indicates a very low arousal of this emotion. The dependency on both variables of equation 3.1 is proportional. The higher each variable, the higher the salience. The change component results in the absolute difference value of the emotional level of the current situation L_{E_i} to the level of the previous situation $L_{E_{i,prev}}$: $C_{E_i} = \left|L_{E_i} - L_{E_{i,prev}}\right|$. High changes of the emotional level result in a high salience level whereas if the level of the emotion remains constant, the salience is zero. If there is a high change of an emotion, something significant happened in that particular situation. The significance of the second component, the absolute level of the emotion is obvious. The higher the absolute level of an emotion, the higher the salience of this emotion. If the absolute value is very low, the salience of the emotion is consequently very low as well. To implement the proportional dependency of the two variables in equation 3.1, the absolute level L_{E_i} and the change component C_{E_i} are associated by a product (equation 3.2).

$$S_{E_i} \propto (L_{E0} + L_{E_i}) \times C_{E_i} \tag{3.2}$$

Both, positive and negative changes are considered. Negative changes are treated weaker than positive changes. The emotional levels decrease by itself because of the regulatory homeostatic processes (section 3.2.1). Such slow decays of emotional levels should not be considered. Thus, a bias value of L_{E0} is added for negative changes (equation 3.3). With this bias value, the component indicating the absolute emotional level is adapted to a value between the current level (L_{E_i}) and the level of the previous situation ($L_{E_{i,prev}}$).

$$L_{E0} \leftarrow \begin{cases} 0 & \text{for positive changes} \\ \frac{C_{E_i}}{2} & \text{for negative changes} \end{cases} \tag{3.3}$$

The relation in equation 3.2 is further normalized by an exponential function with the parameter S_{E0} (equation 3.4).

$$S_{E_i} = 1 - e^{-\frac{(L_{E0}+L_{E_i})\times C_{E_i}}{S_{E0}}} \tag{3.4}$$

This normalization is done to get a plausible salience value for the emotion between zero and one ($0 \leq S_{E_i} \leq 1$). Otherwise, the combination of L_{E_i} and C_{E_i} by a product would lead to very low salience values (emotional changes C_{E_i} are usually very small and both C_{E_i} and L_{E_i} are within a range of zero and one). An exponential function has been chosen to compensate this reverse exponential effect of multiplying values between zero and one (L_{E_i} and C_{E_i}). Because the salience of an emotion is further combined with the other emotions to a salience for the feature emotions and further with the other features to the overall salience of an encoded event, the individual values have to be "compatible" with each other—they have to mean the same. A salience of 0.6 of an emotion has to express the same relevance to a situation as a salience of 0.6 of a template image. Thus, the parameter S_{E0} has to be determined empirically.

Salience of Drives

The salience of an drive measures the relevance of this drive to the current situation, whether the drive is part of the focal element of the encoded event. The causative object for the adoption of the level of drives is usually internal (e.g. a reduction of something vital, like energy). If the affected drive gets out of a certain range, the brain reacts with some regulatory reactions to bring the drive back into a balanced range [Dam03, p. 49]. Thus, it is proposed, that a drive is relevant in a situation, if its level is out of a certain balanced range. A drive is salient when it is in an extraordinary state, both when its level is very high or very low. Consider the drive hunger. To a person, it is an unusual situation both when you are very hungry or very over saturated. Somebody might remember a situation, when he was in such an extraordinary situation. Thus, a drive is relevant and further salient in that particular situation when it is very high or very low. The proposed determination of the salience S_{D_i} ($0 \leq S_{D_i} \leq 1$) of a drive i dependent on the drive level L_{D_i} is illustrated in figure 3.3.

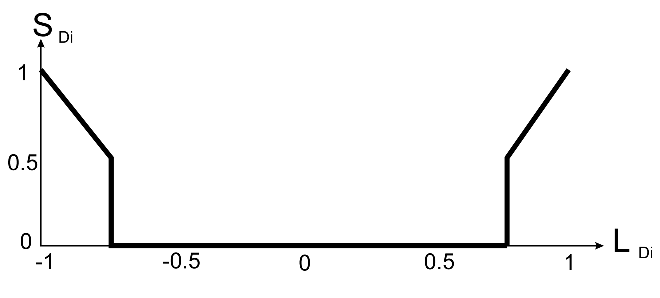

Figure 3.3: Determination of the salience S_{D_i} for a drive

Drive levels are defined in the psychoanalytical model to be in a range between minus one and one ($-1 \leq L_{D_i} \leq 1$). A drive with the level one means that it is very high (e.g. very hungry and appetite for food), whereas a negative drive level close to minus one indicates the opposite effect (e.g. over saturated and disgust for food). If the drive level (L_{D_i}) exceeds a certain threshold (either a positive or a negative threshold), it is to a certain amount salient. The salience is the higher, the higher the absolute value of L_{D_i}.

Salience of Template Images

The salience of a TI-match measures, whether the respective TI-match is relevant to the current situation and further part of the focal element of the encoded event. Template images represent a kind of snapshots of the perceived environment. Consider the example depicted above with a TI representing a loud noise. It is relevant in the particular situation when the noise arises (the TI gets a high match) and to a certain amount in the succeeding situations[4]. If the noise remains constant over a very long time, it is no more that relevant in the particular situations than it was before, even if the match of the TI remains high (the EM only registers changes). Thus, it can be deduced that the indicators for the relevance of a TI in a certain situation are on the one hand the absolute match of the TI (to which amount the TI "appears" in the situation), and on the other hand the change of the TI-match.

Based upon the event based triggering approaches discussed in section 3.2.1 (triggering on threshold and triggering on changes), the salience of the elements of the feature TI-matches is determined in a similar manner like the salience for emotions. The salience of an individual TI-match i S_{TI_i} is obtained by a function f dependent on the absolute level L_{TI_i} of the TI-match and a change component C_{TI_i} (equation 3.5).

$$S_{TI_i} = f(L_{TI_i}, C_{TI_i}) \tag{3.5}$$

with

$$0 \leq L_{TI_i} \leq 1$$
$$0 \leq S_{TI_i} \leq 1$$

The level of an TI-match as delivered to the EM module is between zero and one. A TI-match of one means that this TI totally matches to the currently perceived situation, whereas a TI-match of zero indicates that this TI does not appear in that situation. Similar as depicted before when deriving the salience of the emotions, both components contribute proportionally to the salience value. The higher the absolute level of the TI-match and the change of the match of an TI to the previous situation ($C_{TI_i} = \left| L_{TI_i} - L_{TI_{i,prev}} \right|$), the higher the salience. If the match of an TI remains constant, the salience is zero. This dependency is implemented by associating these two components by a product (equation 3.6).

$$S_{TI_i} \propto (L_{TI0} + L_{TI_i}) \times C_{TI_i} \tag{3.6}$$

Both, positive and negative changes are treated. Consider the theoretically biggest negative change of a TI-match from one to zero. The salience would be zero due to the current TI-match ($L_{TI_i} = 0$), which is not plausible (even the disappearance of a TI should be registered and selected as "relevant" in a certain situation). To compensate this effect, the absolute level

[4]The decay of the salience is considered later in this section.

of a TI-match in equation 3.6 is corrected by a bias value L_{TI0} (equation 3.7) for negative changes.

$$L_{TI0} \quad \leftarrow \quad \begin{cases} 0 & \text{for positive changes} \\ \frac{C_{TI_i}}{2} & \text{for negative changes} \end{cases} \tag{3.7}$$

With this bias value, the component indicating the absolute TI-match is adapted to a value between the current level (L_{TI_i}) and the level of the previous situation ($L_{TI_{i,prev}}$). Similar to the concept described for the feature emotions the salience for a TI-match is further normalized (equation 3.8).

$$S_{TI_i} = 1 - e^{-\frac{(L_{TI0}+L_{TI_i})\times C_{TI_i}}{S_{TI0}}} \tag{3.8}$$

This normalization is done to get a plausible salience value between zero and one ($0 \leq S_{TI_i} \leq 1$). An exponential function has been chosen to compensate the reverse exponential effect of multiplying values between zero and one (L_{TI_i} and C_{TI_i}). Therefore, the parameter of the exponential function S_{TI0} has to be determined empirically, to get absolute salience values that are "compatible" with the values of other features.

Salience of Actions

The current behavior is determined by the composition of simultaneously executed actions. Whether an action is relevant in the current situation and thus part of the focal element of the encoded event is determined by the salience. Following the event based triggering approaches for the actions discussed in section 3.2.1, a situation should be encoded as an event, if there is a relevant change in the current behavior. Imagine when you are driving over a long distance. If you are phoning while driving, the action phoning is more relevant in the particular situation, because the action driving has been executed since a very long time and is thus nothing extraordinary (the EM only registers changes [Tul83]). The event will be recorded into the EM and according to Tulving's terminology the action phoning would belong to the focal element, while the action driving would refer to the setting of the event.

Thus it is proposed, that an action in the EM module is relevant and further salient in a situation, if it was not executed in the previous situation. Furthermore, an action is salient, if it was *accomplished*. An action is accomplished, if its goal was reached successfully—for example the action *search for food* is accomplished, if an energy source is found. The salience value of an individual action i S_{A_i} is thus defined to be one, if a new action is set in a situation or if the goal of an action is accomplished in a situation, otherwise it is set to zero.

The salience of a feature element is determined at the time of triggering encoding. An event is encoded, if the salience of a feature is high enough (section 3.2.1). Thus a feature could be very salient in a particular situation but in the next situation, if the change would be zero, the salience would be zero as well. As the salience represents a kind of activation of a feature in a situation, it would not be plausible that a feature is very salient in a particular situation and absolutely not salient in the infinitely shortly succeeding situation. Thus, it is proposed that the salience of a feature should decay at a specific rate. If a feature element was very salient in a recent situation, it should still be salient—in other words activated—to a specific amount in the current situation too. Thus, a decay rate of the salience S_{F_i} of element i of the feature F is introduced in the model by which the salience of a feature decays by a rate α of the salience of the feature element in the previous situation $S_{Fi,prev}$ (equation 3.9).

$$S_{F_i} = S_{F_{i,prev}} \times \alpha \tag{3.9}$$

If the salience of the feature element in the current situation is higher than this decayed value, it will be replaced by the current salience value. Further, if the decay of the salience falls below a certain threshold S_0, the salience of the feature is cut off to a value of zero. Even if the remaining salience would not affect subsequent encoding based upon this particular feature, but it would influence the overall salience of the event (equation 3.11), if its encoding was triggered based upon another feature. Because the feature is still highlighted as being salient, it will be possible for later spontaneous retrieval to access this feature.

Based upon the definition of the salience for each constituent elements of a feature, the salience S_F for a feature F is now derived as a function f of the salience values S_{F_i} of the feature elements i (equation 3.10). For example, the salience for the feature emotions is determined by a combination of the individual emotions anger, fear and lust.

$$S_F = f(S_{F_i}) \tag{3.10}$$

One possibility for implementing, the function f in equation 3.10 is the maximum function. The salience for the feature F is thus determined by the particular feature element i having the maximum salience level.

The salience of an event S_{Ev} is further determined by the salience values of the features (emotions, TI-matches, actions and drives). Each feature contributes a share to the salience of the event (equation 3.11).

$$S_{Ev} = \alpha_E S_E + \alpha_{TI} S_{TI} + \alpha_A S_A + \alpha_D S_D \tag{3.11}$$

with

$$\alpha_E + \alpha_{TI} + \alpha_A + \alpha_D = 1$$

The individual coefficients $\alpha_E, \alpha_{TI}, \alpha_A$, and α_D have to be determined empirically according to their importance. The salience of an event can be interpreted as the "unusualness" of an event. The more unusual an event is, the better it will be remembered [Bad97, p. 219]. To summarize, the composition of an event of its features and the formation of the salience, originating from the individual salience values of each feature element is illustrated in figure 3.4.

3.2.3 Recoding

Referring to Tulving's General Abstract Processing System (GAPS) (see section 2.1.3), recoding are called the processes which describe all kinds of changes of an encoded engram. It happens when an event is repeated. A repeated event in terms of GAPS is an interpolated event—an event that is very similar to the original event. The question that arises for the adaption of recoding processes to the EM module is the same as Tulving discusses for human memory. Should the original engram be modified by the interpolated event or should the interpolated event be stored separately? Tulving suggests that the encoding of an event may include reference to similar or related events as part of the memory trace (section

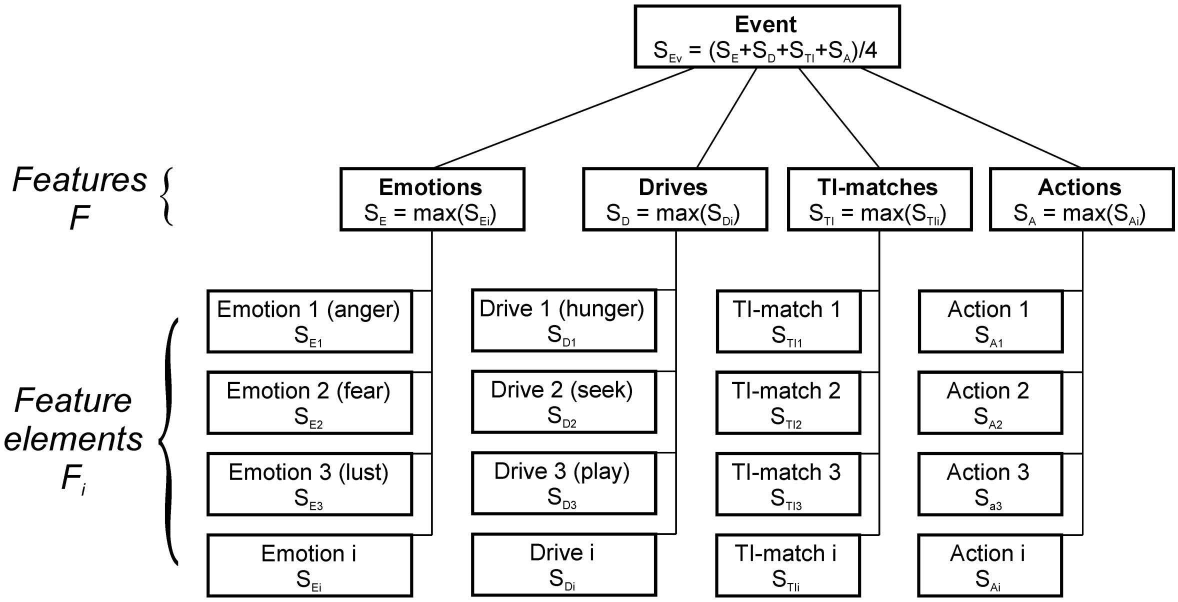

Figure 3.4: Event structure and salience determination

2.1.2). In the EM module, modifications of originally encoded engrams due to similar events are proposed to be omitted due to simplicity. It is not the objective to rebuild human EM, but to make a model inspired by neuro-psychoanalytical theories. As there cannot be found any suitable approach how to implement recoding effects, all engrams once encoded remain unchanged in the memory. But what is realized in the model is that relations between events and episodes are made. Events in the EM module of the emotional arbitration model are defined as synonyms for Tulving's simple events whereas episodes as sequences of events are defined as synonyms for Tulving's complex events (section 3.1.2). As an episode is defined as a series of events, the relation between an episode to events is implicitly encoded in the episode structure. The other way round, the relation between an event and the episodes it belongs to is further encoded as part of an event. An event contains a structure with relations to all episodes it is part of. Relations between episodes are realized in the storage of episodes (see section 3.3). Episodes are mapped together referring to the corresponding scenario template they are realizations of. Relations between single events in order of their contents are not made. Simple events are only related temporally.

3.3 Storage

The task of the storage processes is to define a memory structure and to handle the dynamics of the stored engrams. The issues that are concerned within this memory functionality are:

- How are the encoded engrams maintained in memory?
- How do they change over time?

The first task deals with the determination of a memory structure and the maintenance of the stored events. Episodic memories are temporally ordered, and have many relations to each other. Maintenance is made without control of the entity, it is not aware of alterations [Tul83]. The main requirement to the memory structure in a technical sense is that it should be as efficient as possible. This topic is considered in the first part of this section.

The dynamics of the memory entries, how they change over time, deal with the aspects that are called “forgetting”. Even if computers are able to store everything without any limits of time and amount, there has to be implemented a mechanism that highlights more important memories over less important to make the memory system effective. These aspects are depicted in the second part of this section.

3.3.1 Storage structure

Anatomically, episodic memories are mainly encoded in the hippocampus, tightly coupled with the limbic system (section 2.1.2). These profound theories in neurology does not give a lot advise how to store memories in a computer. The structure of the storage medium is optimized for the most usual case in terms of speed of access. The most usual case of accessing the memory is spontaneous retrieval (section 3.4.1). For each retrieval, all memory entries have to be traversed to get the memory that best matches a given cue. Given a huge amount of existing memories, the burden due to the sequential search gets very high. Thus, a fast access data structure has to be chosen.

Events have to be stored considering their temporal order. One event precedes and follows another chronologically ordered. Thus, events in the EM module are stored in a *memory container* chronologically indexed (figure 3.5). Episodes are sequences of events and repre-

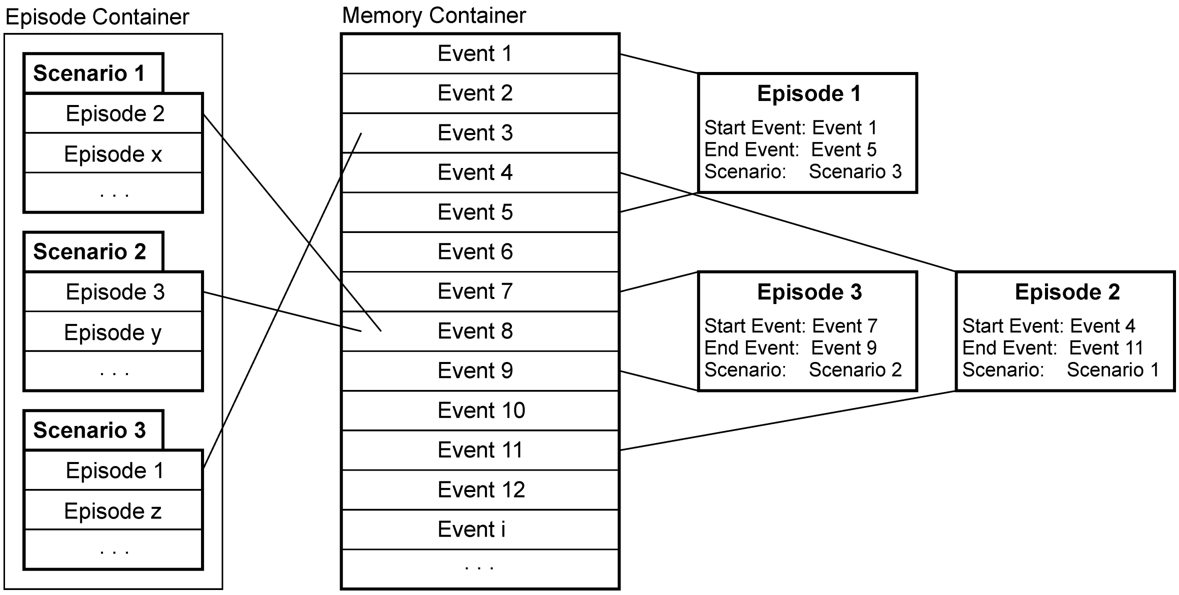

Figure 3.5: Memory structure

sent concrete realizations of predefined scenario templates. As considered in section 2.2.3, a scenario template consists of states and transitions (figure 2.7). A transition is triggered and the scenario gets into the next state, when a specific condition is fulfilled. Such a condition might be that a particular TI-match has to be above a certain threshold. The series of events within which a scenario template is completed, beginning from the event when the first transition triggered until the event in which the triggering condition of the last transition was fulfilled, is then stored as an episode. As illustrated in figure 3.5, *scenario 3* is accomplished within *event 1* and *event 5*. The series of events from event 1 to event 5 is related to *episode 1* (the concrete realization of *scenario 3*), which is further stored into the *episode container*. Episodes stored in the episode container are organized into sub-container. Each sub-container maps together all episodes that are realizations of one specific scenario template. Thus, episodes are grouped by their context—the scenario template. Scenario templates are the part of semantic memory that can be denoted as scripts or schemes (section 2.1.1). Having such a basis of scripts it makes it easier to remember specific experiences (consider the *going to restaurant* example in section 2.1.1).

Episodes may overlap or be nested in time. As illustrated in figure 2.7, episode 3 is nested into episode 2. Consider scenario 1 represents *going-for-dinner*, and scenario 2 *ordering-meal*. Then the episode realizing ordering-meal (episode 3) is part of the overall episode realizing going-for-dinner, it is nested into it. Episode 1 and episode 2 in contrast are overlapped in time. Imagine, scenario 3 would be *business-talk*. Episode 1 (realization of scenario 3), then might begin before going-for-dinner (episode 2), lasts into this scenario realization, but ends before ordering-meal (episode 3). Single events further are related to all episodes they belong to. For instance, event 8 is part of episode 2 and episode 3 whereas event 3 only belongs to episode 1. Thus, if a particular event is later retrieved by a specific cue, it would be possible to directly recall an episode it belongs to.

Events do not have semantic contents. If there were relations between template images, this would have been the semantic information implicitly encoded in events. As in the current implementation of the emotional arbitration model template images are stored into the image memory without any relations among each other, single events do not have semantic contents. Indirectly there might be derived semantic knowledge by a possible relationship to an episode. Consider the particular event 7 (figure 3.5) which might represent *waiter brings menu*. If this event was later retrieved, due to its membership to episode 3 (scenario ordering-meal), the corresponding scenario template would be further retrieved and it would be possible to derive that the succeeding event will be for instance *select meal from menu*. Thus, future situations can be foreseen. With this correlation between episodic information (the concrete episode) and semantic information (the scenario template), the inferential capability (section 2.1.1) of SM is used to extract additional information from an event. In contrast, no semantic knowledge from a single event without any relation to an episode (e.g. event 12) can be derived. If a single event is retrieved, further episodic information can only be obtained by retrieving its preceding or succeeding events.

3.3.2 Forgetting

Forgetting is a central concept in the theory of memory. When talking of forgetting, it is not meant that some memories are lost and discarded from memory. Memories once encoded into long-term memory, remain there very persistent. What is indicated with the term “forgetting” is the inability for successfully recalling a memory given a certain cue [Tul83, p. 201]. Tulving distinguishes between availability and accessibility of memories (section 2.1.2). Whereas memories still exist in memory (they are still available), they can no more be retrieved by an appropriate cue (they are not accessible). Solms also emphasizes on the persistence of memories, and defines in this context the consolidation mechanism as a process of wiring (section 2.1.1). Solms states, that it is a process that is activity-dependent—if some memory circuits are no more used, they die off ("use it or loose it"). Furthermore, the activation of memory traces in long-term memory only comes along with the bringing back to awareness a specific memory—i.e. when remembering an event [ST02, p. 146].

The aspects of human forgetting are considered in detail in section 2.1.2. To summarize, the main determinants for the ability for successfully recalling an event, or reversely, for the rate of forgetting are:

- Time delay since the encoding of the event
- Emotional involvement of the agent to the event
- Frequency of subsequent recollection
- Salience of an event
- Richness of the cue provided

The more salient, the higher its emotional involvement, the more recently it occurred, the more frequent it was subsequently recalled and the richer the cue provided for retrieval, the greater the chance for recalling that particular event. Thus, a forgetting mechanism is derived on the basis of these facts.

For each event the salience is determined (section 3.2.2). The higher the salience of an event, the higher the probability[5] for subsequent retrieval of this event. Each event is further assigned with an emotional tone that represents the emotional involvement of the agent to this event. The higher its emotional arousal, the higher the emotional tone. The emotional tone ET_e of an event e is thus a function f dependent on the involvement of the individual emotions E_i to an event (equation 3.12).

$$ET_e = f(E_i) \tag{3.12}$$

One possibility for implementing the function f is the maximum function. The emotional tone is then determined by the individual emotion having the maximum emotional level.

There has been observed a tendency, that older memories become less accessible than newer ones and that this dependency on time is exponential [Bad97]. Lengthening the time delay between the encoding and retrieval dramatically increases the rate of forgetting, which further slows down with the passage of time [Sch96, p. 73]. To implement this aspect of recency of the occurrence of an event, an activation level is attached to each stored event. Thus, the activation level $A_e(t)$ (equation 3.13) of each stored event e decays with an exponential rate (figure 3.6). The parameter t of the activation function $A_e(t)$ is not meant to be a variable representing the absolute time measured in seconds. This dependency should just indicate a temporal correlation. The decay rate is determined by the parameter T_0.

$$A_e(t) = e^{-t/T_0} \tag{3.13}$$

[5]There are no discrete probability distribution functions implemented in the EM module. When using the term probability for retrieval, it is meant that some events may be more easily retrieved due to a higher activation than others.

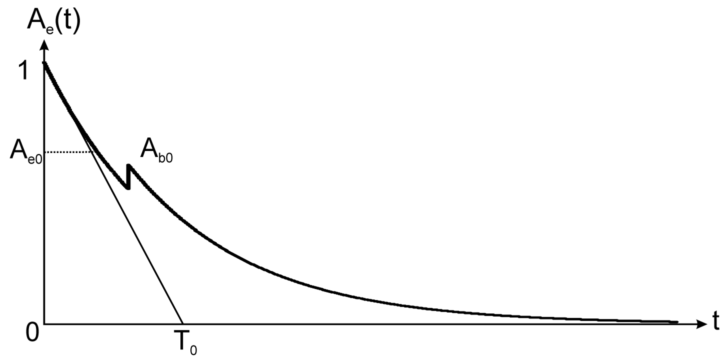

Figure 3.6: Decay of the activation level $A_e(t)$

At the moment of encoding, the activation is initialized by a specific value A_{e0}. The longer an event is *active*[6], the higher its activation compared to the other stored events, which decay while the event is active. If an event is recalled, its activation receives a boost A_{b0} (figure 3.6). The more frequently an event is recalled, the higher becomes its activation and the higher its chance for subsequent recollections (retrieval practice effect (section 2.1.2)). Furthermore, each episode to which an event is related to—for instance episode 1 for event 3 in figure 3.5—is strengthened as well. Strengthening an episode means that all events the episode consists of are reinforced by an activation boost A_{b1} (figure 3.7). Thus, the events 1 to 5 in figure 3.5 are reinforced. The next time, another event (e.g. event 2) of the same

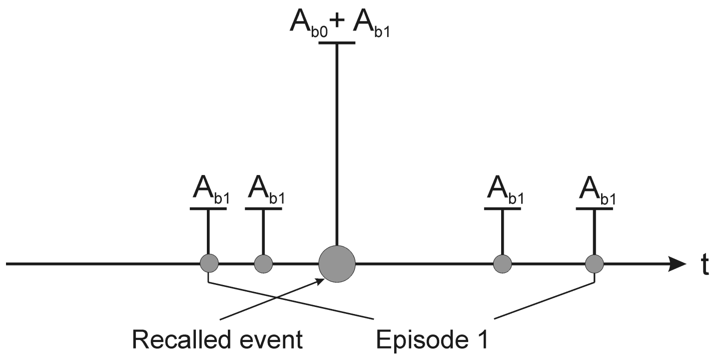

Figure 3.7: Strengthening of an episode

episode is retrieved, all events that belong to the particular episode again receive a boost. Events that belong to the same memory trace (to the same episode), boost their activation levels together. In other words, they *fire* together and further *wire* together. Similar to the consolidation mechanism, it can be said, that the process of jointly strengthening events that belong to the same episode is a process of wiring. Furthermore, the concept of strengthening the whole episode includes the process of laying down the episode deeper and deeper into memory. The more activated the memory trace of the episode, the higher its chance for subsequent retrieval and the less likely it will become inaccessible. Reversely, if memories were not activated by subsequent retrieval, their activation level gets lower and lower, and they become inaccessible ("use it or loose it").

[6]An event is *active* as long as no encoding of another event is triggered. In other words, the active event is the most recently encoded event since which the situation remains unchanged.

Events that belong to an episode decay slower than single events without any relationship to an episode. This feature comes along with Tulving's statement, that episodic memories are organized relatively loose. Initially encoded detailed information can be easily lost. This effect is stronger for simple events than for complex events (section 2.1.1). Events are further better remembered if they occurred in a specific context. The context in which an episode occurs is the scenario template to which it is associated. As events in the EM module are used as synonyms for Tulving's simple events and episodes for complex events that occurred in a specific context, it is proposed that events related to episodes decay slower than single events. This can also be seen as a process of sifting out single events without relations to episodes. Schacter further notes that memories are established durable if the incoming information is associated meaningfully with knowledge that already exists [Sch96, p. 45]. The existing knowledge in the EM module represent the scenario templates. As episodes are associated with such existing knowledge, they remain more durable in the memory than single events.

The decay of the activation level of a stored event further depends on the emotional involvement of the agent at the time of recording. As depicted in section 2.1.2, high emotional arousal to events lead to enhanced later recall. Emotional activity has a positive effect on memory trace consolidation [Bad97, p. 276]. The higher the emotional arousal to the encoded event, the higher its chance for later recall. The effect is implemented in the activation function $A_e(t)$. The higher the emotional arousal of the agent to an event, the slower decays the $A_e(t)$ and the better the event will be remembered. Furthermore, the salience of an event has an impact on the ability for later recalling this event. The positive dependency is further implemented in $A_e(t)$. The higher the salience of an event, the less its chance to be forgotten and the slower decays its activation level. Thus, the parameter T_0 of the activation function $A_e(t)$ in equation 3.13 depends on the emotional tone ET_e, the salience S_e and the relatedness rel of an event to one or more episodes. This dependency is expressed by a function f (equation 3.14).

$$T_0 = f(ET_e, S_e, rel) \tag{3.14}$$

The influence of each component is expressed by the parameter T_{0ET}, T_{0S} and T_{0rel}. T_{0ET} represents the influence of the emotional tone (equation 3.15), T_{0S} the influence of the salience of the event (equation 3.16), and T_{0rel} represents the influence of the relatedness of the event to other episodes (equation 3.17) on the parameter T_0 of the activation function $A_e(t)$.

$$\frac{\partial T_0}{\partial ET_e} = T_{0ET} > 0 \tag{3.15}$$

$$\frac{\partial T_0}{\partial S_e} = T_{0S} > 0 \tag{3.16}$$

$$\frac{\partial T_0}{\partial rel} = T_{0rel} > 0 \tag{3.17}$$

The influence of each component is positive. The higher each component, the higher the parameter T_0 and thus the less decays the activation level. The values of the individual parameter T_{0ET}, T_{0S} and T_{0rel} have to be determined empirically.

Another aspect that is deduced from this concept is an effect of forgetting that arises due to competition between events (interference theory, see section 2.1.2). Two events are in competition in retrieval, when they are very similar to each other and therefore both may be evoked due to a specific retrieval cue. So the event with the higher activation will be recalled, which further implies that it will be boosted and the chance for later recall based upon the same cue is higher. Thus, the other similar event will be "forgotten" because it is in competition with the more activated event referring to a specific retrieval cue.

Imagine, two events are in competition and their activation decays at different rates (figure 3.8). Event 1 ($e1$) occurs at t_1 and event 2 ($e2$) at t_2. Both are similar to a particular retrieval cue, thus they are in competition when stimulating with this cue. The events decay at different rates, characterizes by their different emotional tone, salience level and a possible relatedness to an episode. These different rates are defined by the parameter T_0 (equation 3.14) of their activation function. A_{e1} represents the activation of event 1 and A_{e2} the activation of event 2. If there is no retrieval with the specific cue initiated between t_2 and t_3, event 1 will have advantage over event 2 from t_3 on for further recall. The chance for successfully recalling event 2 after t_3 will be less and less. If a retrieval is initiated between t_2 and t_3, event 2 has advantage over event 1 due to its recency of occurrence and receives a boost if it is recalled. Anyway, this boost would only shift t_3 to the right if event 2 was not retrieved another time shortly after this recall. In the long run, event 1 would have advantage over event 2 if the latter was not frequently recalled shortly after its occurrence. The combination of the three

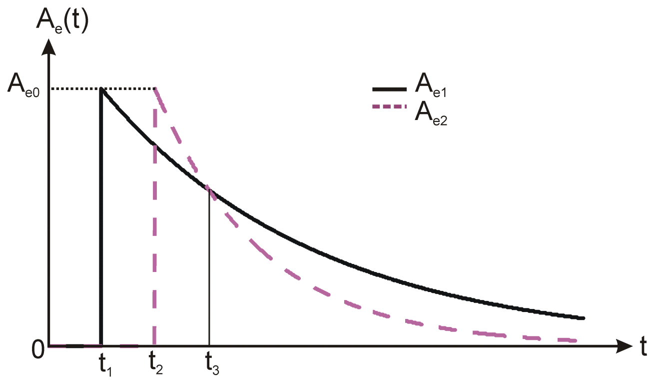

Figure 3.8: Events with different activation decay rates (A_{e1}: activation of event 1, A_{e2}: activation of event 2)

attributes of an stored event in the EM module—activation level, salience and emotional tone—determine the ability for subsequent retrieval of this event by a specific retrieval cue. How retrieval is performed, is considered in the next section.

3.4 Retrieval

The task of retrieval is to recall an appropriate encoded engram given a certain stimulus. This memory functionality determines:

- When is retrieval initiated?
- How is the retrieval cue constructed?
- How is the similarity of a stored event to the indicated retrieval cue measured?

- Which event is recalled and what are the consequences?

The first issue deals with the triggering of a retrieval, to bring the EM system into retrieval mode. Episodic memories may come unbidden into mind. They may pop up spontaneous without the explicit effort of the agent. But memories can also be evoked deliberately (section 2.1.2). The issue, when retrieval has to be triggered, is concerned in the first part of this section. Once the EM system has been brought into retrieval mode, a retrieval is initiated by constructing a retrieval cue. The composition of the data used to cue the retrieval of an stored event is discussed in the second part of this section. Matching concerns the issue of defining the similarity of the individual stored events to the indicated features of the retrieval cue. The compatibility relation between the stored event and the retrieval cue has been chosen to be matching of features. How matching of an event to a cue is performed is detailed in the third part of this section. Which event is recalled when stimulating with a specific cue not only depends on the similarity between a stored event and the features of the retrieval cue. Additional factors like the activation level, which further depends on the salience and the emotional tone of the event at the time of recording also play a crucial role. The determination which event will be recalled including its related episodes based on these factors and what are the consequences of the recollection is dealt in the final part of this section.

3.4.1 Triggering Retrieval

Triggering the retrieval of a stored event defines the particular moment, when it is initiated. Retrieval might occur spontaneous or deliberate (section 2.1.2). Memories may pop up in our mind as a kind of involuntary process, or they may be recalled as the result of an active search process. Imagine, if you come across a particular person you have not seen for a very long time (e.g. a former classmate), memories about experiences with the person might pop up spontaneous at the moment when you see the person. Alternatively, you might explicitly enforce the searching for a memory about a particular happening in the past (e.g. what you did last Sunday). Thus the access to episodic memories depends a lot on the individual's mental state. While semantic memory interprets perceptual changes in the environment immediately (section 2.1.1), EM must be in *retrieval mode* that a stimulus may serve as an effective retrieval cue. To be in retrieval mode means that the subject must be aware for retrieval. So, the challenge for triggering retrieval is to define, when the EM system gets into retrieval mode that causes the retrieval process to be initiated.

As the agent explicitly enforces the recollection of a previous experience for deliberate retrieval, the EM system is brought into retrieval mode by forming a retrieval cue. Deliberate retrieval is initiated explicitly by the agent and the EM module just has to provide an interface to support this. To define when the EM system gets into retrieval mode without an explicit stimulus of the agent and consequently spontaneous retrieval is initiated represents a great challenge to the EM module. There does not exist a profound theory in psychoanalysis that describes exactly when spontaneous retrieval is evoked. Possible approaches to implement spontaneous retrieval are summarized in table 3.3. The first approach, triggering at a fixed frequency brings the system into retrieval mode at a fixed rate. The retrieval cue would be formated by the currently perceived situation and it is searched for a similar situation in the memory. It fulfills the demand that spontaneous retrieval occurs frequently, but it has

Approach		**Pros & Cons**
Trigger at a fixed frequency	Pro:	Easy to implement
	Con:	Occurs at deterministic moments
	Con:	No cause that forces it
Trigger continuously	Pro:	Memories may be evoked at any time
	Con:	EM is always in retrieval mode
	Con:	Very time intensive
	Con:	May continuously evoke the same memory
Trigger based upon the emotional state	Pro:	Emotional state represent mental activity
	Con:	Not clear, which emotional state brings EM into retrieval mode
Trigger when encoding a new event	Pro:	Reminding events with a similar happening
	Pro:	Plausible cue formation
	Con:	Only initiated when something happens to the agent

Table 3.3: Triggering spontaneous retrieval

no plausible basis for its occurrence. It occurs at deterministic moments in time and there is no cause that forces it.

Triggering retrieval continuously implies, that the EM system is in retrieval mode all the time. As there does not exist a profound theory when the system gets into retrieval mode, this may be plausible because the recollection of a memory can be evoked at any time. The retrieval cue again would be the entire situation. Every currently perceived situation is checked with all previously gained experiences and if the relationship between the cue and the stored event is high enough, the agent remembers the event with the best retrieval result. A disadvantage of this approach is the extreme time intensive processing. Another drawback is that if a situation does not change and remains constant for a long time, the agent constantly recalls the same experience until something in the situation changes. Thus the cue formation is not plausible.

Triggering spontaneous retrieval based upon the current emotional state of the agent brings the EM system into retrieval mode, when the agent is in a particular emotional state. This approach seems to be very plausible. As retrieving past experiences from EM depends a lot on the mental state, and the emotions represent more the less the mental state of the agent, it seems to be a good approach to bring the agent into retrieval mode based upon the emotions. The problem is to define, in which emotional state the agent has to be to get into retrieval mode. High emotional arousal suppresses recall immediately whereas events recorded with low emotional arousal are badly remembered after a short while [Bad97, p. 276].

This has been observed in the context of the influence of the emotional arousal when recalling a specific event, but it can be deduced that the emotional levels have to be in a "balanced" range to get into retrieval mode. Both, very high and very low emotional arousal do not seem to be the best mental environments to remember past experiences. Thus, if implementing spontaneous retrieval based upon the emotional state in the EM module, it is proposed that the emotional arousal is in a balanced range.

The fourth approach proposes, that spontaneous retrieval is triggered at the moment, when a new event is encoded. An event occurs when something in a situation changes significantly. Something relevant happens to the agent, both internally and externally. Thus, it seems to be plausible that this happening forces the agent to recollect a similar previous experience. With this approach, the EM system is brought into retrieval mode each time the encoding of an event is triggered. The encoding of an event is mainly triggered based upon the determination of the features of the event that are relevant (salient) to the current situation (section 3.2.2). Thus it is proposed, that the salient features of the encoded event are used as retrieval cue. By using the salient features of an event as cue implies that it is cued for a past experience in which the salient happening was similar. A disadvantage of this approach is that it is only cued for a event when something happens to the agent and not at any arbitrary moment in time.

This approach can be referred to the *madeleine memory* detailed in [Sch96, p. 27]: A person drinks a tea with a madeleine[7] and experiences a mysterious feeling of well-being after dipping the madeleine into the tea. After tasting the tea again and again each experience gets weaker and weaker to the previous one. Thus it can be concluded, that the effect does not lie in the cup but in the person himself. The tea and the cake have activated a past experience. The features of the experiencing event (the combination of the objects, the smells and the tastes) makes them the uniquely effective cue to activate this seemingly lost experience. The madeleine episode highlights that the re-experiencing of one's personal past sometimes depends on the chance of encountering objects that contain the keys to evoke memories that would otherwise be hidden forever.

Even if it is proposed that triggering spontaneous retrieval—bringing the EM system into retrieval mode—is a combination of the two latter concepts, in the current implementation only the fourth approach is used. This is due to the fact that it is not quite clear, how the emotional state has to be composed of. This spontaneous retrieval mechanism, similar to which Tecuci has proposed in his work [Tec05], is called *reminding*.

3.4.2 Retrieval Cue Construction

Once the EM system is brought into retrieval mode, a retrieval is initiated by the formation of a retrieval cue. The retrieval cue is a stimulus, hint, or clue that is used to evoke a previously experienced event. The number and richness of cues provided is essential for the ability to remember autobiographical events. Coming back to Tulving's distinction between availability and accessibility of a memory, a still available memory only becomes accessible, if the cue provided is rich[8] enough (section 2.1.2). Referring to the diary experiment depicted in section 2.1.2, nearly every occasion can be remembered if a cue that is rich enough is given. So the formation of the retrieval cue is very crucial for memory performance.

The general form of an retrieval query in EM usually consists of temporal, spacial, and behavioral information (section 2.1.1). Behavioral information in the EM module is represented by the features actions, spacial information can be interpreted by the feature template

[7]A madeleine is a french cookie.

[8]The richness of a cue in this context refers to the quantity and quality of provided features.

image-matches. Temporal information is only represented by a chronological ordering of the events. It cannot be retrieved past events using a specific moment in time as cue. Temporal queries that can be made in the EM module refer to an immediately retrieved event. On the basis of the retrieved event, the corresponding previous or succeeding event in memory can be queried. Additional to the temporal, spacial, and behavioral queries it can be cued for specific emotional states of previous experiences. Our own states of mind can also serve as valuable cues to remember events [Sch96, p. 62]. A retrieval in the EM module can be cued in various forms:

- Use an event as cue
- Use a subsets of features as cue
- Use a scenario template as cue
- Cue for previous or succeeding events

When an event is used as retrieval query, the salient features of the event are used as retrieval cue. To remember, the salient features of an event are the currently (and recently due to its decay) relevant, highly activated features to a current situation, which represent the focal element of the event (section 3.2.2). By using the salient features as retrieval cue implies that it is queried for a stored event, in which the salient happening was the same as in the current situation. This cue construction mechanism is used for spontaneous retrieval (section 3.4.1) and refers to what Tulving states in its *encoding specificity principle*: When a person encodes something, "he encodes it in terms of its meaning or other salient properties, with the result that corresponding cues again are effective reminders when presented at the time of retrieval" [Tul83, p. 221]. Tulving emphasizes with this theory, that a cue for retrieval is only effective, if its properties are similar to the properties of the encoded event. If a stimulus leads to the retrieval of an event, it is supposed that it has been encoded whereas if not, it is assumed not to have been encoded [Tul83, p. 264]. This interaction between encoding and retrieval phenomena is implemented in the EM module with the concept of selecting the salient features of an encoded event. The salient properties of an event are highlighted in the stored engram to serve as effective reminders when they are again presented for retrieval. These salient features establish the *compatibility relation* (section 2.1.2) between the retrieval cue and the engram to evoke it for recollection. The salient properties initially encoded at the occurrence of the event represent the properties that may serve as effective cues when presented again to recall that particular event.

Using a subset of features as retrieval cue implies that some features of a situation are used as cue, independent whether the features are salient or not. An arbitrary number of features containing an arbitrary composition of feature elements might be used to cue a memory. If all features with all feature elements are used, it is cued for an entire situation. For example, a cue might be formed by a single TI-match, a specific action and the emotional state. Thus it is queried for a stored event that matches best the formed subset of features of a situation. This cue construction mechanism is usually used for deliberate retrieval in the EM module. The agent deliberately selects some features which are used to cue for a stored event.

The presented possibilities of constructing a retrieval cue optionally can be complemented by a scenario template. When adding a scenario template to a retrieval cue, only the episodes, which are realizations of the particular scenario template, are traversed to retrieve an appropriate event stimulated by the cue. Thus it is cued for an event that happened in a specific context. Only events that happened in the indicated context are traversed to get a match. If the concrete scenario template *make-coffee* is attached to the retrieval cue which queries for an event where the milk fell down to the floor, only the events belonging to the episodes that are realizations of the scenario *make-coffee* are traversed. This is a retrieval cue referring to: "remember when the milk fell on the floor, when you made coffee". Without the additional scenario information, an event could be retrieved when you were not making coffee but for example when you were shopping and the milk fell down the cooling shelf.

As mentioned above, when retrieving an event it can be further cued for the corresponding previous or succeeding event in memory. Thus, a temporally ordered sequence of events can be reconstructed. While for spontaneous retrieval ("reminding") the retrieval cue is formed by an event, for deliberate retrieval the cue can be constructed by any of the proposed concepts.

3.4.3 Matching

Once the retrieval is initiated by bringing the EM system into retrieval mode and forming a retrieval cue, the task of matching is to find a similarity relation between the retrieval cue and the stored engram. A cue is effective, if it activates an appropriate memory trace. Tulving describes the effectiveness with a theory which he terms the *encoding specificity* (section 2.1.2). The chance that a stored event will be retrieved depends directly on the similarity between the properties of the memory trace and the properties of the retrieval cue. There must exist a compatibility relation, which can be matching of features, associative relatedness, informational overlap and the like [Tul83, p. 224].

In the EM module the compatibility relation between the cue and the engram is defined to be matching of features. Matching in a narrower sense is the process of measuring the identicalness of the features of the cue to the features of an event. When using a scenario template as part of the retrieval cue, the compatibility relation can also be considered as associative relatedness. Scenarios and episodes are associatively related. Episodes are mapped in an episode container (figure 3.5) to their associated scenario template (its context). Thus, when cuing with a scenario template, only the events belonging to the episodes associated to the scenario template indicated in the cue are measured for their similarity to the cue based upon matching of features. This retrieval mechanism is applied when it is desired to search for a particular event happening in a certain context.

The EM system has to search for the stored event that best matches to the given retrieval cue. To accomplish this, all available memories are traversed sequentially and for each stored event a match to the cue is determined (when cuing additionally with a scenario template, only the events belonging to the episodes are traversed). In the most frequent case—for spontaneous retrieval, when the retrieval cue is constructed by the salient features of the currently encoded event—only the features of the engram that were salient at the moment of encoding are accessible. This comes along with Tulving's statement, that the salient properties initially encoded serve as effective reminders [Tul83, p. 221] for this event by establishing the compatibility relation between the cue and the engram. Thus, if a particular feature $F_{i,c}$ of the cue is salient and the same feature $F_{i,m}$ of the event in memory is not salient, then the match for this specific feature will be zero. Excepted from this access restriction is the feature emotions. The match for the feature emotions is always determined.

The affective component of episodic memories play a crucial role especially when retrieving. Anything experienced in a specific mood tend to be recalled more easily when the specific mood is reinstated (mood state-dependency, see section 2.1.2). Thus, sad events are better recalled when feeling sad and happy events when feeling happy. To highlight this important dependency of emotions on EM, it is further proposed that the matching of the feature emotions is weighted higher than the matching of the other features.

For deliberate retrieval, the cue can either be formed by an event (then only the salient features of the engrams are traversed), or the cue can be arbitrary constructed, and all features of the stored events are traversed. The fact that all detailed information are accessible for retrieval does not come along with Tulving's theory (not the entire, detailed information about the event is accessible). Anyway, the intent of the work is not to rebuild human episodic memory in a computer, but to implement a model that is inspired by psychoanalytical theories about human memory. As computers are able to store huge amounts of data without restrictions in detail, the accessibility of all features of a stored event is additionally supported in the EM module. Thus, for deliberate retrieval, when cuing with an arbitrary number of features, the EM module provides access to all features of a stored event to get a match.

Matching of Emotions, Drives and TI-matches

The matching of the features emotions, drives and TI-matches is accomplished in the same manner. For the $i-th$ feature element[9] $F_{i,c}$ in the cue, the match to the same feature element $F_{i,m}$ of each event in the memory is determined. Consider the feature emotions. The match of each individual emotion is determined by measuring whether the levels of an emotion from the cue and the stored emotion are within a certain range. Thus, the match for the $i-th$ emotion $E_{i,m}$ from an event in memory is determined by a function that depends on the difference Δ_{E_i} of the levels of the cue emotion $L_{E_{i,c}}$ and the level of the emotion from the event in memory $L_{E_{i,m}}$ (equation 3.18). The match $M_{E_{i,m}}$ for this stored event is the higher, the smaller the difference Δ_{E_i} between the emotional levels. This dependency is expressed by function f (equation 3.19).

$$\Delta_{E_i} = \left| L_{E_{i,c}} - L_{E_{i,m}} \right| \tag{3.18}$$

$$M_{E_{i,m}} = f(\Delta_{E_i}) \tag{3.19}$$

One possibility for implementing the matching function f is the Gaussian bell curve (equation 3.20). The advantage of the Gaussian curve is its smooth course. Differences within a certain range (Δ_0) get a very high match, whereas the greater becomes Δ_{E_i}, the smaller becomes the match.

$$M_{E_{i,m}} = e^{-\frac{1}{2}(\frac{\Delta_{E_i}}{\Delta_0})^2} \tag{3.20}$$

The parameter Δ_0 defines the shape of the Gaussian matching function (figure 3.9). If $\Delta_{E_i} = 0$ (the emotional levels of the cue event $L_{E_{i,c}}$ and the stored event $L_{E_{i,m}}$ are the same), the match for this emotion of the stored event is one ($M_{E_{i,m}} = 1$). The greater Δ_{E_i}, the less becomes the match. In figure 3.9, the emotional level of the stored event $L_{E_{i,m}}$ is 0.6, the level of the cue event $L_{E_{i,c}}$ is 0.4, thus the difference Δ_{E_i} is 0.2 and the match $M_{E_{i,m}}$ becomes 0.3.

[9] An element of a feature is one individual item the feature consists of—e.g. the feature emotions consists of the individual emotions anger, fear and lust (see section 3.2.2).

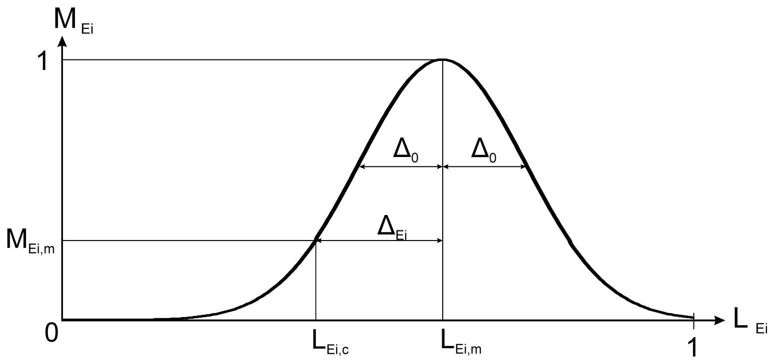

Figure 3.9: Gaussian matching function

The same concept as depicted for the feature element emotion is used for the features drives and TI-matches as well. The match for each feature element (the individual drives D_i and the individual TI-matches TI_i) is determined in the same manner like for the feature emotions. The sign E (indicating the feature emotions) of the equations 3.18), 3.19 and 3.20 has to be replaced by the sign D for the feature drives and TI for the feature TI-matches.

The overall match M_F for a certain feature F of a stored event to a feature F indicated in the retrieval cue is then derived from the individual matches M_{F_i} of its constituent feature elements F_i. Each feature element contributes the same share. If a feature of the cue consists of N feature elements, the match M_F results in the sum of the $N-th$ share of each individual feature element match M_{F_i} (equation 3.21).

$$M_F = \frac{1}{N} \sum_{i}^{N} M_{F_i} \tag{3.21}$$

Thus, if the number of elements for a particular feature indicated in the cue is $N = 5$, each individual match M_{F_i} contributes a fifth to the match of a feature of a stored event to the corresponding feature in the retrieval cue.

Matching of actions

The behavior of the agent in a particular moment is determined by the composition of the executed actions. Matching of the feature actions means therefore, that the behavior indicated in the retrieval cue corresponds to the behavior of an event executed at the time of recording to EM. Thus an action indicated in the retrieval cue matches, if the same action occurs in the behavior of the stored event. The match for this individual action M_{A_i} is therefore one if it occurs in the behavior of the stored event and zero if it does not. The overall match for the feature actions M_A is further determined by the averaged sum of the matches of the individual actions from the cue (equation 3.22).

$$M_A = \frac{1}{N} \sum_{i}^{N} M_{A_i} \tag{3.22}$$

Thus, if the behavior indicated in the cue is composited by $N = 4$ actions, the match of each single action M_{A_i} contributes one fourth to the overall match of the feature actions M_A.

The match of a stored event e to a certain retrieval cue is now derived from the combination of the matches of its constituent features (equation 3.23). Each feature contributes the same share. Thus, if there are $N = 3$ features indicated in the retrieval cue (e.g. emotions, TI-matches and actions), each match of a feature M_{F_i} contributes a third to the match of a stored event M_e.

$$M_e = \frac{1}{N} \sum_{i}^{N} M_{F_i} \tag{3.23}$$

The determination of the match of an event to a certain retrieval cue based upon its constituent features is similar to the determination of the salience of an event (section 3.2.2) and summarized in figure 3.10.

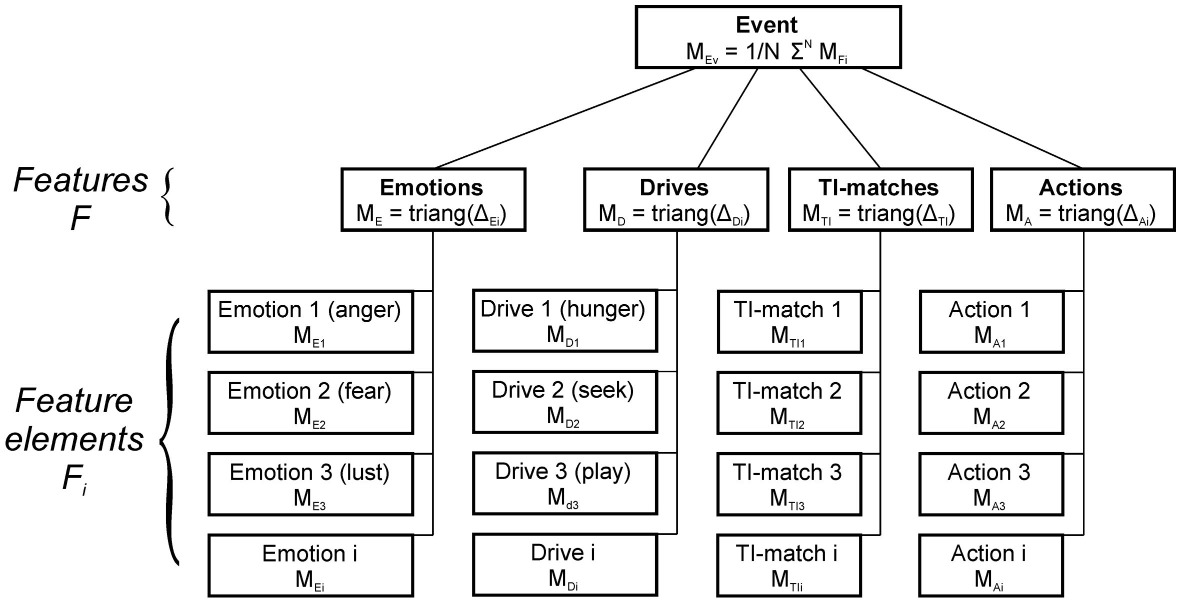

Figure 3.10: Matching structure

3.4.4 Recall

Which stored event is recalled not only depends on the matching of the individual features to a specific retrieval cue. As depicted in section 2.1.2, the ability of successfully recalling a stored event also depends on other factors like the activation level, the salience, or the emotional tone of the event. Therefore a retrieval result R_e (equation 3.24) is derived for each stored event e to a particular retrieval cue on the basis of the matching of features and the activation level which further depends on the salience and the emotional tone of the event.

$$R_e = \alpha \times M_e + \beta \times A_e(t) \tag{3.24}$$

with

$$\alpha + \beta = 1$$

The retrieval result finally determines whether the event is recalled or not. The higher each component (match M_e and activation $A_e(t)$) of a stored event, the higher its chance to be recalled. The event with the highest retrieval result will be recalled, if it exceeds a certain threshold. A mandatory condition for recalling a particular event e is that its retrieval result R_e is above the recall threshold R_{th} (equation 3.25).

$$R_e \geq R_{th} \tag{3.25}$$

It is proposed that the coefficient α in equation 3.24 is set higher than β, for example $\alpha = 0.7$ and $\beta = 0.3$. The influence of matching of features on the chance that an event will be recalled is higher than the activation level. This choice of the coefficients is made because the main part for the retrieval of an event is determined by the similarity relation between the stored event and the retrieval cue (represented by the matching of features). If the value of the coefficient α would not have been chosen that high, events that absolutely do not coincide with the features of the retrieval cue could be recalled if for example their activation level was very high. But also the activation level has a relatively high influence ($\beta = 0.3$) on recall. The chance for events that have a low activation level—that are relatively "forgotten" (section 3.3.2)—to be recalled is small, especially if they are in competition with other events that have a higher activation level (interference theory, see section 2.1.2). Events that are related to other events within a context dependent sequence of events (an episode), recorded with high emotional arousal and a high salience are frequently recalled due to its slower decaying activation. This further increases the chance for subsequent recall (section 3.3.2).

Each event is related to the episodes it is part of (figure 3.5). If the recalled event belongs to one or more episodes, these are further accessible for recall. For example the agent recalls *event 4* in figure 3.5. As this specific event belongs to both *episode 1* and *episode 2*, these episodes including the associated scenario templates are further recalled. By traversing forward or backward originating from the recalled event with the supported temporal retrieval queries (section 3.4.2), the agent is able to reconstruct the particular episodes (figure 3.11). As all events are stored chronologically ordered in memory, the agent is also able to recall events that are beyond the limits of the recalled episodes. Based upon the recalled episodic

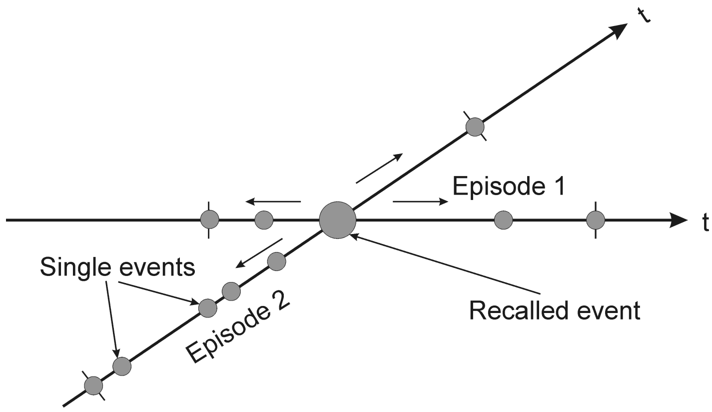

Figure 3.11: Reconstruction of episodes originating from a recalled event. The arrows indicate that the respective episodes can be traversed forward or backward.

information represented by the series of events and the recalled semantic information represented by the scenario template (the context of the episode), the agent is able to anticipate future sequences of situations and their impact. As each episode is rated by the emotional impact it caused to the agent, this can further be recalled and used for situation evaluation.

Tulving states that what is recalled from EM is not only the stored episodic information which is found by the retrieval mechanism. Informations of the retrieval cue is also involved in the recollection of events. Recollective experience is a process which combines episodic information from EM and semantic information from the retrieval cue. This synergistic effect, which complements the recalled engram by information from the retrieval cue, Tulving calls ecphory (section 2.1.2). Anyway, the synergistic process of ecphory is currently not implemented in the EM module, because there has not been found a suitable way to combine this information in the EM module. At the moment, the event with the highest retrieval result and the corresponding episode including its associated scenario template is recalled and not complemented by information from the retrieval cue.

Evoking an event due to a retrieval cue tends to change the stored information about the event and tends to make it more likely to be retrieved to a similar stimulus (section 2.1.1). This recoding of the recalled engram can be seen as a further consequence of ecphory. As depicted in section 3.2.3, recoding effects that change the composition of the features of stored engrams are not considered in the EM module. But what is influenced by the recollection of an event is its activation level. The activation of a recalled event receives a boost that increases the chance for subsequent retrieval to the same or similar cue. Furthermore, the episodes the recalled event is related to, are also strengthened (section 3.3.2).

The *act-of-remembering*, described by Tulving's General Abstract Processing System (section 2.1.3), which starts with the perception and encoding of an event, now ends with recollective experience of an stored event (figure 2.3). The agent remembers a past experience in the form of a recollection of this single event or an episode as a sequence of events. With this recollection, the agent is able to better evaluate the current situation and may additionally be capable to anticipate future sequences of situations. Current behaviors may be analyzed by evaluating their (emotional) impact based on past experiences. Memory performance—the use the EM has to the agent to improve its behavior—heavily depends on the formation of an appropriate retrieval cue and the kind and amount of stored events and episodes. A retrieval cue that is too general—i.e. a cue that is unexactly specified by too less selected features—and a big amount of events stored in memory would imply that a lot of memories match to this query and thus an improper event could be recalled. The retrieval of this event further leads to a boost and increases its chance for subsequent recall. A too specific retrieval cue—i.e. a lot of features selected in the cue—and too less events and episodes stored in memory may imply that each memory matches only a small part of the features indicated in the cue. This might lead as well to equalized retrieval results where an improper event could be recalled. Thus, memory performance is best, if the amount of stored events and episodes is pretty high and the retrieval cues are sufficiently specific. Then useful memories gain advantage over less important memories because of their activation level and the use to the agent is high.

3.5 Summary

An event based approach implementing the fundamental functional stages of human episodic memory processing—encoding, storage, and retrieval—is proposed in this chapter. A perceived situation—which is composed of various features—is encoded as an event, when something significant changes in the situation. A salient feature selection is made to measure significant changes and to highlight the relevant part of an event. This selection further establishes the compatibility relation between the stored event and the retrieval cue according to Tulving's encoding specificity principle. Events represent the prototypical memory units and are stored chronologically ordered. An episode consists of a sequence of events which happens in a certain context and is emotionally rated. Episodes are stored according to their semantic contents (the context in which they happened). A forgetting mechanism is implemented by attaching an activation level to the stored events and episodes. The activation decays by an exponential rate, which depends on the emotional involvement of the agent to the event and the salience of the event. Recalling an event or episode further boosts its activation. A retrieval can either be deliberate or spontaneous and is initiated by the formation of a retrieval cue. Spontaneous retrieval is evoked each time an event is encoded (reminding). Therefore, the salient features of the encoded event serve as cue. Which event is recalled depends and the amount of the matching of features and the activation level of the stored event. The episodes that are related to the retrieved event are further recalled.

Chapter 4

Technical Implementation

The episodic memory (EM) module is designed to be integrated into the psychoanalytical model of the "Artificial Recognition System" (ARS) (section 2.2). Java is the programming language for the implementation of the psychoanalytical model because of its platform independence, universality, speed, and simplicity of implementation. Apart from some hardware realizations (the "Smart Kitchen" [Rus03] or the "Tinyphoon robot" [MON05]), a simulation environment is used to evaluate the psychoanalytical model. The simulation environment—called the "Bubble Family Game" (section 2.2.2), is implemented in AnyLogic, a Java-based simulation framework. AnyLogic provides various libraries that support agent-based simulations. Thus, AnyLogic is chosen as development environment and Java as programming language for the EM module.

The further description of the Java implementation of the EM module is done by dividing the system into functional units. The description of the Java classes is split into blocks representing these functional units. The first section of this chapter provides an an overview to the implementation of the EM module. It describes the decomposition of the EM module into the sub-blocks alarm system, memory unit, and recall unit, how they interact, and how the EM module is used. The second section of this chapter deals with the functional unit alarm system. Its task is to monitor all incoming data to detect when an event or episode has to be encoded and spontaneous retrieval is triggered. The memory unit, which is the topic of the third section of this chapter, stores and maintains the events and episodes. It is composed by a memory container and an episode container which provide the basic operations on the events and episodes like *encode* or *retrieve*. The derivation of retrieval results of the stored events to a certain retrieval cue on the basis of matching of features and the activation level is further considered. The issue of the final section of this chapter is the recall unit, which provides an interface to access and recall the events evoked by the retrieval procedure. It describes what information about the retrieved event is recalled and how episodes can be reconstructed.

4.1 System Overview

In the first part of this section, the division of the implementation of the EM module into the functional units alarm system, memory unit, and recall unit is described. This breakdown into blocks follows the top-level structuring of the EM module into Java classes representing these functional units in the second part of this section. The interface to the EM module is further considered in detail. In the final part of this section, the structuring of the data units used in the EM module is considered.

4.1.1 Functional Units

The EM module is divided into the units *alarm system*, *memory unit*, and *recall unit* (figure 4.1). The structuring of the EM module into these three units is made based upon their functionalities. The input to the EM module covers all data gathered about the currently perceived situation. Furthermore, the progress of the scenario recognition processes is handed over via the Store API (Application Programming Interface) (section 4.1.2).

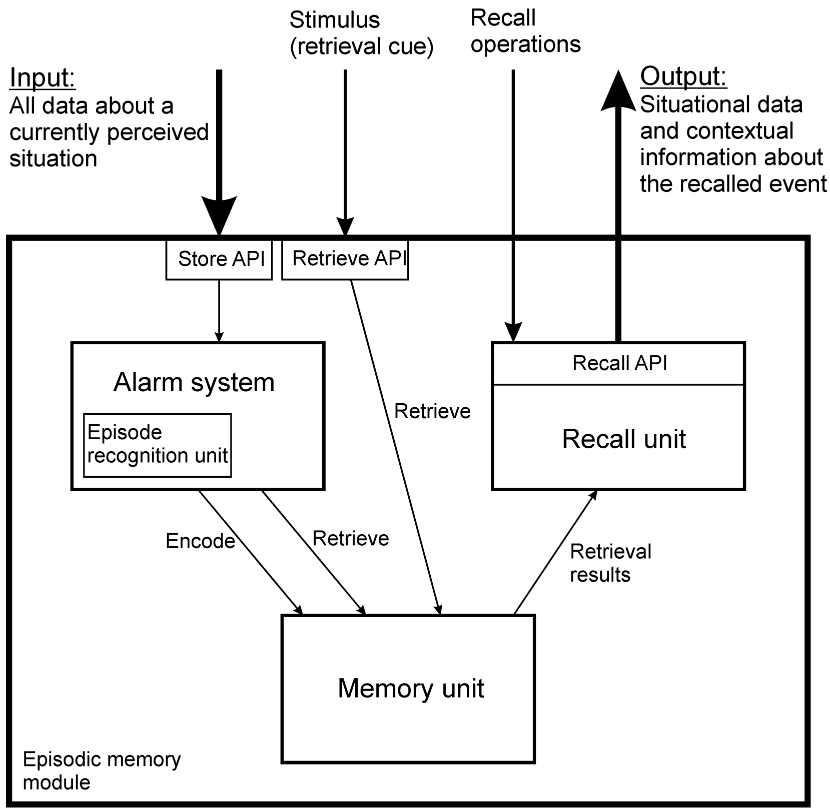

Figure 4.1: Schema of the EM module

The task of the alarm system (see section 4.2) is to process and to monitor all incoming data. It detects when an event has to be recorded, forces its encoding into the memory, and decides when spontaneous retrieval is triggered—it brings the system into retrieval mode. Therefore an appropriate retrieval cue is formed. The alarm system further incorporates an *episode recognition unit*. It checks, if an episode as a realization of a scenario template is accomplished and initiates its encoding into the memory.

The memory unit (see section 4.3) is the central part of the EM module. It deposits and maintains the stored events and episodes. It supports the basic operations like *encode* and *retrieve* on the stored events and episodes. Encoding is initiated by the alarm system if it is decided that an event or episode has to be recorded. A retrieval is either initiated by the alarm system (spontaneous retrieval) or by a certain stimulus (retrieval cue) from outside the memory system (deliberate retrieval) via the Retrieve API (see section 4.1.2). It performs the matching procedure to the given retrieval cue and derives a retrieval result on the basis of matching of features and the activation level for each event. The output of the memory unit is a container with the retrieval results after executing the retrieval procedure. This container is delivered to the recall unit. The memory unit further handles some internal processes on the memories like the decay of the stored events and episodes or some recoding effects. Recoding of the stored events occurs when episodes are encoded or events are retrieved by relating the events with the corresponding episodes and boosting their activations.

The recall unit (see section 4.4) provides an API (Recall API) to access the retrieval results determined by the retrieval procedure of the memory unit. The functionalities supported by the recall unit make it possible for the agent to step back into its past by recalling appropriate events or reconstructing whole episodes. The output of the EM module covers the situational data about the recalled events which are selected with the operations supported by the recall unit. Contextual information about the recalled event—the scenario template in which the event happened—is further accessible (section 4.4).

4.1.2 Interface to the EM Module

The basic structure of the Java implementation of the EM module is illustrated in figure 4.2. The interface to the EM module is represented by the class clsMemory[1]. It provides the interface to continually deliver the gathered data about the current situation to the EM module and to initiate a deliberate retrieval. Retrieved events and episodes (triggered either spontaneous or deliberate by the agent) are returned and the data about these experiences are further accessible for recall. The classes in figure 4.2 represent the main classes

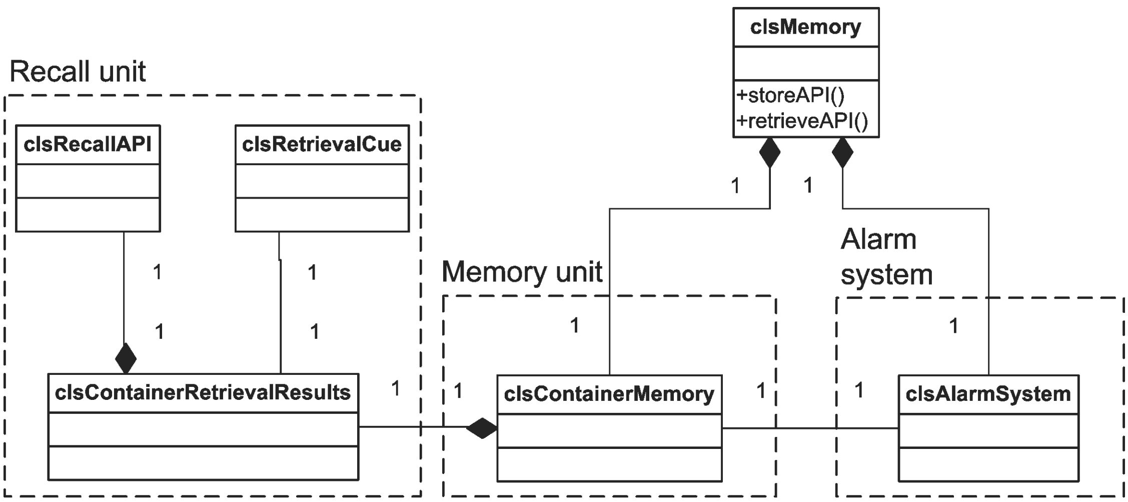

Figure 4.2: Interface to the EM module: clsMemory is the main class of the EM module

of the functional units depicted in figure 4.1. The alarm system is represented by the class clsAlarmSystem and the memory unit by the class clsContainerMemory. The recall unit is constituted by the container with the retrieval results (clsContainerRetrievalResults) and the

[1]Class names are further denoted with the prefix “cls” which is an abbreviation for “class”.

Recall API (clsRecallAPI). If a retrieval—initiated either by the alarm system or explicitly by the agent—has been processed by the memory unit, it returns the retrieval results of a certain retrieval cue (clsRetrievalCue) to the recall unit. The agent is then able to access and recall the results determined by the retrieval procedure via the Recall API.

Memory Operations

The class clsMemory of figure 4.2 serves as the interface of the EM module. It supports the API's to add data to the EM module (Store API) and to deliberately retrieve previously experienced events and episodes (Retrieve API). These API's provide the usage of the EM module and are further considered in detail.

`StoreAPI`— The data about the currently perceived situation is continually handed over to the EM module via the Store API. It is delivered in the specific data structures defined in the implementation of the psychoanalytical model of the ARS project. These cover the following container:

Container Drives —This container consists of a list of the individual drives. Each drive is represented by the data structure defined for a drive. The drives in the current implementation are *hunger, seek*, and *play* (section 2.2.3).

Container Emotions —It consists of a list of the individual emotions. Each emotion is represented by the data structure defined for an emotion. The emotions in the current implementation are *fear, anger*, and *lust*(section 2.2.3).

Container Template Images —This container consists of the individual template image-matches (TI). Each TI-match is represented by the data structure defined for a TI-match.

Container Actions —This container consists of a list of the currently executed actions. Each action is represented by the data structure defined for an action.

Container Recognition Processes —This is a structure that consists of the three scenario recognition process container `initialized`, `aborted`, and `recognized`. A scenario recognition process represents the progress of a certain scenario template (i.e. it indicates in which state a specific scenario template currently is). The container `initialized` consists of a list of the scenario recognition processes that currently are initialized, i.e. the first transition of a scenario template has triggered. The container `aborted` consists of a list of the recognition processes that currently are discarded due to a timeout condition and the container `recognized` consists of all recognition processes that are recognized in that situation—i.e. the last transition of a scenario template is triggered and the corresponding recognition process is thus completed. Scenario recognition processes are treated in more detail in section 4.2.2.

This data delivered to the EM module is directly handed over to the alarm system. The alarm system processes this data, appropriately initiates the encoding of an event or episode, and triggers a spontaneous retrieval. After the spontaneous retrieval is processed by the memory unit, a container with the retrieval results (clsContainerRetrievalResults) (section 4.3.3) is returned by the Store API. Otherwise, if no spontaneous retrieval is triggered, nothing is returned.

`RetrieveAPI`— It provides the interface to the agent for a deliberate retrieval. Before a deliberate retrieval is initiated by accessing the Retrieve API, an appropriate retrieval cue has to be constructed and handed over. The retrieval is processed by the memory unit which creates a container with the retrieval results. This container is further returned by the Retrieve API, if an appropriate retrieval result to the indicated cue has been found.

`RecallAPI`— The retrieval results container, which is returned by the Store API or the Retrieve API as a result of the retrieval process, consists of a list of the retrieval results in an ascending order (the best retrieval result first). Each retrieval result corresponds to one specific event. The container further incorporates a Recall API, via which the data of the events represented by the individual retrieval results can be accessed and recalled. The handling of the operations provided by this container and the Recall API to recall events and to reconstruct whole episodes is depicted in section 4.4.

Retrieval Cue

Before a deliberate retrieval is initiated via the Retrieve API, an appropriate retrieval cue has to be constructed. It can be constructed either by an event as cue or by an arbitrary number of features. An event is used for spontaneous retrieval whereas the arbitrary composition of features is usually used for deliberate retrieval. Additionally, a scenario template can be indicated in the retrieval cue. The retrieval cue is represented by the class clsRetrievalCue (figure 4.2). To construct a retrieval cue, the corresponding features have to be set appropriately. These are:

`Event`— If using an event as retrieval cue, the salient features of the event are used as cue. The data structure of an event (clsEvent) is given in section 4.1.3.

`ContainerDrives`, `ContainerEmotions`, `ContainerTImatches`, `ContainerActions`— If the retrieval cue is constructed by an arbitrary composition of features (the event is not set in the retrieval cue), the respective features to be cued for have to be indicated in the corresponding container.

`ScenarioId`— If it is additionally cued for a scenario template (optional), the corresponding id of the scenario template id has to be indicated. If cuing for a specific scenario, only the events related to episodes that are realizations of this scenario are traversed (section 4.3).

The basic operations provided by the interface of the EM module are illustrated in the sequence diagrams of figure 4.3 and figure 4.4. Figure 4.3 shows the procedure of adding data to the EM module via the Store API. In this particular case, the alarm system triggers the encoding of an event and a spontaneous retrieval. The retrieval cue is constructed by the alarm system and the retrieval results are returned in a container which further includes the Recall API to access the retrieved information.

Figure 4.4 illustrates a deliberate retrieval. A retrieval cue has to be set explicitly by the agent before initiating the retrieval procedure via the Retrieve API. The container with the retrieval results is returned and the data about the retrieved events can be accessed via the Recall API.

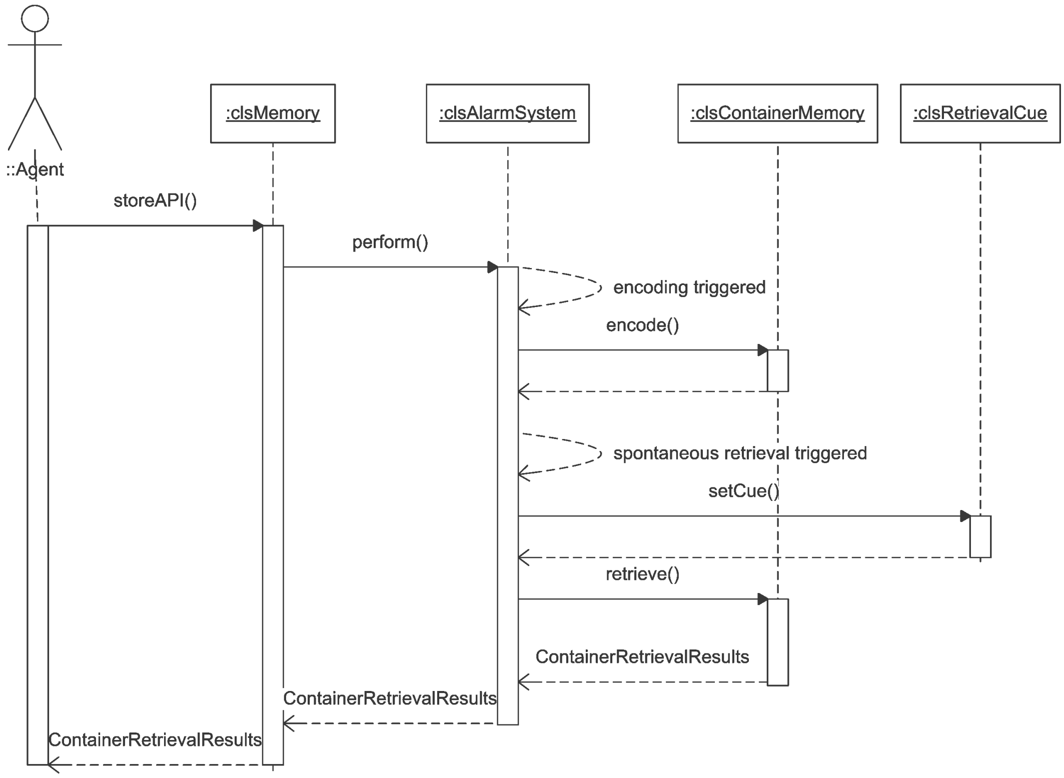

Figure 4.3: Sequence diagram illustrating the basic functioning of adding data to the EM module. An event is encoded and spontaneous retrieval is evoked.

4.1.3 Data Structure

The structure of the data that is continually delivered to the EM module via the Store API (4.1.2) has a fixed structure, i.e. the container emotions covers the individual emotions which are further represented by a predefined data structure. The fundamental processing (triggering encoding, matching) on this data in the EM module has to be done on the lowest level. On each individual emotion, drive, TI-match and action operations have to be made and additional information has to be added for the processing within the EM module. As the structures of the delivered data are fixed, a new data structure is made for the processing within the EM module.

The class diagram of the data structure is illustrated in figure 4.5. According to section 3.1.2, a situation (clsSituation) is decomposed into the features drives (clsFeatureDrives), emotions (clsFeatureEmotions), actions (clsFeatureActions), and TI-matches (clsFeatureCompareResults[2]) (figure 4.5). A situation consists of exactly one feature TI-matches, one feature actions, one feature drives, and one feature emotions. Some of the operations and properties are common for each individual feature and are defined in the abstract class clsFeature. Each feature further consists of various feature elements (clsFeatureElement) which must be of the same type. For example, the feature drives (clsFeatureDrives) can only contain drives (clsElementDrive) as feature elements. This implies that there must exist exactly one type of feature element for each type of feature. A feature element is the smallest data unit in the

[2]The notation of the Java objects and classes named "CompareResult" or "ImageCompareResult" refer to TI-matches.

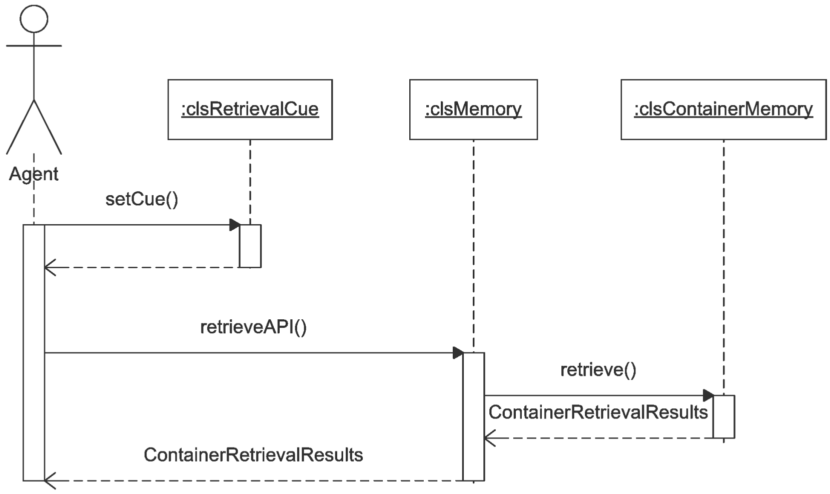

Figure 4.4: Sequence diagram of a deliberate retrieval

EM module and represents an extension of the individual data structures delivered to the EM module. Thus, the element action (clsElementAction) contains the data structure of an individual action (`Action`), the element emotion (clsElementEmotion) contains an individual emotion (`Emotion`), the element TI-match (clsElementCompareResult) contains an individual TI-match (`TImatch`), and the element drive (clsElementDrive) contains an individual drive (`Drive`). Properties that are common to each feature element are defined in the abstract class clsFeatureElement. A situation becomes an event (clsEvent), if something significant in the situation happens (see section 3.1.2). An episode (clsEpisode) is further a sequence of events that is a realization of a certain predefined scenario template. Episodes represent the highest order data units in the EM module.

Feature Element

A feature element is the smallest order data unit in the EM module. Each feature element (clsFeatureElement) gets a salience value. This salience value is derived at the time of triggering encoding (section 3.2.2) and its value is between zero and one. A salience of zero means that this feature element (e.g. this TI-match) is not relevant in the current situation whereas a high salience value close to one means that this feature element is very relevant or activated to the current situation. The salience further decays by a specific rate. Thus, a particular feature element may remain to a certain amount salient or activated in some succeeding situations, if it was recently highly salient (section 3.2.2). The operations that are common for all types of feature elements are defined in the class clsFeatureElement:

`triggerEncoding`— Defines, whether the encoding of an event based upon this feature element is triggered. Therefore, the particular feature element is compared the feature element of the previous situation. (e.g. the emotion anger is compared to the emotion anger of the previous situation). The triggering concept is discussed in section 3.2.1 and is based upon the salience of the particular feature element.

`getMatch`— Defines the match of the feature element to the feature element of the same type indicated in the retrieval cue. The amount of matching is individually determined by each respective feature element (see section 4.3.3).

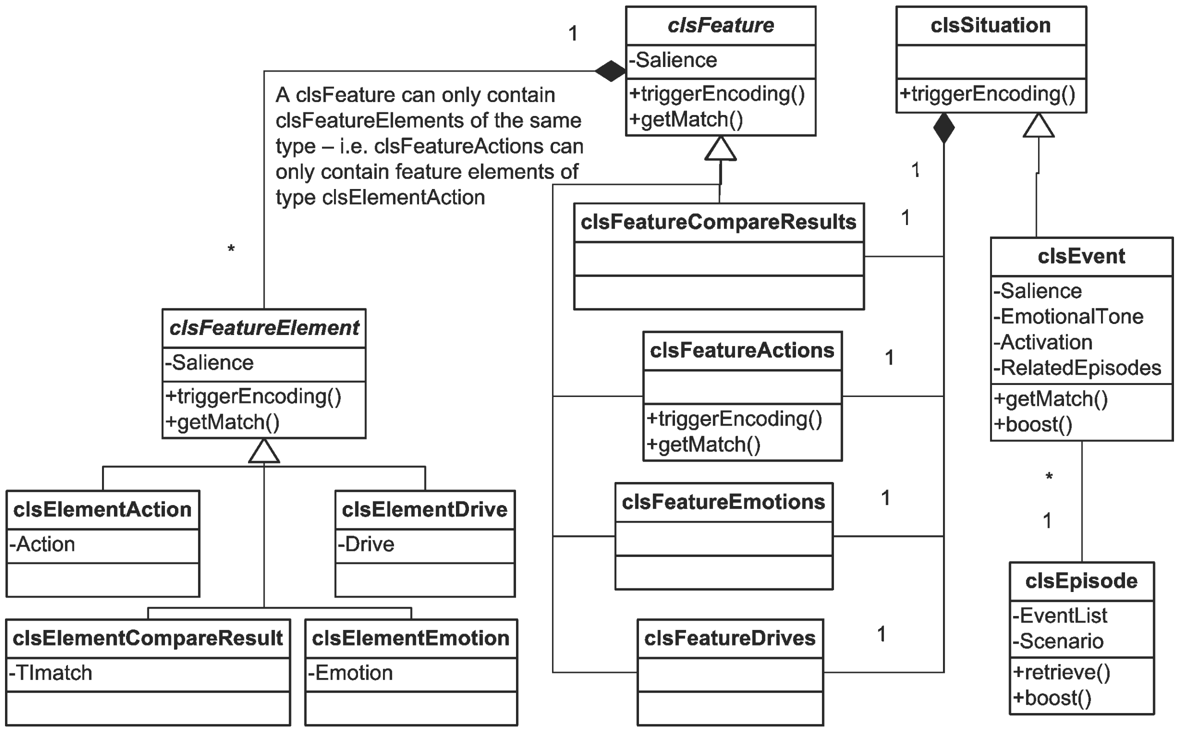

Figure 4.5: Class diagram of the data structure

Feature

The collection of feature elements of a particular type are mapped to the feature of the same type. Each feature (clsFeature) gets a salience value that is derived on the basis of the constituent elements of this feature at the time of triggering the encoding of an event. In the current implementation the salience of a feature is determined by the feature element having the maximum salience level (section 3.2.2). The operations that are common for all types of features are defined in the class clsFeature:

`triggerEncoding`— Determines whether the encoding of an event on the basis of the feature of the same type of the previous situation should be triggered. Therefore the operation `triggerEncoding` of each constituent feature element (i.e. each element emotion for the feature emotions) is called (see section 4.2.1).

`getMatch`— Determines the match of the particular feature to the feature of the same type indicated in the retrieval cue. The matching is based on the individual constituent matches of the feature elements determined by the operations `getMatch` of each feature element. If the retrieval cue is constructed by the salient features of an event, this operation is optimized on speed by making use of the fixed composition of feature elements of an entire event.

Situation

An entire situation (clsSituation) is composed by exactly one type of each feature. As the emotions, drives, and TI-matches consist of the same type of feature element each situation (i.e. the feature emotions is always composed of three feature elements anger, fear, and lust), the structure remains almost the same for each situation. Only the composition of the feature actions may be different for each situation. The matching for spontaneous retrieval is optimized using this fixed structure. The operation `triggerEncoding` determines, whether a situation has to be encoded as an event on the basis of each feature (see section 4.2.1).

Event

An event (clsEvent) is an extension of a situation and thus inherits properties from the class clsSituation. As defined in section 3.1.2, an event is a happening that arises at a particular time, when something in a situation changes significantly. Additionally to the "static" data of a situation (the features) an event consists of information about the happening within a situation. A salience value indicating its unusualness and the emotional tone of the agent to the event are attached. Each event has a certain activation indicating the accessibility for later recall. Furthermore it may be related to one or more episodes. As depicted in section 3.3.2, the activation of a stored event depends on the emotional arousal of the agent to the event (expressed by the emotional tone of the event), the event's salience and the relatedness of the event to one or more episodes. This dependency is expressed by the parameter T_0 of the activation function $A_e(t)$ (equation 3.13). The activation of the stored events represents a very crucial concept of the EM model. The parameter to adjust the activation function of an event can be adjusted in the class clsEvent. The initial activation level when an event is encoded (A_{e0}) and the amount of boosting (A_{b0}) when it is recalled have to be set appropriately. The most important operations of an event are:

`getMatch`— Derives a match for this event to the features indicated in the retrieval cue. The matching procedure is detailed in section 4.3.3.

`boosting`— Boosts an event with the activation A_{b0} if it is recalled (section 4.4.1). It further strengthens the episodes that are related to that event with the activation A_{b1}.

Episode

An episode (clsEpisode) is a chronologically ordered sequence of events that represents a concrete realization of a scenario template (see section 3.1.2). It is the highest order data unit in the EM module. An episode is composed of the three components:

Context — The semantic information of an episode in the form of a scenario template is denoted as the *context* of the episode.

Content — The series of events from the initialization of a scenario recognition process (section 4.2.2) until its completion is recorded and stored as the *content* of an episode.

Impact — The emotional *impact* the episode has to the agent is further accessible and can be used to evaluate the outcome of the episode.

The most important operation of an episode is `retrieve`. It determines the event with the best retrieval result of the particular episode on the basis of matching of features and the activation level (see section 4.3.3). The data structure of the EM module depicted in this section from the highest order data unit—an episode—to the lowest order unit—a feature element—is summarized in figure 4.6.

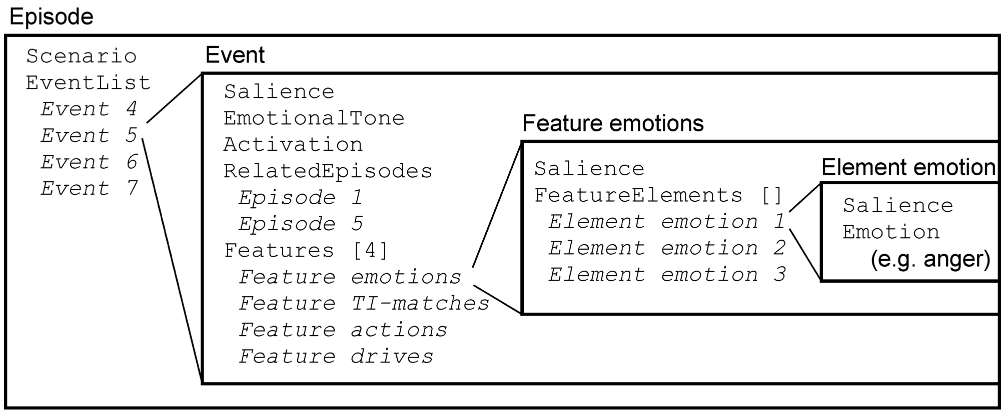

Figure 4.6: Data structure within the EM module

4.2 Alarm System

The task of the functional unit alarm system is to monitor the incoming stream of data, to detect, when the encoding of an event has to be triggered, and when spontaneous retrieval should be initiated. This is considered in the section 4.2.1. Further it has to recognize, when an episode has been completed and its encoding should be initiated (section 4.2.2). These functionalities are controlled by the class clsAlarmSystem (figure 4.7). It checks the

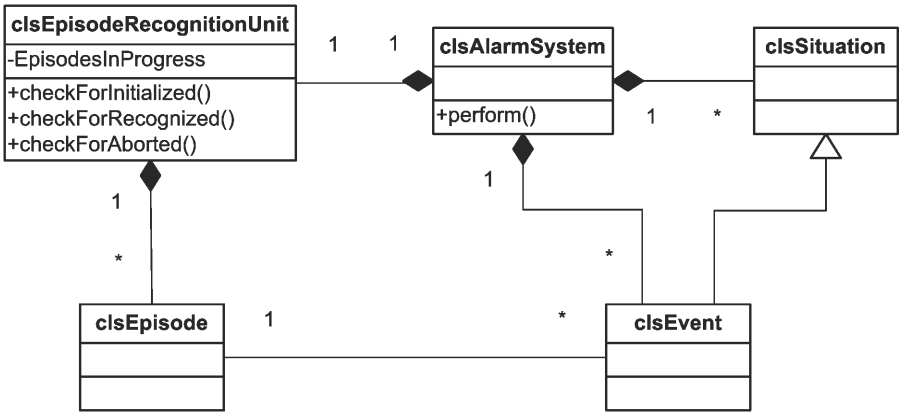

Figure 4.7: Class diagram of the functional unit alarm system

incoming data representing situations and appropriately encodes an event, if something significant happens in a situation. The alarm system further contains an episode recognition unit (clsEpisodeRecognitionUnit) that manages the delivered scenario recognition processes and appropriately encodes an episode. The data processing in the alarm system is performed in the following manner. The continually delivered data about a perceived situation (drives, emotions, TI-matches, and actions) is adapted to the data structure within the EM-module (section 4.1.3). Thus the data is mapped into the corresponding feature elements and features which further compose a situation. The current situation is compared to the temporally stored previous situation and it is determined whether an event is encoded. In this case, the situation becomes an event and is encoded into the memory container. Furthermore, spontaneous retrieval is triggered if an event is encoded, which is called *reminding* (section 3.4.1).

The retrieval is processed by the memory unit and the result is a container with the retrieval results (see section 4.3.3), which is returned by the alarm system. Further, the processing of the episode recognition unit is controlled and episodes are appropriately encoded.

4.2.1 Triggering Encoding and Spontaneous Retrieval

The encoding of an event is triggered if something in a situation changes significantly (section 3.1.2). Thus, all features and their consisting elements of the current situation are traversed by comparing them with the previous situation to detect a significant change. The encoding of an event further implies that spontaneous retrieval is released. It is searched for a similar event to the one currently encoded (*reminding*). A particular procedure of triggering the encoding of an event and the spontaneous retrieval of a similar event is illustrated in the sequence diagram of figure 4.8.

The continually delivered data is packed into the data structure situation and the operation `triggerEncoding` of this situation is called. Each feature of the situation (TI-matches, emotions, actions, and drives) is checked whether encoding should be triggered. The constituent feature elements of each feature are further checked. The salience of each feature element is determined and encoding is released on the basis of the salience of a feature element. If the salience of any feature element is high enough, the corresponding feature gets salient and thus the current situation is encoded as an event. In the illustrated procedure, encoding is triggered because of one or more emotions and actions. Further, the salience of each feature is derived based on the salience values of the feature elements. The situation becomes an event and its encoding into the memory container is released (section 4.3). Spontaneous retrieval is initiated with the currently encoded event as retrieval cue. The container with the retrieval results of the spontaneous retrieval (reminding) is returned. The agent is able to access and recall the events referring to the corresponding retrieval results via the Recall API which is incorporated in the retrieval results container (see section 4.4).

4.2.2 Episode Recognition

The task of the episode recognition unit is to detect when the agent experiences an episode as a realization of a scenario template. It manages the scenario recognition processes continually delivered to the EM module and indicates if an episode has to be encoded into the episode container. As mentioned in section 2.2.3, the scenario recognition processes represent the progress of a certain scenario template (i.e. it indicates in which state a specific scenario template currently is). Scenario recognition processes may run parallel. Thus, at a particular moment in time, various recognition processes may be in progress of the same or different scenario template, each in a different state. For example, the recognition processes for the scenario templates *search and consume energy source* and *help to crack energy source* may be in progress at the same time. A recognition process may further be discarded, if a certain abort condition is fulfilled (e.g. a temporal condition which defines the maximum time span until which the succeeding state of a scenario template must be reached). Three different container of scenario recognition processes are delivered to the EM module and further to the episode recognition unit. One contains the processes that are currently initialized, the second the aborted, and the third the completely recognized processes.

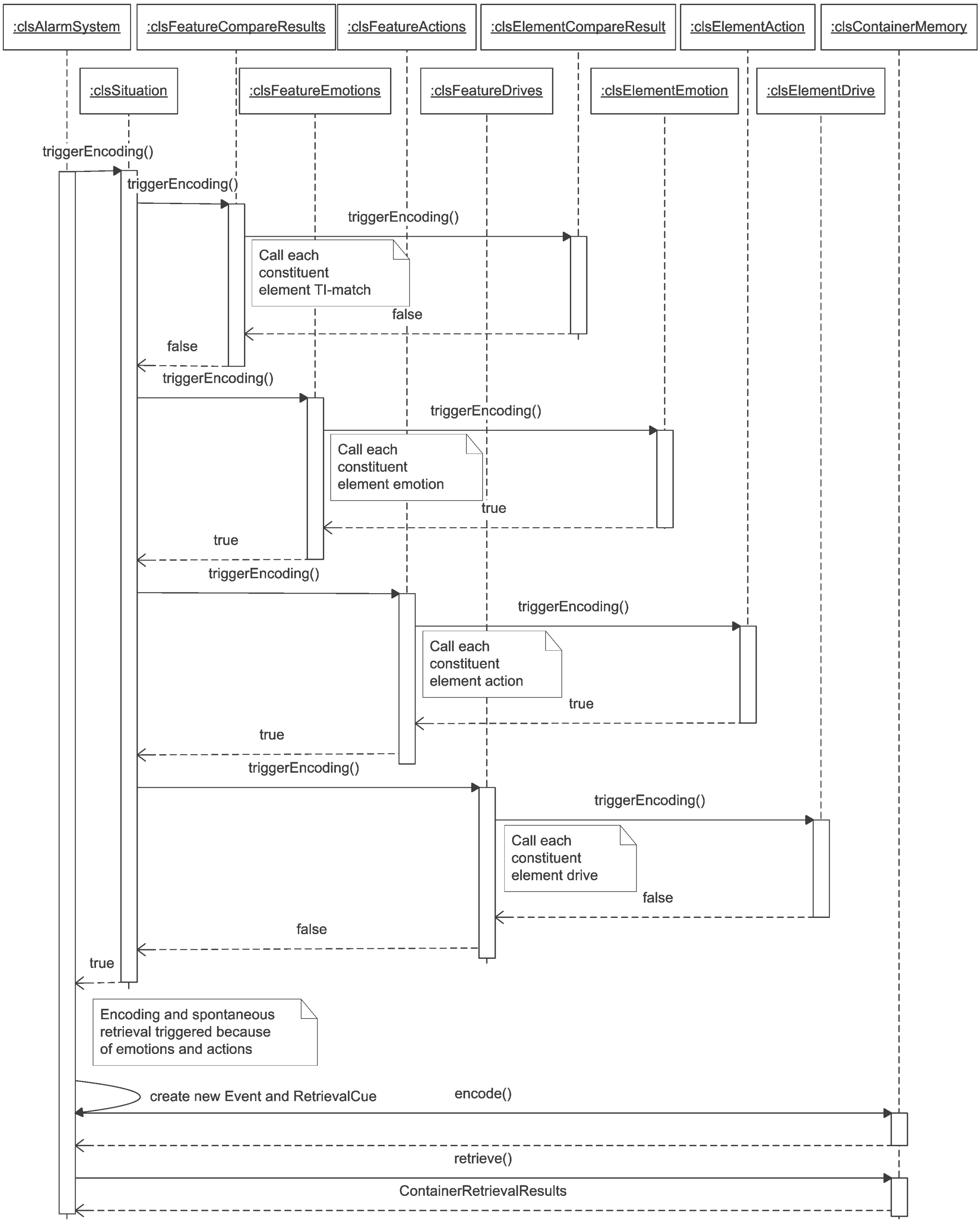

Figure 4.8: Sequence diagram of a specific procedure trigger encoding. The encoding of an event and the spontaneous retrieval of a similar event is triggered on the basis of the features emotions and actions.

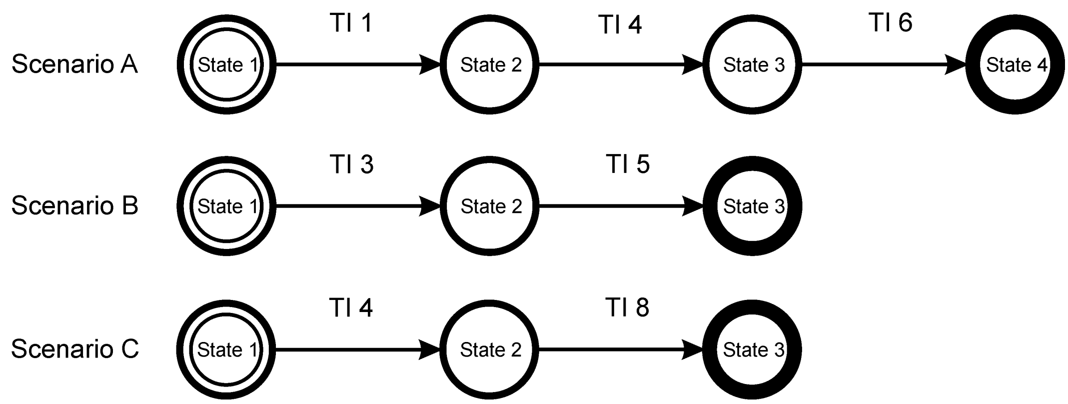

Figure 4.9: Scenario templates

The processing of the episode recognition unit on the basis of three different scenario templates (figure 4.9) is illustrated in figure 4.10. A new recognition process of a particular scenario template is initialized, if the first transition of this scenario template is triggered. Event 10 in figure 4.10 arises at a particular time having two salient template images: TI 3 and TI 4. The salience in this case comes along because of their high match-level. The occurrence of TI 3 triggers the first transition of scenario B (figure 4.9) and the recognition process 3 is initialized. Furthermore, TI 4 triggers scenario C and initializes recognition process 4. Each recognition process contains a link to the corresponding scenario template (there may run more recognition processes parallel for one scenario template) and a list of the yet recognized states. To each initialized recognition process, a new episode is created by the episode recognition unit and stored into the temporary container *EpisodesInProgress*. Episode 2 is created for the recognition process 3 and episode 3 for recognition process 4. The start event—the event that forced the initialization of a scenario recognition process—is registered into the created episodes. The end event—the event that triggers the last transition of a scenario template and thus finalizes a recognition process—remains *null* which indicates that the episode is in progress and not yet completed. Additionally there exist episode 1 in the temporary container that was created by the initialization of recognition process 1 due to the occurrence of event 6.

In the succeeding events 11, 12, and 13 no scenario recognition process container is delivered to the episode recognition unit which implies that no recognition process is initialized, aborted, or completely recognized. With the occurrence of event 14 two recognition processes are completely recognized. TI 6 brings the recognition process 1 of scenario A from state 3 into state 4 and TI 5 triggers the last transition of scenario B and brings the recognition process 3 into the terminal state. The end event (event 14) is registered into the corresponding episodes (into episode 1 for recognition process 1 and into episode 2 for recognition process 3). Episode 3 remains in the temporary container until it is completed or removed due to a timeout condition. The episode recognition unit returns the two recognized episodes to the alarm system which further initiates their encoding.

The episode recognition unit is implemented by the class clsEpisodeRecognitionUnit (figure 4.7). It consists of a temporary container that deposits the episodes in progress. An episode in progress is an episode of a scenario recognition process that has been initialized but not yet completed. If a recognition process is completely recognized, the corresponding episode from the temporary container will be stored to the episode container (see section 4.3.2).

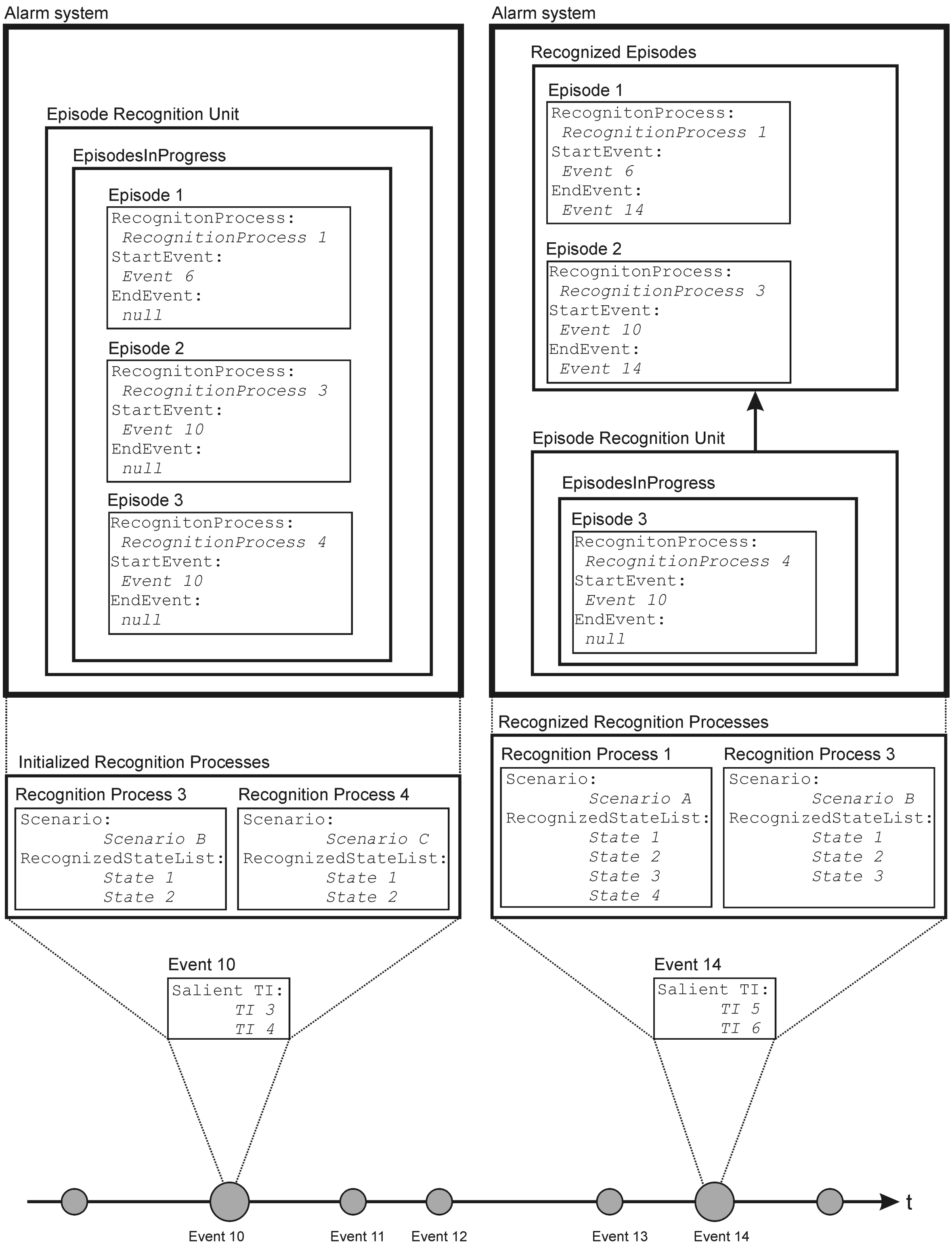

Figure 4.10: Processing of the episode recognition unit

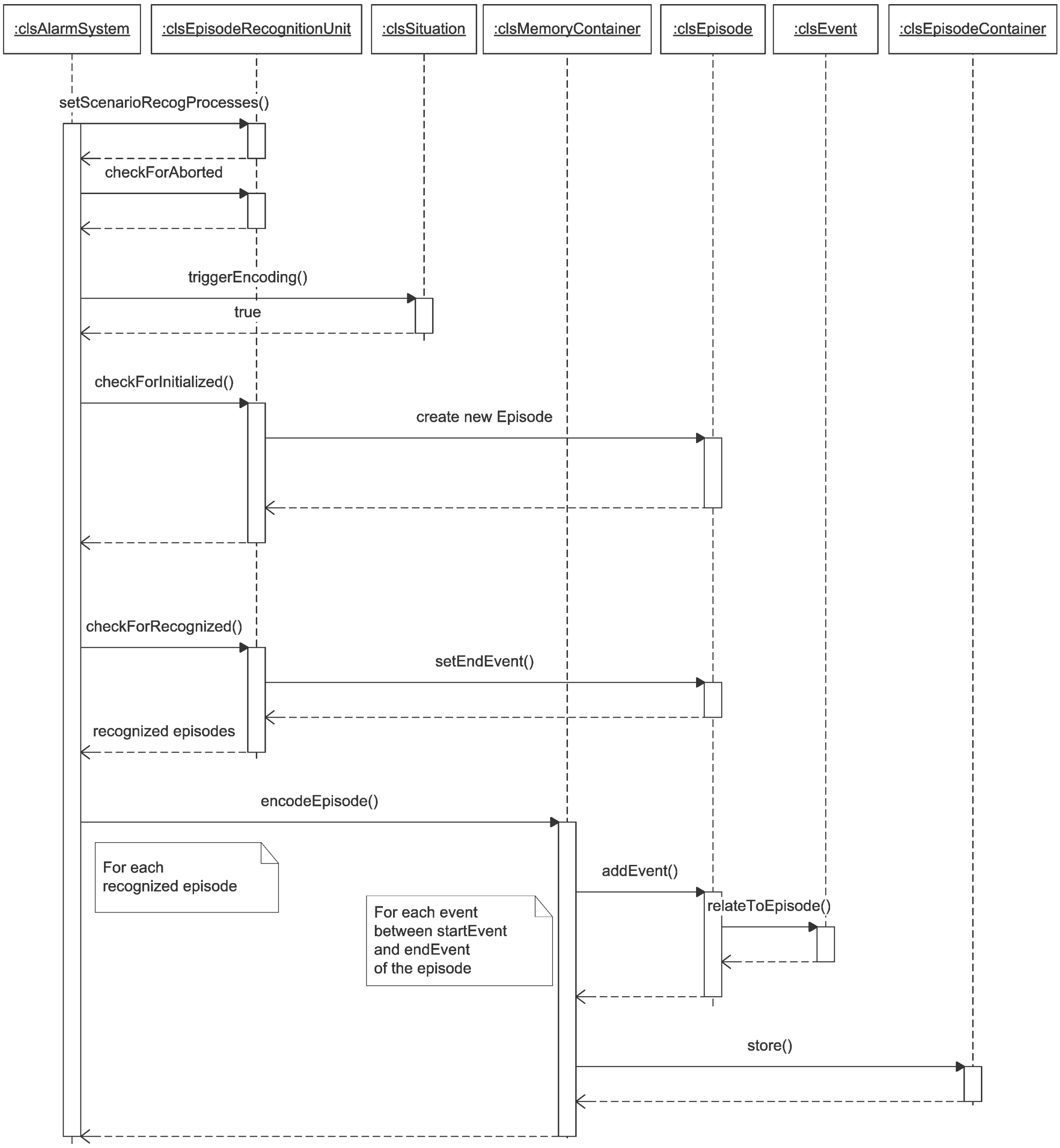

Figure 4.11: Sequence diagram of the scheduling of the episode recognition

The sequence diagram in figure 4.11 illustrates the scheduling of the episode recognition unit. Each process cycle, the initialized, aborted, and recognized processes are delivered to the episode recognition unit (`setScenarioRecogProcesses`). Aborted recognition processes (due to a timeout condition) are removed from the temporary container (`checkForAborted`). If the encoding of an event is triggered and there exists an initialized scenario recognition process (`checkForInitialized`), a new episode is created and the start event is set to the currently encoded event. If there are some recognized processes (`checkForRecognized`), the end event of the corresponding episodes is set and returned to the alarm system. The alarm system forces the encoding of the recognized episodes into the memory container. The encoding of this episode, which is the task of the memory unit, is further illustrated in figure

4.11. For each recognized episode, the series of events within the realization of this episode is related to the episode (`addEvent`). The link of an event to a certain episode is further indicated in the stored event (`relateToEpisode`). This implies, that the activation of the events related to episodes decay slower (section 3.3.2). After the events and episodes are related to each other, the episode is stored into the episode container.

4.3 Memory Unit

The memory unit is the central part of the EM module. It consists of the memory container and the episode container which deposit and maintain the events and episodes. As pointed out in section 3.3.1, the memory container contains all stored events considering its chronological order. The episode container further consists of sub-containers, each depositing the episodes corresponding to the context in which they happened (the scenario template) (figure 3.5). The memory unit further provides the basic functionalities for the processing of the episodic memories. These are encoding and retrieval of the events and episodes. The dynamics of the memories—their changes over time—are further regarded by the memory unit. The class

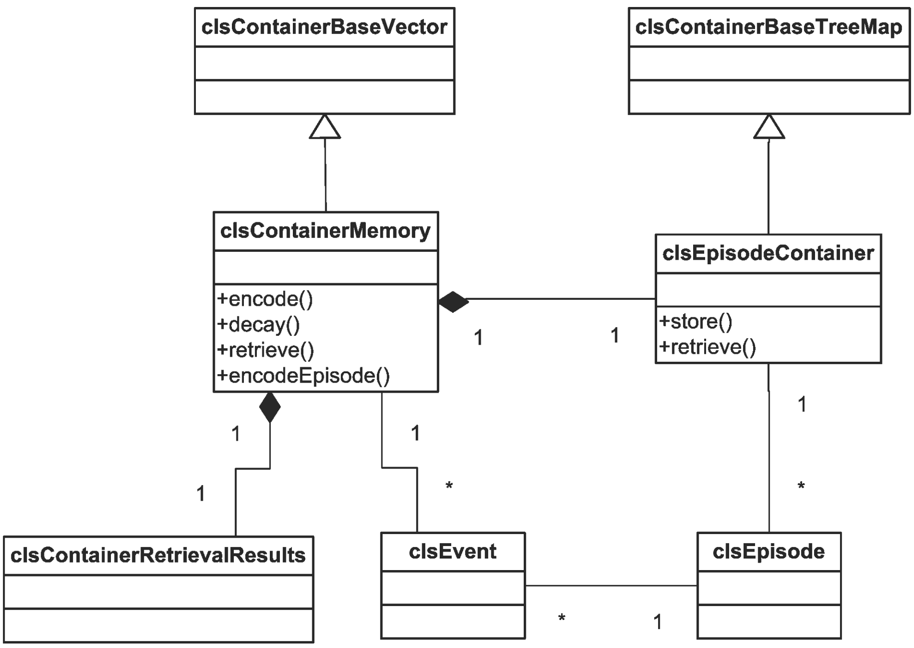

Figure 4.12: Class diagram of the functional block memory unit

diagram of the memory unit is illustrated in figure 4.12. Events (clsEvent) and episodes (clsEpisode) that are recognized by the alarm system are stored to the memory container (clsContainerMemory) and episode container (clsEpisodeContainer). The activation of the events decay at different rates (corresponding to their emotional tone and salience as well as the number of episodes it is linked to), which influences their accessibility for later recall. If a retrieval is initiated (spontaneously by the alarm system or deliberately by the agent), it will be processed by the memory unit, the results (see section 4.3.3) are written into a container (clsContainerRetrievalResults), and delivered to the recall unit (figure 4.1).

In the first part of this section the memory container is described. It represents the depot of the events which are stored chronologically ordered. The fundamental operations like `encode` or `retrieve` are supported for encoding events into the memory container (initiated by the alarm system) or retrieving them. The episode container, which is part of the memory container, is further considered in the second part of this section. It contains the episodes that have been recognized as realizations of scenario templates by the alarm system. It further supports basic operations for encoding and retrieving episodes. In the final part of this section the determination of retrieval results of the stored events as a result of the retrieval procedure can be found. Special consideration is spent to the implementation of the similarity relation between the retrieval cue and the stored event, which is matching of features in the EM module.

4.3.1 Memory Container

The memory container is the depot of the chronologically ordered events. As depicted in section 3.1.2, an event is the prototypical unit of the EM module. An event precedes one and succeeds another. A well accessible data structure is required to provide high performance in operation. A sorted list has been chosen, where the events are stored chronologically indexed. If a new event is stored into the EM, it will be appended to the end of the list. Once an event is stored, it will never be removed from memory. An event always remains available in memory, but it may become inaccessible due to a very low activation level (section 3.3.2).

As illustrated in figure 4.12, the memory container (clsContainerMemory) inherits properties from the class clsContainerBaseVector. This represents a predefined list structure and supports additional functionalities on the list elements. The fundamental operations provided by the memory container are:

`encode`— Sets the salience, the emotional tone and the initial activation of an event and stores it into the memory container. It is usually called by the alarm system when the encoding of an event has been triggered (section 4.2.1). The salience is derived from the individual salience values of the constituent features. Each feature contributes the same share to the overall salience of the event (section 3.2.2). The emotional tone is determined by the individual emotion of the features emotions having the maximum emotional value. The emotional tone indicates the emotional arousal of the agent to the happening and is relevant for the decay of the activation of the event (section 3.3.2).

`encodeEpisode`— Adds events between the start and the end of the episode for encoding (see section 4.2.2). These events are further related to the episode. The episode is then stored into the episode container. The execution of this procedure is illustrated in figure 4.11.

`decay`— Performs the dynamics of the stored events over time. As described in section 3.3.2, the activation of the stored events decay over time depending on the emotional tone, the salience, and the number of related episodes. Thus, events decay at different rates. The higher the activation of an event, the higher its chance for subsequent recall. To accomplish the decay of the activation, all stored events are frequently traversed and their individual activation levels are updated.

`retrieve`— Traverses the stored events in the memory container and returns their results in a container (clsContainerRetrievalResults) when stimulating with a certain retrieval cue. If a scenario template is indicated in the cue, only episodes that happened in the context of the indicated scenario template will be passed through (see section 4.3.3).

4.3.2 Episode Container

The episode container consists of the episodes that are recognized by the episode recognition unit. It is embedded into the memory container, which stores the events. An episode is a sequence of events, that represents an experience of a known procedure. The knowledge represented by a scenario template is semantic. It is known, how a certain procedure has to be accomplished. For example, to accomplish the scenario *make coffee*, the following operational sequence might be stored: first the coffee has to be put into the coffee maker, then water has to be filled in, and the like. An episode is a concrete realization of such a pattern of a procedure that can be later remembered. As described in section 3.3.1, episodes may overlap or be nested in time—they may occur parallel. Episodes represent a more sophisticated data unit in the EM module compared to the single events as extended situations which are stored in a sequential order. As there does not yet exist relations among the elements of the features of an event, single events do not have semantic contents.

Episodes in the episode container are organized in sub-containers corresponding its context (the scenario template) (figure 3.5). Each sub-container consists of the particular episodes of one scenario template. Thus, the various sub-container are organized by their semantic content—the realized scenario template. An associative memory has been chosen for the organization of the episode container. It contains key-value pairs which are sorted by a tree according the key. Thus, each memory entry maps an object (value) to an unique key. The objects are the sub-container of episodes and the corresponding key is the id of the scenario template. Thus, the episodes are mapped to the id of the scenario template they are realizations of. Each sub-container of episodes is organized in a sorted set. It sorts the episodes according to the time of their completion. The episode that has been recognized most recently (which has the highest index of the end event) is the first entry in this structure.

As illustrated in figure 4.12, the episode container (clsEpisodeContainer) is embedded into the memory container (clsContainerMemory) and inherits properties from the class clsContainerBaseTreeMap, which represents a predefined structure of an associative memory with additional functionalities. The most important operations provided by the episode container are:

`store`—Stores an episode into the sub-container respectively the id of the scenario template it is realization of (figure 4.11).

`retrieve`—Performs a retrieval if a scenario template is indicated in the retrieval cue. Therefore, the episodes of the sub-container corresponding to the scenario id indicated in the cue are traversed. For each episode of this sub-container the event with the best retrieval result to the indicated features of the cue is determined and stored into the retrieval results container (clsContainerRetrievalResults) (figure 4.13). This container which consists of exactly one (the best) retrieval result for each episode is then returned.

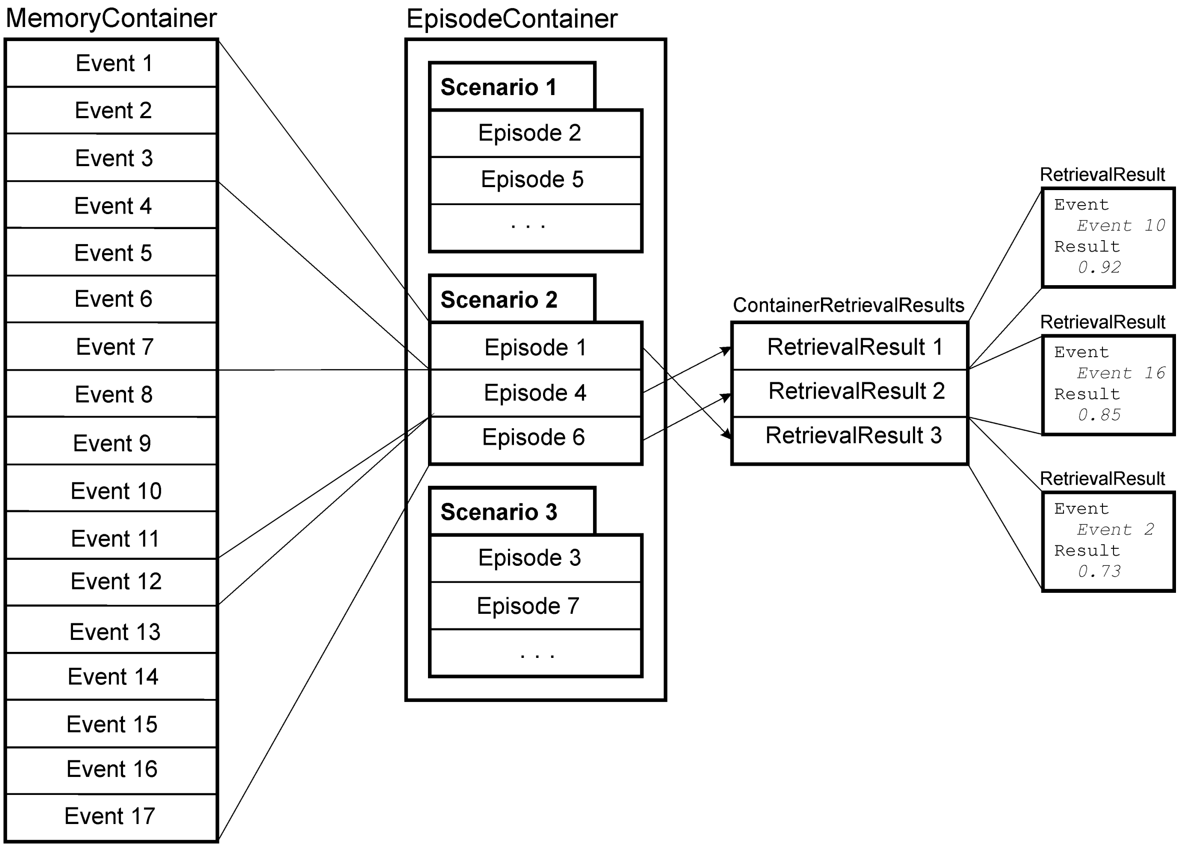

Figure 4.13: Retrieval procedure where the template *scenario 2* is indicated in the retrieval cue

The retrieval of an event with an scenario template (*scenario 2*) indicated in the cue is illustrated in figure 4.13. Only the events related to episodes of the sub-container for template *scenario 2* in the episode container are traversed. For each such episode (*episode 1, episode 4* and *episode 6*), the event with the best retrieval result is determined (see section 4.3.3) and stored into a container (ContainerRetrievalResults), which stores the retrieval result in an ascending order. *Event 10* of *episode 4* has the best result and will thus be recalled (section 4.4.1).

To get a better understanding of the scheduling of such a retrieval, a concrete retrieval procedure is further discussed. A retrieval is initiated by the formation of a retrieval cue either by the alarm system (spontaneous retrieval) or explicitly by the agent. The cue may either be constructed by an entire event as cue (only the salient features serve as cue) or by an arbitrary number of features. Optionally, a scenario template might be additionally indicated in the cue, which implies that only events of episodes that are realizations of the particular scenario template are traversed. The sequence diagram in figure 4.14 illustrates the scheduling of the retrieval operations with a scenario template indicated in the cue as discussed above (figure 4.13). The retrieval procedure is initiated by the `retrieve` operation of the memory container. As there is a certain scenario template indicated in the cue, only the episodes that are realizations of the particular scenario template are traversed. Thus, only the sub-container of the episode container which consists of the corresponding episodes is passed through. For each episode of this scenario template, all events related to this episode

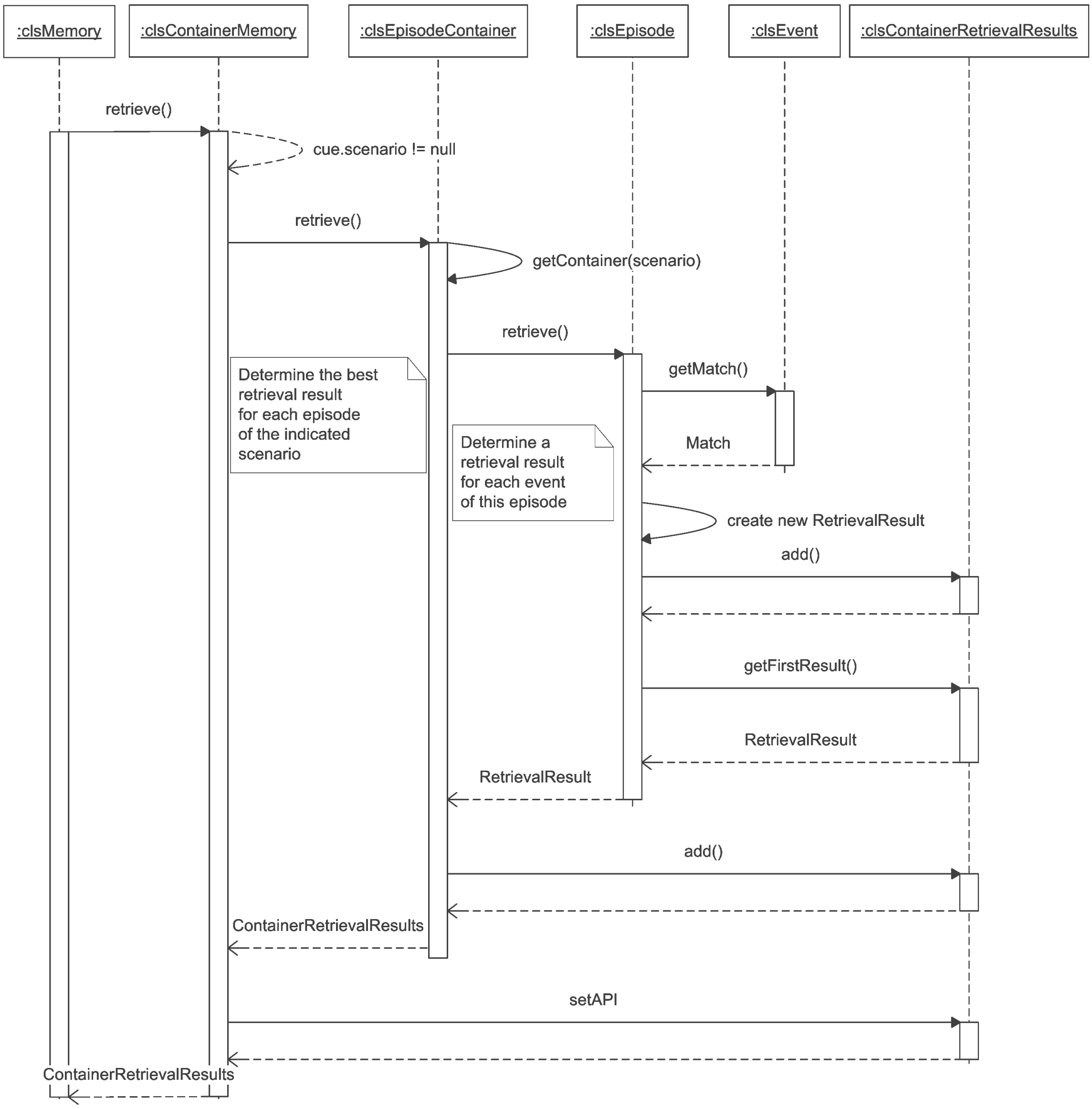

Figure 4.14: Sequence diagram of a retrieval with a cue containing a scenario

are traversed and a retrieval result for each event is determined on the basis of matching of features and the activation level (the matching procedure and the derivation of a retrieval result is detailed in section 4.3.3). The event of each episode with the best retrieval result to the indicated features in the cue is determined and entered into a container. The entries of this container are sorted corresponding to their retrieval results. This container is returned which further incorporates a Recall API. This provides the interface to the agent to access and recall the retrieved events (section 4.4).

4.3.3 Retrieval Results

Once, a retrieval is initiated by bringing the EM system into retrieval mode and constructing a retrieval cue, it has to be determined which event is evoked from the memory. The retrieval process traverses all stored events in the memory container (or rather the events belonging to certain episodes when cuing additionally with a scenario template) to derive a retrieval result for each event to the indicated retrieval cue. A retrieval result measures the relevance of the stored event to the cue, whether it will be recalled or not. It is derived on the basis of a similarity relation between the retrieval cue and the stored event and its activation (equation 3.24).

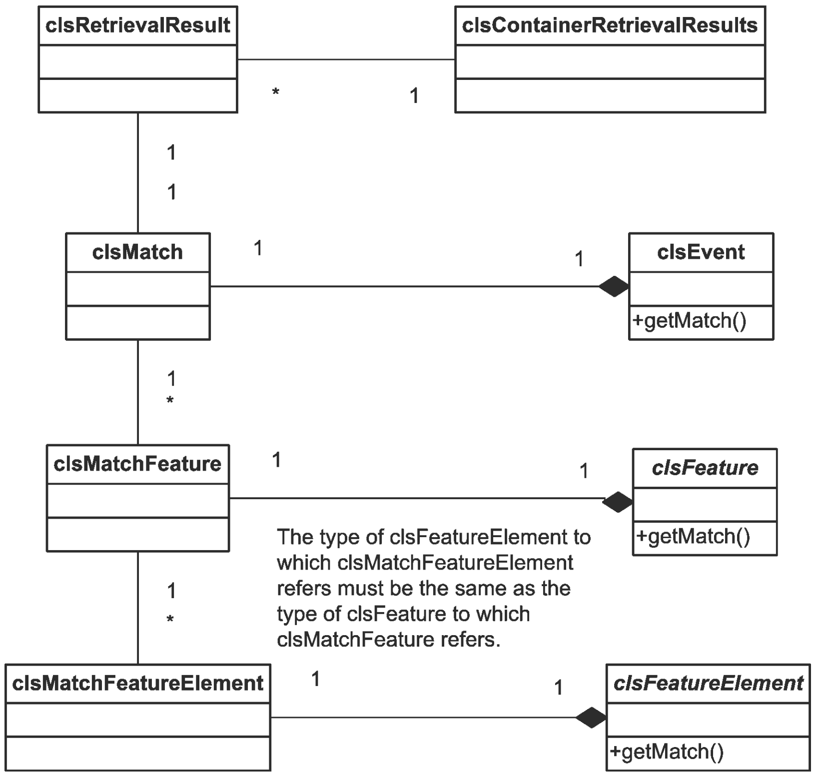

Figure 4.15: Hierarchical class diagram for the determination of a match of an event to a certain retrieval cue

The compatibility relation between the retrieval cue and the stored event is defined to be matching of features and is considered in detail in section 3.4.3. The matching procedure is realized bottom up (figure 4.15). For each feature element indicated in the cue, a match (clsMatchFeatureElement) for the corresponding feature element of the stored event is derived. The composition of the matches of the feature elements of the same type leads to a match of the feature (clsMatchFeature) and these matches are further fused to the overall match (clsMatch) of the event. The matching procedure is illustrated by a sequence diagram in figure 4.16. Consider a spontaneous retrieval, where the retrieval cue is constructed by the salient features of the currently encoded event. Imagine, the feature actions and the feature TI-matches are salient and the same features of the stored event are salient too (for spontaneous retrieval, only the salient features of the stored event are accessible (section 3.4.3)). Thus, for each feature element indicated in the retrieval cue, a feature element match is determined and added to the feature match of the same type of feature (i.e. the feature element matches of the individual actions are added to the feature matches of the feature actions). Each feature match is then added to the overall match of the event. Note that the emotional state is also considered for matching, even if the feature emotions of the encoded event is not salient and

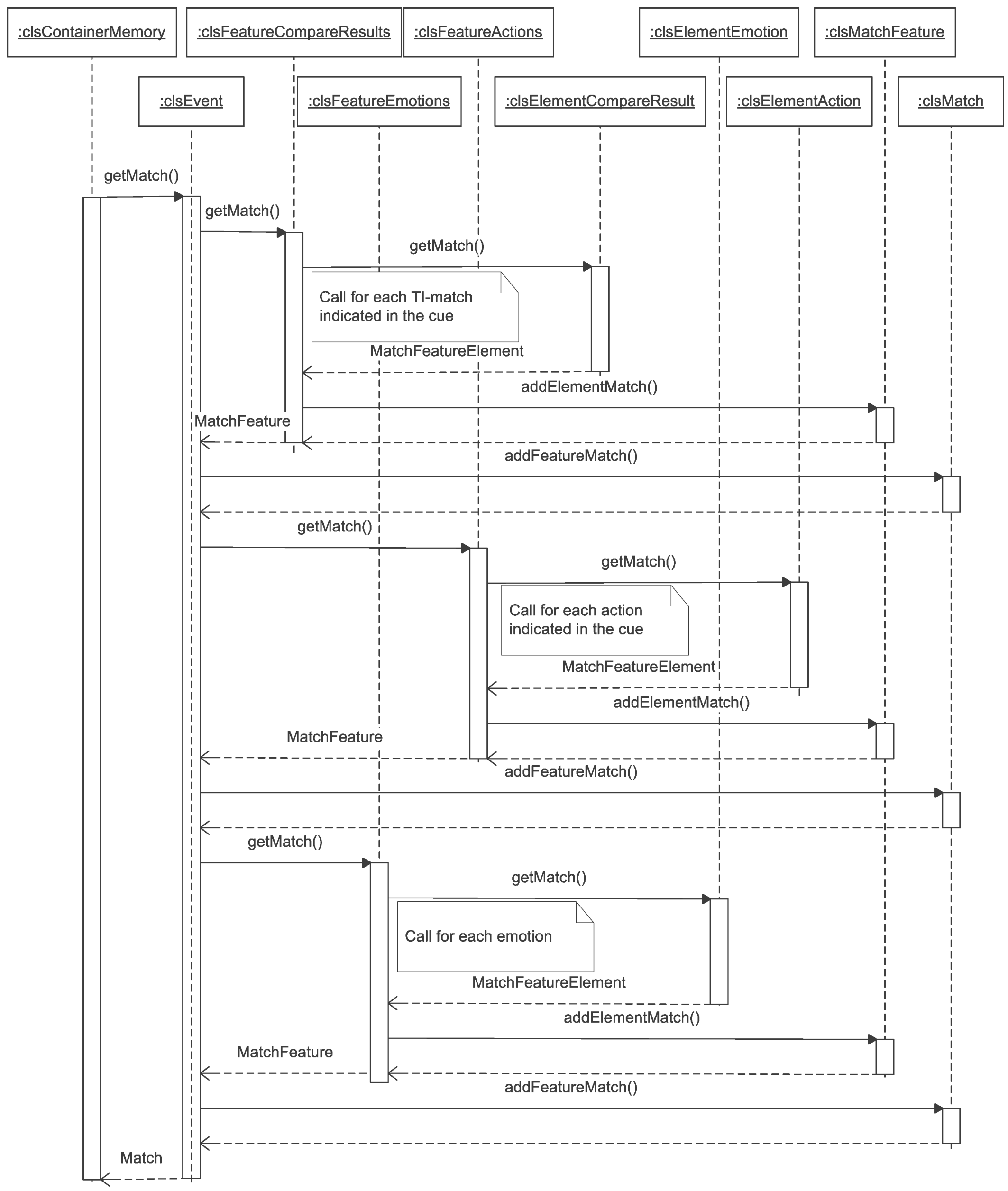

Figure 4.16: Sequence diagram of the matching procedure of a spontaneous retrieval with the salient features TI-matches and actions indicated in the retrieval cue

thus not indicated in the retrieval cue. The emotional state is always checked for spontaneous retrieval. This realizes the mood state-dependency (see section 3.4.3). Anything experienced in a specific emotional state (mood) will be better remembered, if the mood is reinstated. For deliberate retrieval, the emotional state is not checked implicitly. The individual emotions with their level to be searched for have to be indicated explicitly in the cue.

After determining the similarity relation between the cue and the stored event by a match, the retrieval result for the event is determined by this match and the activation level of the event. Each event that is traversed by the retrieval process gets a retrieval result. If the retrieval result exceeds a certain threshold, it will be written into the retrieval results container (figure 4.15). This container deposits the retrieval results to a certain retrieval cue in an ascending order (the best result first). The agent can later access these retrieval results and thus recall the corresponding events via the Recall API (see section 4.4).

4.4 Recall Unit

After determining the retrieval results to a certain retrieval cue, it is decided, which events and episodes are recalled by the agent. Therefore, the recall unit provides an API to access and recall the results of the retrieval procedure. The class diagram of the recall unit is illustrated in figure 4.17.

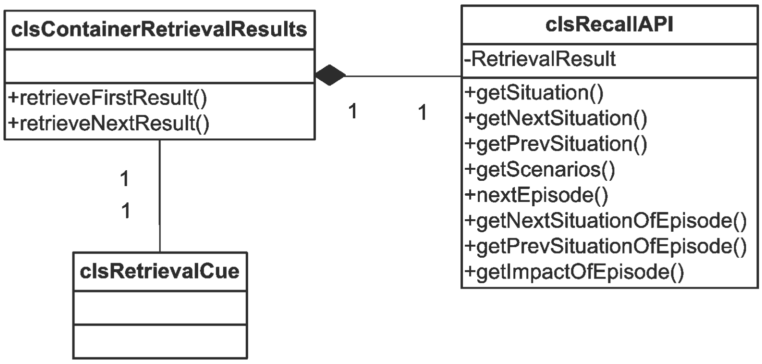

Figure 4.17: Class diagram of the recall unit

The container with the retrieval results (clsContainerRetrievalResults) to a certain retrieval cue (clsRetrievalCue) is the result of the retrieval procedure executed by the memory unit (section 4.3.3). These retrieval results are further accessible by the agent via the Recall API (clsRecallAPI) to recall its past experiences. In the first part of this section, recalling events with the Recall API on the basis of the retrieval results container is described. In the second part it is explained, how the operations provided by the returned container and the incorporated Recall API are applied to reconstruct episodes.

4.4.1 Recalling Events

If a retrieval (either spontaneous or deliberate) is completed, the container with the retrieval results will be returned by the memory unit. While the retrieval mechanism has measured the similarity of a stored event to a certain retrieval cue and combined with its activation level to a retrieval result, the agent will now decide, which events will be recalled via the Recall API. All features about an event are accessible. The returned data structures correspond to the initially delivered structures when adding data to the EM module via the Store API (section 4.1.2).

Container Retrieval Results

The retrieval results container (clsContainerRetrievalResults) (figure 4.17) consists of all retrieval results to a certain cue in an ascending order[3]—one for each event that has been

[3]The retrieval result with the highest result first.

traversed. If a scenario template is indicated in the cue, only the best result for each episode representing the specific scenario will be registered in the container (figure 4.3.3). Otherwise, if no scenario template is indicated, all events in the memory are traversed and each event gets a retrieval result. The container is returned to the agent when performing a retrieval. For spontaneous retrieval, it is returned by the Store API, for deliberate retrieval it is returned by the Retrieve API. To access and recall the events with the best result determined by the retrieval mechanism, the container incorporates a Recall API, which points on a particular retrieval result (figure 4.18). The respective retrieval result to which the Recall API points to can be selected by the following operations of the retrieval results container:

`retrieveFirstResult`— Sets the pointer within the Recall API on the first (the best) retrieval result. The data of the corresponding event is accessible by the Recall API which is returned.

`retrieveNextResult`— Sets the pointer within the Recall API to the next best retrieval result. If the retrieval was cued with a scenario template, the next retrieval result would be the best result of the next episode that is a realization of the indicated scenario template.

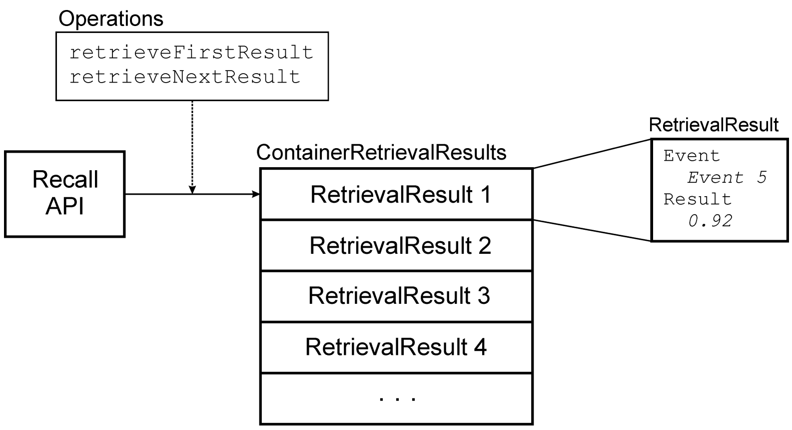

Figure 4.18: Selection of the retrieval result and the corresponding event to make it accessible for recall via the Recall API

Recall API

The Recall API (clsRecallAPI) (figure 4.17) provides the interface to access the event data of the results of the retrieval process stored in the container retrieval results. All data stored about an event is accessible (the entire event has been recorded). The EM module specific data structure (section 4.1.3), which has been defined to execute the basic operations on the individual elements like `triggerEncoding` or `getMatch` (section 4.1.3), have to be reconverted to the structures used by the implementation of the psychoanalytical model. Thus, the returned data structure corresponds to the structure that initially has been delivered to the EM module via the Store API (see section 4.1.2). A container with drives, emotions, TI-matches, and actions, which represent the entire situational information about the retrieved event, is returned (figure 4.19). If the event is part of one or more episodes, these will further be accessible for recall. For example, if the agent stands above an energy source, and the

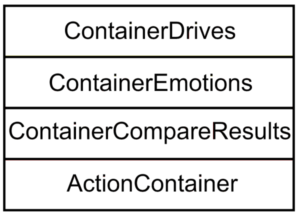

Figure 4.19: Structure of the recalled situational information of an event

retrieved event is part of an episode of the scenario *search and consume energy source*, the semantic knowledge represented by the scenario template will also be accessible for recall.

The recalled data refers to the respective event of the retrieval result to which the Recall API currently points to (figure 4.18). The operations of the Recall API that provide the agent to recall its past experiences are divided into operations that access the situational information about the retrieved event and operations that access the data about the episodes the retrieved event is related to. The latter are only available if the retrieved event belongs to one or more episodes. They are further divided into operations accessing the context, the content, and the impact of the episode.

Event Operations — As events do not have a context or an impact like an episode, only the situational information of the event is accessible (`getSituation`). Furthermore, the data about the succeeding respectively the previous event in the memory container is accessible by the temporal operations `getNextSituation` and `getPrevSituation`. The first operation can be seen as a synonym for the temporal question "what happened next?", the latter operation as a synonym for "what happened before?" The activation of the respective event that is recalled will be boosted. The episodes, the event belongs to are further strengthened.

Episode Operations — Episodes consist of a context, a content, and an impact (section 4.1.3). If the retrieved event is related to one or more episodes, the respective information of the episode is accessible via operations corresponding to the respective category of the episode.

Context Operations — A retrieved event may be related to various episodes which occurred concurrently. The context of these episodes (the respective scenario template) is accessible via the operation `getScenarios`. This refers to the question "in which context happened the event?". To select a particular episode for further recollection, the operation `nextEpisode` has to be called with the respective scenario template indicated in the parameter of the operation. The internal handled episode is set to the next episode of the indicated scenario template. The content and the impact of this episode are further accessible.

Content Operations — The content of an episode consists of the series of events in which the episode happened. After selecting an episode (`nextEpisode`), the situational data about the first, the last, the succeeding, or the previous event of the internal handled episode can be accessed (`get{First,Last,Next,Prev}SituationOfEpisode`). These operations are relevant for the reconstruction of an episode (section 4.4.2). The corresponding episode that is recalled is further strengthened.

Impact Operations — The emotional impact of a selected episode can be recalled with the operation `getImpactOfEpisode`. The emotional state of the agent at the end of the episode is returned which can be referred to the question "what emotional impact had the episode to the agent"? This operation can be used to evaluate the previously experienced episodes to select the most favorable for further episode reconstruction.

4.4.2 Episode Reconstruction

As depicted in section 4.4.1, the Recall API is the interface to the agent to recall its past experiences. It provides various methods to step back into the agents history. How these functions are applied to reconstruct episodes is considered in this section.

After the retrieval procedure has been accomplished, the container with the retrieval results is returned to the agent. By calling the operation `retrieveFirstResult` of the retrieval results container, the Recall API that points on the first retrieval result (figure 4.18) is returned to recall the respective event. Imagine an event related to three different episodes, that has achieved the best retrieval result, and is thus accessible via the Recall API (figure 4.20).

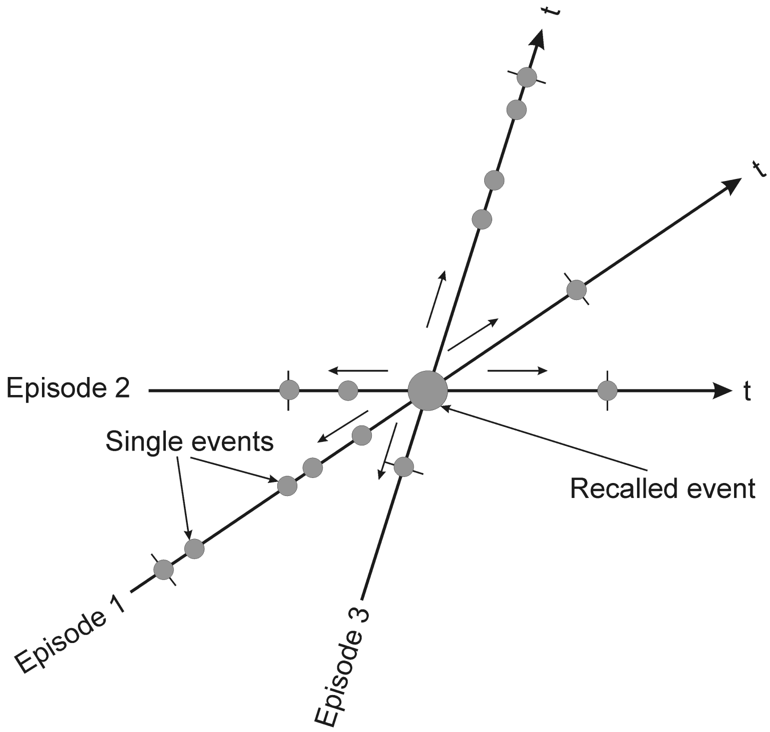

Figure 4.20: Recalling an event that is part of three concurrently happening episodes. The arrows indicate that the respective episodes can be traversed forward or backward.

Thus, these three episodes happened concurrently at the particular moment when the retrieved event was recorded. For example, episode 1 represents an episode of the scenario template *search and consume energy source (ES)*, episode 2 is a realization of the scenario template *help to crack ES* and episode 3 represents an episode of the scenario template *fight* *against enemy*. The agent was searching for an energy source to consume it, while it found a big ES and someone helped to crack it at the same time as it fought against an enemy. The beginning and the end of the individual episodes may be different, but at the time of experiencing the particular event, all three episodes were concurrently active.

By calling the context operations of an episode of the Recall API, the scenario templates of these three episodes are returned. Thus, it is known, that the recalled event occurred while the agent concurrently was searching for an energy source, another agent helped to crack it and it fought against an enemy. If it is desired to reconstruct the particular episode 2 (realization of the scenario template *help to crack ES*), the operation `nextEpisode` with the corresponding scenario template as parameter has to be called to set the internally handled episode pointer on episode 2. The emotional impact of the episode can be recalled by the impact operations of the Recall API for evaluation. To recall the particular events of episode 2 (indicated as the circles on the time line of episode 2), the content operations for an episode have to be called. The corresponding situational data is returned. The episode may be reconstructed and the desired final state may be reached again by re-executing the respective actions that were executed in the past episode.

If a retrieval is initiating with a scenario template indicated in the cue, only the best result of each episode representing this scenario will be stored in the container with the retrieval results (figure 4.13). The typical usage of such a retrieval is to remember a specific event that occurred within a certain context. For example, it is searched for a situation, where the agent was very angry and no energy source was visible (this composes the features of the retrieval cue) while searching for an energy source (represents the scenario template additionally indicated in the cue). For each episode that exists in the episode sub-container for this scenario the best retrieval result is determined (figure 4.13). If calling `retrieveFirstResult` of the retrieval results container, the Recall API that points to the first retrieval result will be returned. The same procedure as depicted above to reconstruct the episode can be executed. By further calling `retrieveNextResult`, the Recall API is set to the second best result, which is the best result to the retrieval cue of another episode of this scenario (e.g. the retrieval result for event 16 in figure 4.13). Again the episode within which this event (event 16) occurred (episode 6 in figure 4.13) can be reconstructed. Calling another time `retrieveNextResult`, the event referring to the best result of the next episode can be recalled (e.g. event 2 in figure 4.13). Thus, all episodes that occurred in the context indicated in the retrieval cue by a scenario template may be recalled. The respective experienced episode, that fits best to the agent's currently desired goal may further be reconstructed by executing the operations supported by the Recall API described above.

Chapter 5

Evaluation

In order to evaluate the proposed episodic memory (EM) model, the simulation environment "Bubble Family Game" (BFG) (section 2.2.2), which has been developed for evaluating the psychoanalytical model, serves as test environment. It consists of various agents (the "bubbles") which navigate in the virtual environment in order to keep their internal variable (e.g. energy) in a balanced range by finding and consuming energy sources (ES). Some bubbles are equipped with the psychoanalytical model, others act rule based. The bubbles that act on the basis of the psychoanalytical model further incorporate the EM module proposed in this work. It deposits and maintains the bubble's individual experiences and appropriately returns them. The task of the evaluation is to determine, if the evoked memories are suitable to the bubble in the respective situations. It is elucidated, whether the remembered experiences support the bubble in future decision making. The objectives of the evaluation are outlined in the first section of this chapter. The issues by which the EM module is evaluated follows a description of the simulation setup in the second section. In the third section of this chapter, the results to the stated objectives are presented and discussed.

5.1 Objectives

The objective of the evaluation of the proposed EM model is to assess the suitability of the EM model against the following issues:

- Can previously experienced episodes be remembered when retrieving with an appropriate cue?
- Are the retrieved episodes relevant to the current situation for spontaneous retrieval?
- Will anticipated future situations really occur?

The first issue deals with the rudimentary capability of the EM module of returning appropriate episodes to a certain retrieval cue. It is tested, whether previously experienced episodes will be remembered when retrieving with an appropriate cue. This refers to a deliberate retrieval, where some particular features are selected for the retrieval cue which should evoke a particular expected episode that is a realization of a certain scenario template. For example, when asking the EM module the question "remember an episode where you have been above an energy source, another bubble was visible and you called for help to crack food"—in this case the action *call for help* and the template image indicating that the bubble is above an ES and another bubble is visible are selected as retrieval cue. An episode of the the scenario template *cooperation for food* (see section 5.2) is expected to be retrieved.

The second question refers to the retrieval accuracy. It is tested, whether the retrieved episodes are relevant to the currently perceived situations for spontaneous retrieval. This measure can also be seen as the effectiveness of the cue construction mechanism for spontaneous retrieval. As depicted in section 3.4.2, the retrieval cue for spontaneous retrieval is formed by the salient features of the currently encoded event. The retrieval results of the event as cue will be confronted with the retrieval results of the entire features of the situation as cue. The latter represent the "accurate" results and thus it can be determined, whether the retrieval results to the salient features of the encoded event as cue are relevant to the current situation. To measure the accuracy of a retrieval it is defined that a retrieval is correct, if the retrieved episode of the spontaneous retrieval (the salient features of the event form the cue) is the same as the retrieved episode with the entire current situation as cue. The accuracy of a retrieval is thus defined as the quotient of the number of correctly retrieved episodes and the number of totally retrieved episodes (equation 5.1).

$$\text{accuracy} = \frac{\#\text{correct retrieved episodes}}{\#\text{total retrieved episodes}} \tag{5.1}$$

The third dimension for the evaluation of the EM module deals with the anticipative capability. When retrieving an event that is related to an episode, the agent is able to anticipate and predict future situation on the basis of the past experiences gained by the recalled episode. The individual situations (the events) of the episode can be recalled (see section 4.4), which are further expected to occur another time. The anticipative capability is defined as the quotient of the number of re-occurring anticipated situations of retrieved episodes and the total number of situations of retrieved episodes, that were expected to re-occur (equation 5.2).

$$\text{anticipative capability} = \frac{\#\text{anticipated situations}}{\#\text{total expected situations}} \tag{5.2}$$

This capability provided by the EM module is very essential, because it allows the agent to foresee possible impacts of certain behaviors. On the basis of this capability, the agent might act to avoid possible bad outcomes or to set behaviors to reach a particular desired goal.

5.2 Simulation Setup

As described in section 2.2.2, the simulation environment BFG consists of various autonomous robots called "bubbles", which navigate in the simulator in order to find and consume energy and may influence other objects. Objects within the bounded environment are obstacles, energy sources (small and big ones) and other bubbles. Bubbles loose energy, when they bump against obstacles or other bubbles. The total amount of energy within the simulation environment is limited—when a bubble consumes an ES, the energy level of the bubble increases and the level of the ES decreases. The setup of the BFG has been chosen to consist of two small ES (the orange squares), one big energy source (the red square), three obstacles (the gray squares), and two teams of bubbles (figure 5.1). The energy sources provide the energy to the bubbles. When an ES is consumed, the amount of energy will be regenerated by a certain rate. Small ES only contain a few amount of energy, but are consumable by one single bubble. Big ES contain more energy, but a second bubble is needed for cooperation to crack and to consume it. The energy sources are randomly distributed for each simulation run. One bubble team consists of two psycho bubbles (the magenta circles) and the other team consists of two non-psycho bubbles (the blue circles). Non-psycho bubbles act rule based, while psycho bubbles incorporate the psychoanalytical model with the EM module for decision making. The configuration of the psycho bubbles is further depicted in more detail.

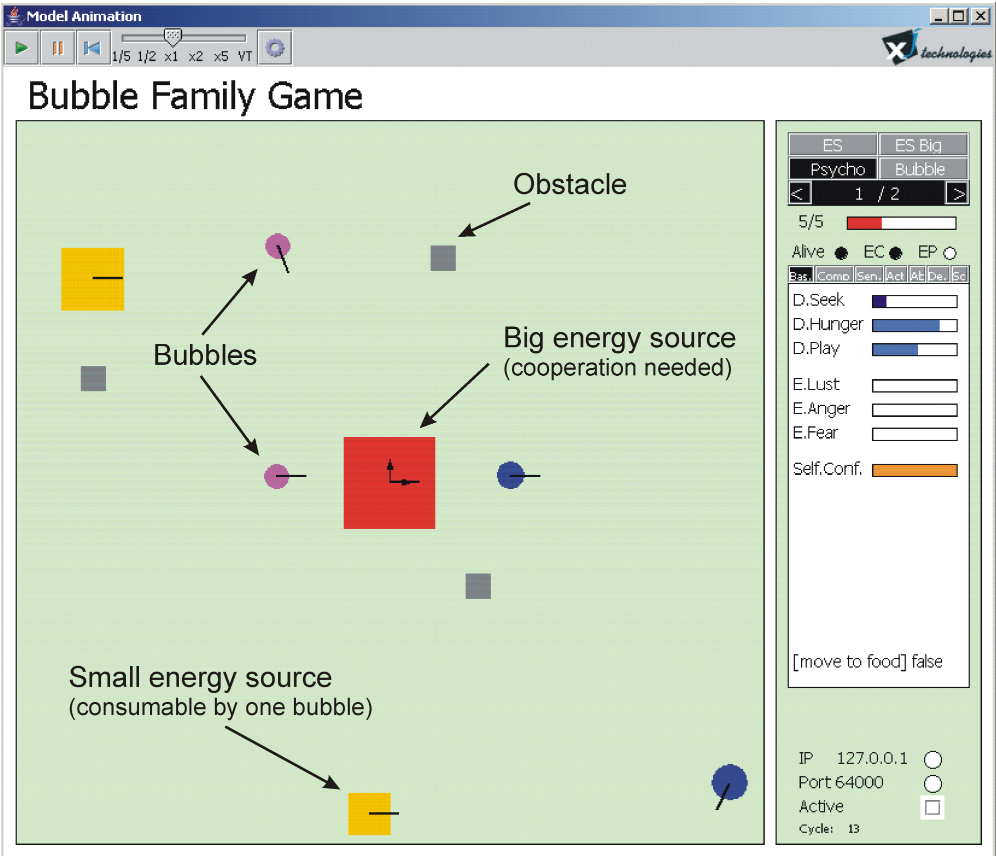

Figure 5.1: The simulation environment. Autonomous agents (bubbles) navigate in order to find and consume energy sources.

Bubble Configuration

Each psycho-bubble is configured with a predefined set of drives, emotions, actions, template images (TI), and scenario templates. In this setup it is the same for each psycho-bubble. The drives are hunger, seek, and play. Direct behavior may be triggered based on drives if they exceed a certain threshold. The emotions each bubble is equipped with are anger, fear, and lust. The externally and internally perceived situation is evaluated by the emotions. The actions represent the possible behaviors the bubble can perform. Only one action is carried out at a particular moment in time. The actions that can be executed by a psycho bubble are:

Promenade — The bubble navigates in the environment without any special purpose. This is the standard behavior which is executed when nothing special happens (no scenario is in progress).

Move to food — The bubble searches for an ES and moves to it.

Call for help — The bubble calls its team mates for help to crack a big ES.

Eat food — The bubble consumes the ES if it is consumable.

The set of template images represent the bubbles implicit knowledge base about the objects in the world. They represent a kind of snapshots of what is perceivable and each TI gets a match when perceiving a new situation. The predefined template images in the simulation setup for each psycho-bubble are:

TI 101: *hungry, not above ES*— The bubble is hungry and is currently not located above an ES.

TI 102: *above ES, bubble visible*— The bubble stands above an ES (either a big or a small one) and can see another bubble of the same team within a close range.

TI 103: *above ES and not consumable*— The bubble stands above and ES which is not consumable (it stands alone above a big ES)

TI 104: *above ES and consumable*— The bubble stands above a consumable ES (may either be a small ES or a big if another bubble helped to crack it).

TI 105: *saturated and lust*— The bubble is saturated (hunger level is low) and has a high level of the emotion lust.

TI 120: *receive call for help, not above ES*— The bubble receives a call for help to crack a big ES from another team mate. Further it is currently not located above an ES.

TI 70: *bubbles visible, want to attack*— The bubble can see another bubble within a close range and it wants to attack it (the levels of the emotions anger is high and fear is low).

TI 71: *bump against bubble*— The bubble bumps against another bubble (it is "fighting") while its anger is high and its fear is low.

TI 72: *bubble close, lost energy*— Another bubble is visible within a close range and the internal energy level of the bubble is low (drive hunger is high). After bumping against an object, a bubble "looses" energy.

The scenario templates are the set of known procedures. They represent the only kind of semantic memory the bubbles possess. On the basis of theses scenario templates, episodes are recorded in the form of series of events. Scenario templates consist of a sequence of states and transitions. Transitions are triggered by template images. If the specified TI-match exceeds a certain threshold, the scenario gets in the succeeding state. For each state, a certain action

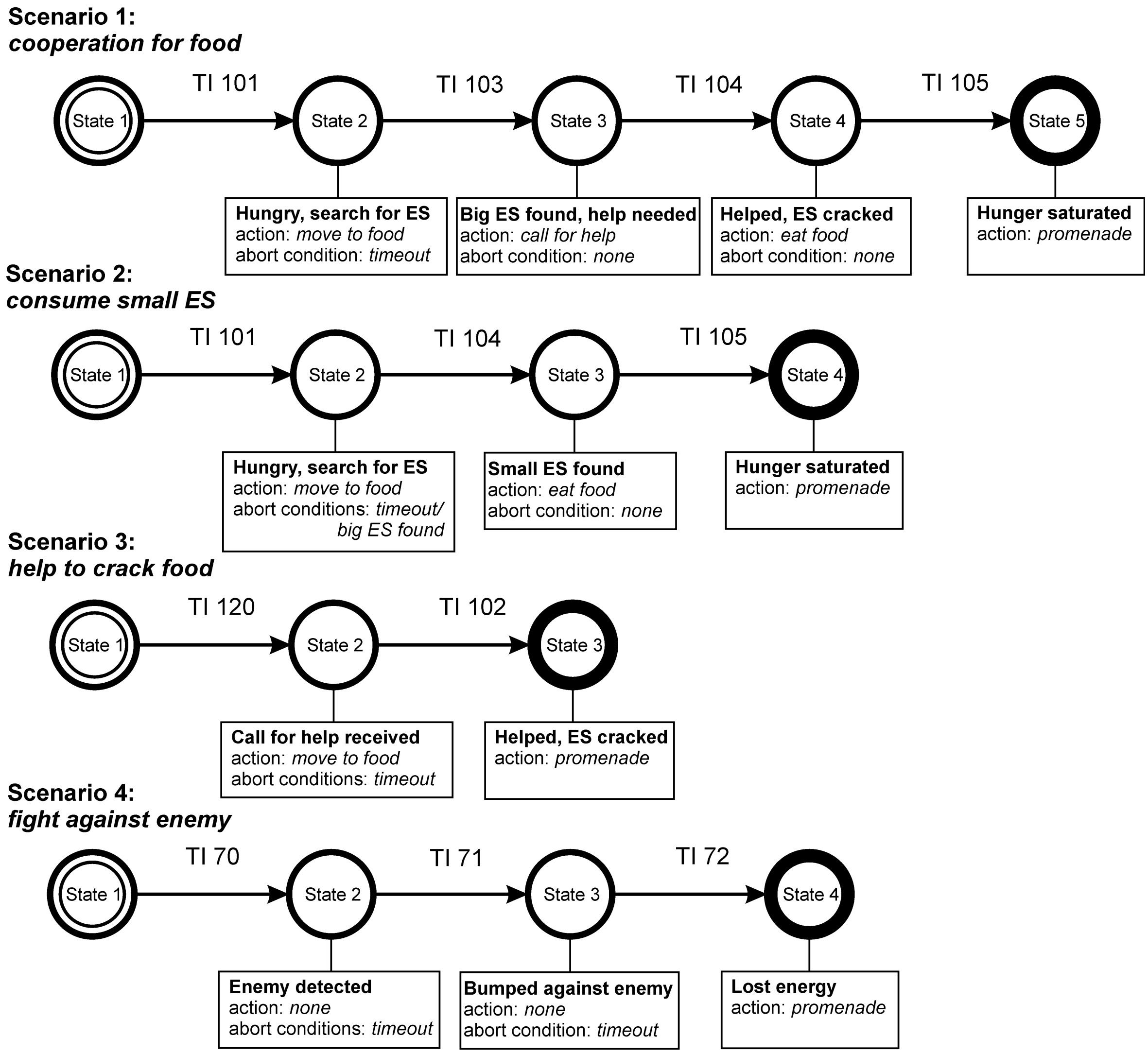

Figure 5.2: Defined scenario templates in the simulation setup for the evaluation of the EM module

may be indicated that is executed when the scenario is in this particular state. Thus, the behavior of the bubble is mainly coupled to the execution of scenarios. If no scenario is in progress, the standard behavior (*promenade*) is executed. A scenario is aborted, when a certain abort condition—e.g. a timeout condition—is fulfilled.

As illustrated in figure 5.2, there are four scenario templates defined in the simulation setup for the evaluation of the EM module:

Cooperation for food — This scenario represents the successful cooperation of two bubbles of the same team which help together to crack a big ES and consume it. It is initiated (the first transition triggers), when the internal energy level of the bubble is low and the bubble gets hungry. It searches for an ES, finds a big ES and calls the team mates for help to crack the ES. Another bubble of the same team helps to crack the ES (by executing the scenario *help to crack food*), and the ES can be consumed. The scenario ends when the bubble is saturated.

Consume small ES — This scenario represents the procedure of satisfying the hunger without the help of a team mate by consuming a small ES. Similar to the scenario *cooperation for food*, it is initiated when the bubble gets hungry. It searches for food, and if a small ES is found, it is consumed. If a big ES is found, this scenario is aborted (the scenario *cooperation for food* is executed).

Help to crack food — If another bubble of the same team needs help to crack a big ES, the bubble moves to the ES to help its team mate. Thus, this scenario is executed by a bubble, when the scenario *cooperation for food* is in progress of a team mate and it calls for help (state 3).

Fight against enemy — This scenario represents the procedure of fighting against another bubble when it is in"attack mode" (the level of the emotion anger is high and of the emotion fear is low). It is initiated when another bubble is visible in a very narrow range, and ends when it is bumped against this bubble which leads to a loss of energy.

5.3 Results

In this section, the results of the evaluation of the EM module to the three objectives issued in section 5.1 are discussed. The data is obtained from the psycho bubbles in various simulation runs. To evaluate the suitability of the EM module for the first issue—whether previously experienced episodes can be remembered when retrieving with an appropriate cue—deliberate retrievals are evoked at particular moments which are randomly selected. The scenario template that has been chosen to be considered for this evaluation is *cooperation for food*. It is examined, whether episodes that are realizations of this scenario template can be remembered with an appropriate retrieval cue. Thus, the retrieval cue used for the deliberate retrievals is constructed of the action *call for help* and the template image *above ES, bubble visible* (TI-id 102). The respective current emotional state of the bubble is additionally used as feature of the retrieval cue to highlight the mood state-dependency (section 2.1.2). The deliberate retrieval performed with this retrieval cue can be seen as forming a question like: "remember a situation standing above an ES and a bubble is visible, calling for help to crack the ES and being in a similar emotional mood to the current one". As this situation is very usual to appear in an episode of the scenario *cooperation for food*, it is expected that the EM module returns a stored event that is part of an episode of this scenario template.

Five simulation runs are observed, where three deliberate retrievals are made in each simulation run. The results are illustrated in table 5.1. As the moments of time when a deliberate retrieval is initiated are occurring randomly, there are different numbers of stored events (third column) and episodes in the EM module of the psycho bubble that is considered. The number of stored episodes to the corresponding scenario templates ("Sc") are indicated by the column "Stored episodes". The id of the retrieved event is further noted in the column "Retrieved event". If this event is part of one or more episodes, the id's of the corresponding scenario templates are indicated in the column "Retrieved scenarios". As the scenario template considered in this part of the evaluation is the scenario *cooperation for food*, the retrieved event should be part of an episode that is a realization of this scenario (id 1) (see figure 5.2). The last column shows whether the retrieved event is part of an episode of scenario 1 (OK) or not (NOK). For example, consider record 3 of simulation run 2. The bubble has yet experienced seven episodes of scenario 1 (*cooperation for food*), two episodes of scenario

Sim. run	Delib. retr.	Stored events	Stored episodes Sc1	Sc2	Sc3	Sc4	Retrieved event (id)	Retrieved scenarios	Result
1	1	82	1	1	1	0	70	1	OK
	2	154	2	1	2	0	70	1	OK
	3	323	5	1	3	0	153	1,3	OK
2	1	215	3	2	3	0	5	1	OK
	2	424	7	2	4	0	418	-	NOK
	3	614	7	2	9	2	360	1,3	OK
3	1	144	2	1	1	0	138	-	NOK
	2	198	4	1	1	0	159	1	OK
	3	240	6	1	1	0	159	1	OK
4	1	52	1	0	0	4	16	1,4	OK
	2	131	2	0	1	4	15	1,4	OK
	3	149	4	0	1	6	15	1,4	OK
5	1	193	1	1	1	0	1	1	OK
	2	290	2	1	2	0	194	1	OK
	3	379	4	1	3	3	320	1,3	OK

Table 5.1: Evaluation results whether experienced episodes can be remembered with an appropriate cue. Five simulation runs with three deliberate retrievals for each simulation run are illustrated.

2 (*consume small ES*), nine episodes of scenario 3 (*help to crack food*) and 2 episodes of scenario 4 (*fight against enemy*). The retrieved event (id 360[1]) is both part of an episode of scenario 1 (*cooperation for food*) and scenario 3 (*help to crack food*). Thus, in this particular event, the bubble was concurrently helping another bubble to crack the ES while it needed energy by itself and cooperated with a bubble to crack it. Record 3 of simulation run 4 for example shows, that the bubble retrieves an event (id 15) that is both part of a *cooperation for food* episode (scenario id 1) and a *fight against enemy* episode (scenario id 4). Thus, in this event the bubble had to fight against an enemy while cooperating with another bubble to crack a big ES. It may further be frightened when cooperating again with other team mates because of this experience and may thus avoid cooperations and prefer searching for small ES that it can consume by itself. As illustrated in table 5.1, in simulation run 4 the bubble remembers in all three deliberate retrievals this respective event that reminds it on this happening when it had to fight against an enemy to cooperate with another bubble to crack a big ES.

The EM module performed successful in thirteen of fifteen cases. In both cases where the deliberate retrieval failed (the second deliberate retrieval of simulation run 2 and the first retrieval of simulation run 3), respectively a single event is retrieved that has a very high matching of features and that occurred very recently. Thus the activation has not yet decayed that strong and the event gains a higher retrieval result than other stored events that belong to an episode of scenario 1. A further effect of the recency of the retrieved event can be deduced from the very high match of the retrieved single events. The emotional state has not yet changed since the occurrence of the event which leads to a very high match. In all the other deliberate retrievals, events that belong to an episode of scenario 1 are retrieved, even if some particular single events have a higher matching of features. The activation of these

[1]The id of an event results from an internal numbering within the EM module and has no further meaning.

high matching events is still very low (single events decay faster than those that are related to episodes), which makes them hardly accessible. Another problem that may arise is an effect that can be denoted by "over-strengthening" of events. Events that belong to more episodes that occurred concurrently decay very slow. If they are further recalled various times, their activation gets very high and remains that high. Thus, such events will be very frequently recalled when retrieving with an appropriate cue. This occurs especially in simulation run 4, where the events 15 and 16 have a very high activation and thus the events of other—even more recently—episodes are not retrieved to the particular cue.

To evaluate the EM module for the second issue depicted in section 5.1—whether the retrieved episodes are relevant to the current situation for spontaneous retrieval—the different recollections of the two psycho bubbles are observed where they are confronted to similar situations. The individual recollections of the bubbles heavily depend on what they have experienced so far. Based on these stored subjective episodes, the recollections of spontaneous retrieval of some predefined situations are examined whether they are relevant to the current situation. This measure is denoted in section 5.1 as retrieval accuracy. It is defined that an episode is correctly retrieved for spontaneous retrieval—and thus relevant to the current situation—if it is also retrieved with the entire situation used as cue. The accuracy of the retrieval is further defined as the quotient of the correctly retrieved episodes by the total number of spontaneous retrievals (equation 5.1). Therefore, two representative situations are declared, for which the spontaneous retrievals should be examined. Each time, a psycho bubble is confronted to such a situation, the retrieved episode[2] of the spontaneous retrieval is compared to the result of a deliberate retrieval using all features of the current situation as cue. If the latter retrieval will retrieve the same episode, the spontaneous retrieval is defined to be correct, otherwise it is not. The two representative situations for this evaluation are:

Situation "eat" — The bubble is above a consumable ES, it consumes it and another bubble is visible. Thus the action *eat food* must be set and the TI *above ES, bubble visible* (TI-id 102) must have a high match.

Situation "fight" — The bubble promenades, is in "attack mode", detects another bubble and bumps against it. Thus the action *promenade* must be set, and the TI's *bubbles visible, want to attack* and *bump against bubble* must have a high match.

The spontaneous retrievals of the two psycho bubbles where they are in such situations are illustrated in table 5.2 in a chronological order. Twenty spontaneous retrievals are recorded. They are listed chronologically (column "Retr.") according to the moments in time of occurrence. In the second column it is indicated which retrieval belongs to which bubble. The type of situation ("eat" or "fight") is shown in the third column. For example, the retrieval record 14 of bubble 1 occurs while the bubble faces a situation like "fight" as described above. The columns "Sto. ev." and "Stored episodes" indicate the number of already occurred and therefore stored events and episodes at the time of the spontaneous retrieval. The two succeeding columns show the retrieved episodes of the spontaneous retrieval (column "Spontaneous") and the deliberate retrieval (column "Deliberate") with the entire situation as cue for refer-

[2] The result of a retrieval is not directly an episode, it is an event. With the notation retrieved episodes are meant the episodes, the retrieved event is related to. As depicted in section 4.4, the episodes that are related to a retrieved event can further be recalled and reconstructed.

Re-tr.	Bub-ble	Situ-ation	Sto. ev.	Stored episodes				Spontaneous		Deliberate		Res-ult
				Sc1	Sc2	Sc3	Sc4	Ep-id	Sc-id	Ep-id	Sc-id	
1	1	eat	77	1	0	0	0	0	1	-	-	NOK
2	1	eat	113	2	0	0	0	4	1	4	1	OK
3	2	eat	84	2	0	0	4	4	1	4	1	OK
4	2	eat	104	2	1	1	4	13	3	4	1	NOK
5	2	eat	151	2	1	3	4	12,13	2,3	12,13	2,3	OK
6	1	eat	166	2	0	2	0	0	1	4	1	NOK
7	1	eat	177	2	1	2	0	4	1	4	1	OK
8	**2**	**eat**	**165**	**3**	**1**	**3**	**4**	**12,13**	**2,3**	**12,13**	**2,3**	**OK**
9	2	eat	195	3	2	3	13	12,13	2,3	12,13	2,3	OK
10	1	eat	224	2	1	2	3	4	1	4	1	OK
11	1	eat	235	3	1	2	3	4	1	4	1	OK
12	1	eat	240	4	1	2	3	11,12	4,4	4	1	NOK
13	2	eat	209	4	2	4	13	13	3	13	3	OK
14	**1**	**fight**	**256**	**4**	**1**	**2**	**3**	**12**	**4**	**12**	**4**	**OK**
15	1	fight	262	4	1	2	5	12	4	12	4	OK
16	1	fight	272	4	1	2	6	12	4	12	4	OK
17	2	eat	221	5	2	4	13	12,13	2,3	12,13	2,3	OK
18	1	eat	284	4	1	2	7	4	1	4	1	OK
19	1	fight	296	5	1	2	7	12	4	12	4	OK
20	2	fight	239	5	3	5	13	13	3	13	3	OK

Table 5.2: Evaluation results to measure the retrieval accuracy of spontaneous retrieval

ence. The sub-column "Ep-id" shows the id[3] of the retrieved episode. The corresponding id of the scenario template is indicated in the sub-column "Sc-id". For example, the retrieval recored 8 indicates that the episodes 12 and 13 are retrieved (the retrieved event is related to these two episodes). Episode 12 is a realization of scenario 2 (*consume small ES*) and episode 13 a realization of scenario 3 (*help to crack food*). Thus, in this recalled event, the bubble was concurrently consuming a small ES while it was called to help another bubble to crack a big ES. A retrieval result is relevant to the current situation (OK), if the same episode is retrieved with the deliberate retrieval, otherwise it is not (NOK).

An important fact that can be deduced from the results of table 5.2 is that the bubbles remember different episodes in comparable situations. The recollections depend on the experiences the individual bubbles have made so far. The episodes are specific for each bubble. They occur at different moments in time, have different sequences of events, and different contents. For example, episode 4 of bubble 1 (record 2) is completely different to episode 4 of bubble 2 (record 3) although both are episodes of scenario 1 (*cooperation for food*) (they happened in the same context). The id's of the episodes of each bubble are specific for each bubble and have nothing to do in common. A further noticeable fact is that bubble 1 nearly always remembers episode 4 (*cooperation for food*) when it is confronted to a situation like "eat". The activation level of this event is very high that it is more easily retrieved than other episodes. In contrast, bubble 2 often remembers episode 12 and 13. Even in a "fight"

[3]Similar as the id of an event, the id of an episode represents a number that is unique and assigned by an internal indexing. In contrast, the id's of the scenario templates represent the corresponding scenarios defined in figure 5.2.

situation (record 20), an episode of scenario template 3 (*help to crack food*) is remembered. According to the definition of the retrieval accuracy, in 16 of 20 spontaneous retrievals the same episode is remembered with the salient features of the encoded event as cue and the entire situation as cue. Following the definition of the retrieval accuracy in equation 5.1, this results in an accuracy value of 0.8. This value expresses that in eight of ten cases the result of the spontaneous retrieval with the salient features of the encoded event as cue is the same like the result with the entire situation as cue. In record 1, a single event is retrieved by the deliberate retrieval which has a high activation. Even if the action *eat food* does not match in this single event, the match of the other features and the activation are that high that the retrieval result is the best. The other three mismatches result in different values of matching of features due to the differently formed retrieval cues. This does not conclude that these four spontaneous retrievals are "incorrect". It does only mean that not exactly the event respectively the episode is remembered that has the very best match to all features of the current situation. Imagine, humans do not always remember exactly the experience that "best" matches to the current situations. This result reflects, that the cue construction mechanism for spontaneous retrieval ("reminding") is pretty accurate. Eighty percent of the spontaneous retrievals with the salient features as cue retrieve the same episode as a retrieval with the entire, detailed data with all features of a situation as cue. Even though the concept of selecting salient features from an event has not merely been introduced to lower the computational burden on the retrieval procedure[4], the result shows, that the salient features of the currently encoded event as cue for spontaneous retrieval serve as effective reminders for similar episodes.

The evaluation of the EM module for the third issue stated in section 5.1—whether future situations can be anticipated—is done by observing the spontaneous retrievals of the two psycho bubbles. If a bubble retrieves an episode, it is able to recall the constituent events of the episode and thus is able to foresee possible future situations. Whether these predicted situations really occur is further measured by the anticipative capability of the EM module. Therefore, it is checked, whether the last situation (the end event) of a retrieved episode will re-occur within a certain timespan after the retrieval of the episode. This observation is made, if an episode of the same scenario template as the retrieved episode is currently in progress (see section 4.2.2). This means that the agent reminds an episode that occurred in the same context as the current happening. Thus it is expected, that a similar situation like the last situation of the retrieved episode will occur again at the end of the episode that is currently in progress. For example, if an episode of the scenario *cooperation for food* is in progress and an episode of the same scenario is retrieved, it is expected that the bubble will be saturated and happy in the end of the episode, because it was the same in the retrieved episode. To determine whether the anticipated situation re-occurs, all subsequent situations after the retrieval of the corresponding episode within a timespan of 200 simulation cycles are checked for matching of features to the expected situation (the end event of the episode). If the match M_e of the end event of the episode is greater than 85 percent ($M_e > 0.85$)[5] , the situation is defined to be re-occurred. Thirty spontaneous retrievals have been recorded from the two psycho bubbles in a simulation run which are illustrated in table 5.3.

For each spontaneous retrieval, the number of currently stored events and episodes are shown.

[4]The main purpose of the salient feature selection (section 3.2.2) is to establish the compatibility relation between the retrieval cue and the engram according to Tulving's encoding specificity principle (section 2.1.2).

[5]The value of 0.85 for the match that has to be gained for a situation that it is defined to be correctly anticipated has been chosen empirically.

Retr-ieval	Bub-ble	Stored events	Stored episodes				Spontaneous r.		Anticipated situation	Occ-urred
			Sc1	Sc2	Sc3	Sc4	Ep-id	Sc-id		
1	1	20	1	0	0	0	0	1	after eat	OK
2	1	34	2	0	0	0	2	1	after eat	OK
3	1	52	2	1	0	1	6	4	after fight	OK
4	1	71	2	1	0	2	0	1	after eat	NOK
5	1	80	2	2	0	2	15	2	after eat	NOK
6	1	81	2	2	0	2	0	1	after eat	OK
7	1	102	3	2	0	4	15	2	after eat	OK
8	1	112	3	3	0	4	16	1	after eat	OK
9	1	150	4	3	0	4	15	2	after eat	OK
10	2	319	0	4	6	0	20	2	after eat	OK
11	1	309	4	4	1	4	15	2	after eat	OK
12	1	348	4	5	1	4	15	2	after eat	OK
13	2	378	1	5	6	0	25	1	after eat	OK
14	1	406	4	6	2	9	15	2	after eat	OK
15	2	395	2	5	6	0	20	2	after eat	OK
16	1	417	4	7	2	9	15	2	after eat	OK
17	1	429	4	8	2	10	15	2	after eat	NOK
18	2	420	2	6	6	2	35	4	after fight	OK
19	2	421	2	6	6	3	35	4	after fight	OK
20	1	436	4	8	2	13	16	1	after eat	OK
21	2	430	2	6	6	3	35	4	after fight	OK
22	2	441	2	6	6	4	37	4	after fight	OK
23	1	454	5	8	2	13	16	1	after eat	OK
24	1	476	6	8	2	13	15	2	after eat	NOK
25	1	477	6	8	2	13	16	1	after eat	OK
26	**1**	**491**	**7**	**8**	**2**	**13**	**16**	**1**	**after eat**	**OK**
27	2	640	2	6	9	4	20	2	after eat	OK
28	1	511	8	8	2	13	16	1	after eat	OK
29	2	677	2	7	10	4	38	4	after fight	NOK
30	2	681	2	7	10	4	20	2	after eat	OK

Table 5.3: Evaluation results to measure the anticipative capability of the EM module. Thirty spontaneous retrieval are recorded of the two psycho bubbles from one simulation run.

The retrieved episode and the id of the scenario template (its context) is indicated in the column "Spontaneous r.". The respective last situation (the end event) of the retrieved episode is then taken to be the anticipated situation. For example, record 26 shows that bubble 1 has stored 491 events at the moment of this spontaneous retrieval. It has yet experienced seven *cooperation for food* episodes, eight *consume small ES* episodes, two *help to crack food* episodes, and thirteen *fight against enemy* episodes. The retrieved episode is episode 16 which is a realization of scenario 1 (*cooperation for food*). The last situation of this episode is further used as the to-be-anticipated situation which is in this case of the type "after eat". The last column shows, if this situation really occurs (OK) or not (NOK) within the given timespan. Depending on the scenario in which context the retrieved episode happened, two different kinds of situations have emerged. These are denoted as "after eat" for situations that occur in the end of either a *cooperation for food* episode (scenario id 1) or a *consume small ES* episode (scenario id 2), and as "after fight" for situations in the end of an episode of a scenario *fight against enemy* (scenario id 4). The characteristics of these situations are:

Situation "after eat" — In a typical situation in the end of an episode where the bubble has consumed an ES, it is saturated (the level of drive hunger is low) and the level of the emotion lust is high. Further, the TI indicating that the bubble is above a consumable ES has a high match. In an episode of the scenario *cooperation for food* additionally the TI indicating that another bubble is visible matches. The action typically set in this situation is *promenade* (nothing special happens).

Situation "after fight" — The relevant features in a typical situation after a *fight against enemy* episode are that the bubble has lost some energy and is thus hungry. Other bubbles are visible and the level of the emotion anger is typically high and the level of the emotion fear low. The action typically set is *move to food* (a scenario recognition process of scenarios 1 or 2 may be initiated due to the high level of drive hunger).

The respective situations that are anticipated of course are different for each individual retrieved episode, but these patterns of situations are observed to occur frequently in the end of the corresponding episodes.

There are some features observable from the gained data. First of all, it can be traced back in the data that if an episode of a certain scenario template is retrieved and the corresponding anticipated situation later on occurs, a new episode of this scenario will be encoded. Thus, if an episode of the scenario 1 (*cooperation for food*) is retrieved and the anticipated situation occurs, the number of stored episodes of this scenario is incremented by one (e.g. from record 8 to record 9). An exception shows retrieval record 2, where bubble 1 retrieves an episode of scenario 1 (*cooperation for food*), the anticipated situation occurs, but an episode of the scenario *consume small ES* (scenario 2) is encoded. This happens, because the bubble has not experienced a *consume small ES* episode before and thus retrieves a *cooperation for food* episode of the (similar) scenario 1. Furthermore it is observable that episodes of the scenario 4 (*fight against enemy*) are rarely retrieved.

Another interesting feature observable in the data is that each time one bubble experiences an episode of scenario 1 (*cooperation for food*), the other bubble experiences an episode of scenario 3 (*help to crack food*). While bubble 1 helps bubble 2 only twice[6] to crack the

[6] The fact, that bubble 1 experiences twice a *help to crack food* episode is observable in the last recorded record of bubble 1 (record 28).

big ES (bubble 1 has two stored *cooperation for food* episodes and bubble 2 has two stored *help to crack food* episodes), bubble 2 helps bubble 1 eight times[7]. The fact, that bubble 2 experiences ten *help to crack food* episodes (record 30) instead of eight comes along with two erroneous recognized episodes. While bubble 2 wants to help bubble 1 to crack a big ES, it finds a small ES where another bubble is close and visible and thus it thinks that it has cracked a big ES. Consider record 13, where bubble 2 experiences a *cooperation for food* episode. The number of stored *cooperation for food* episodes (scenario 1) of bubble 2 increments by one (record 15), while the number of stored *help to crack food* episodes (scenario 3) of bubble 1 concurrently increments by one (from record 12 to record 14). This underlines the characteristic of the episodic memory that it is subjective. In contrast to the general knowledge of the semantic memory which is the same for each agent, each agent has its individual autobiographical memories that reflect its own history.

25 of 30 situations are anticipated correctly. Following the definition of the anticipative capability of the episodic memory (equation 5.2), this results in a value of 0.83 for the anticipative capability of the EM module. In five spontaneous retrievals the expected situation does not occur. In the retrieval record 4, a *cooperation for food* episode is retrieved, but the bubble finds a small ES and consumes it (a *consume small ES* episode is recorded). Thus the anticipated situation of the retrieved *cooperation for food* episode does not occur. In the spontaneous retrievals of records 5, 17, and 24 the same happens but the other way round. In these cases an episode of the scenario *consume small ES* is retrieved, but a *cooperation for food* episode occurs (it can be observed that in the respective succeeding retrievals a *cooperation for food* episode is retrieved which further occurs). In the retrieval record 29, bubble 2 retrieves an episode of the scenario *fight against enemy* which does further not occur and thus the anticipated situation does not happen. The bubble sees another bubble and wants to attack it, but they do not collide and the expected situation does not occur.

The results show, that the agents equipped with an EM module are able to anticipate possible future situations in a reliable way. In 83 percent of the spontaneous retrievals, where the agent reminds an episode that occurs in a similar context as the current happening, the respective anticipated situations of the previously experienced episode really occur. This ability supports the autonomous agents to better evaluate perceived situations and enhances their decision making. Foreseen situations with a bad impact[8] can be avoided by selecting an appropriate behavior, whereas desired impacts can be reached by reconstructing the reminded episode (section 4.4.2). The autonomous agents are able to foresee future situations and to predict their outcome which supports their decision making to reach certain desired goals. Beside these capabilities, the agents are able to deliberately reflect about their personal past which can be used for further learning (e.g. to learn from its own mistakes or from rewarding some excellent behaviors).

[7]Bubble 1 experiences eight *cooperation for food* episodes (record 28).

[8]The impact of an episode respectively a situation is evaluated in the EM module by emotions and is further accessible for recall (see section 4.4).

Chapter 6

Conclusion and Further Work

An episodic memory (EM) module has been designed and integrated into the psychoanalytical (PA) model of the "Artificial Recognition System" (ARS). On the basis of this EM module, past situations and episodes can be remembered including an emotional evaluation. Future situations can be anticipated and their impact can be predicted. This capability enables the system to better evaluate perceived situations and to improve decision making. Equipped with the ability of reflecting its personal past, a system can reconstruct whole episodes to reach a certain desired goal. Beside a successful integration of the EM module into the PA model of the ARS project, some further work has to be done to improve an intelligent system by using memories of past experiences. A main issue is to couple the EM system with a semantic memory (SM) system.

6.1 Conclusion

The long-term goal of the ARS project is to build an intelligent autonomous system that copes with unforeseen situations in a reliable and flexible way (section 2.2). The developed psychoanalytical model—a bionic approach for decision making—has therefore been extended by an episodic memory module that enhances the capability of the system to anticipate and predict future situations on the basis of past experiences. The episodic memory covers the whole history of an agent and is crucial for remembering its personal experiences. The approach to build a computational model of an EM proposed in this work is based on concepts of profound scientists from the areas of neuro-psychoanalysis and psychology. Some important aspects of episodic memory as a part of human long-term memory is depicted in section 2.1 under special consideration of its similarities and differences to semantic memory. The fundamental functionalities of episodic memory processing (encoding, storage, and retrieval) and the concept of consolidation are elucidated in more detail. Further, a conceptual framework for the study and understanding of EM is outlined in this section. The intention of this work is not to reproduce the complete structural architecture of a human episodic memory in a technical system, but to implement a computational model that coincides with the most important features of episodic memory research.

The architecture of the computational model inspired by these concepts of neuro-psychoanalysis and psychology is presented in chapter 3. An event based approach is proposed that captures experiences of an agent when something significant happens in a situation. An event is defined to be a prototypical experience. Episodes are sequences of events happening in a certain context, which is characterized by the semantic information of predefined scenario templates. The emotional impact of episodes is further attached and accessible for later recall. The fundamental stages of episodic memory processing are divided into encoding, storage, and retrieval. Encoding determines when an experience has to be recorded as an event and selects the relevant, highly activated part of a situation as the salient features of an event. Storage covers the processes that maintain the stored events and episodes. Events are stored chronologically ordered into the memory, whereas episodes are stored according to the context in which they happen and can be nested or overlapped in time. The proposed forgetting mechanism describes the crucial maintenance functions to improve the accessibility of essential events and episodes. Therefore, an activation level is attached to the stored events that depends on various factors like the time delay since the encoding of the event, the emotional involvement of the agent to the event, or the frequency of subsequent recollections. Retrieval of stored events and episodes may either be deliberate or spontaneous. Spontaneous recollections characterize the memories, that pop up without the explicit effort of the system and are termed as "reminding". For deliberate retrieval, a certain retrieval cue has to be constructed. A similarity relation—which is declared to be matching of features—between the cue and the stored events determines which events and episodes will be evoked. What is recalled not only depends on the matching of features, but also on the activation level. The respective event and the potential related episodes are recalled and may be used for further situation evaluation. Possible possible future situations can be anticipated or whole episodes can be reconstructed.

The technical implementation of the architectural memory model is described in chapter 4. The interface of the EM module to the implementation of the psychoanalytical model of the ARS project is detailed and the EM module is described by the functional units alarm system, memory unit, and recall unit. The task of the alarm system is to monitor the incoming data stream and to detect, when an event or episode has to be encoded. If further determines, when spontaneous retrieval has to be evoked. The memory unit deposits the events and episodes, and handles its dynamics over time. The recall unit provides the interface to access the data of the retrieved experiences.

To evaluate the EM module (chapter 5), the "Bubble Family Game"—a simulation environment designed for testing and evaluating the PA model of the ARS project—serves as a test environment. The stored events and episodes of autonomous agents equipped with the PA model are analyzed in various simulation runs. The results (section 5.3) show, that previously experienced episodes can be successfully remembered when retrieving with an appropriate cue. Moreover, a high retrieval accuracy—the relevance of a spontaneously retrieved episode to the current situation—is measured. Furthermore, the anticipative capability of the EM module—whether anticipated situations really occur—proofs to be very reliable.

The obtained results demonstrate, that the EM module supports the situation evaluation capability of an intelligent, autonomous system like the PA model of the ARS project. Past episodes happening in a similar emotional state can be recognized, possible future situations can be anticipated, and their impact can be analyzed. Equipped with this capability, the system is supported to better evaluate perceived situations and to enhance decision making. Whole episodes can be reconstructed to reach again a desired goal. Alternatively, not desired impacts can be avoided by selecting appropriate behaviors.

6.2 Further Work

Some further work is proposed to enhance the efficient support of an autonomous system like the ARS system with an episodic memory.

Interaction with Semantic Memory

In the current implementation of the PA model of the ARS project, semantic memories are only provided by predefined scenario templates. This kind of SM represents scripts of known procedures like how to make coffee. To better interpret the happenings within events and to equip the system with the ability to ask semantic questions, it is proposed, that the EM module interacts with an SM module. Semantic relations between the features of events may be deduced and further attached to the stored events and episodes. For example, if a specific emotion like *lust* frequently appears simultaneously with a particular action like *eat food*, a tight relationship between these features will be established. Furthermore, a whole semantic network of relations between features can be generated that serves as a knowledge base derived from experienced events.

Episodic Learning

Semantic knowledge can either be predefined by scenario templates or relationships among features and feature elements, but may also be learned from repeatedly occurring series of events and episodes. New scenario templates can be generated from template images, that regularly appear in a certain sequence of events. These scenario templates are stored to the semantic memory and serve as contextual information of episodes. The ability of the system of deliberately reflecting its personal past can further be used for learning. Behaviors can be improved by learning from own mistakes or from rewarding excellent behaviors.

Algorithms

The algorithms of the EM module in the current implementation are not fully optimized. Especially the retrieval algorithm may be potentially enhanced by pre-selecting events with features that are relevant to a given retrieval cue. Furthermore, the retrieval strategy may be improved. The retrieval of an event or episode is initiated by the formation of a retrieval cue, that only consists of information for the retrieval of one event. It is proposed, that a retrieval is extended by sequences of events that serve as retrieval cue. This can lead to a more accurate retrieval of events, that better matches to the current happenings.

Extensibility

The EM module can be extended by further features like complex emotions or desires. Complex emotions are more socially emotions like shame or compassion and are composed by a combination of basic emotions (section 2.2). The EM module is designed to be easily extended, thus features of an event can be simply added. The respective weighting of the features have to be adapted appropriately.

Appendix A

Acronyms

API	Application Programming Interface
ARS	Artificial Recognition System
BFG	Bubble Family Game
EM	Episodic memory
ES	Energy source
GAPS	General Abstract Processing System
ID	Identification
ISAC	Intelligent Soft Arm Control
LTM	Long-term memory
MOP	Memory organization packet
PA	Psychoanalysis
PC	Perception
PM	Procedural memory
SM	Semantic memory
Soar	States, Operators and Reasoning
STM	Short-term memory
TI	Template Image

Bibliography

[Bad97] Alan Baddeley. *Human Memory: Theory and Practice*. Psychology Press, 1997. ISBN 0-86377-431-8.

[Dam03] Antonio Damasio. *Looking for Spinoza: Joy, Sorrow, and the Feeling Brain.* Harvest Books, 2003. ISBN 0-15-100557-5.

[DG05] Will Dodd and Ridelto Gutierrez. The role of episodic memory and emotion in a cognitive robot. *IEEE International Workshop on Robots and Human Interactive Communication*, pages 692–697, 2005.

[DLP+06] Tobias Deutsch, Roland Lang, Gerhard Pratl, Elisabeth Brainin, and Samy Teicher. Applying psychoanalytic and neuro-scientific models to automation. *The 2nd IET International Conference on Intelligent Environments*, pages 111–118, 2006.

[Dod05] Will Dodd. The design of procedural, semantic, and episodic memory systems for a cognitive robot. Master's thesis, Faculty of Graduate School of Vanderbilt University, 2005.

[FG97] Stan Franklin and Art Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. *ECAI'96: Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages*, pages 21–35, 1997.

[Fue03] Clara Tamarit Fuertes. *Automation System Perception – First Step towards Perceptive Awareness*. PhD thesis, Faculty of Electrical Engineering and Information Technology, Vienna University of Technology, 2003.

[HDN05] Wan Ching Ho, Kerstin Dautenhahn, and Chrystopher L. Nehaniv. Autobiographic agents in dynamic virtual environments - performance comparison for different memory control architectures. *IEEE Congress on Evolutionary Computation*, pages 573–580, 2005.

[HDNB04] Wan Ching Ho, Kerstin Dautenhahn, Chrystopher L. Nehaniv, and Rene Te Boekhorst. Sharing memories: An experimental investigation with multiple autonomous autobiographic agents. *IAS-8, 8th Conference on Intelligent Autonomous Systems*, pages 361–370, 2004.

[LLR05] Pat Langley, John E. Laird, and Seth Rogers. Cognitive architectures: Research issues and challenges. Technical report, Stanford University, Stanford, CA 94305, 2005. [Online `https://ctools.umich.edu/access/content/user/laird/`

Cognitive%20Architecture%20Papers/Cognitive%20Architecture.pdf; accessed 20-October-2006].

[LLR06] John E. Laird, Fill F. Lehman, and Paul Rosenbloom. A gentle introduction to soar, an architecture for human cognition: 2006 update. Technical report, University of Michigan, 2006. [Online http://ai.eecs.umich.edu/soar/sitemaker/docs/misc/GentleIntroduction-2006.pdf; accessed 5-November-2006].

[MON05] Stefan Mahlknecht, Roland Oberhammer, and Gregor Novak. A real-time image recognition system for tiny autonomous mobile robots. *Real-Time Systems, Volume 29*, pages 247–261, 2005.

[NL02] Andrew Nuxoll and John E. Laird. Enhancing intelligent agents with episodic memory. Technical report, University of Michigan, Artificial Intelligence Laboratory, 2002. [Online http://www.eecs.umich.edu/~anuxoll/Andrew%20Nuxoll_Prelim_Episodic%20Memory02.html; accessed 15-October-2006].

[NL04] Andrew Nuxoll and John E. Laird. A cognitive model of episodic memory integrated with a general cognitive architecture. *International Conference on Cognitive Modeling*, pages 220–225, 2004.

[NLJ04] Andrew Nuxoll, John E. Laird, and Michael R. James. Comprehensive working memory activation in soar. *Proceedings of the 6th International Conference on Cognitive Modeling*, pages 226–230, 2004.

[Pra06] Gerhard Pratl. *Processing and Symbolization of Ambient Sensor Data*. PhD thesis, Faculty of Electrical Engineering and Information Technology, Vienna University of Technology, 2006.

[RLVF06] Charlotte Roesner, Brigitte Lorenz, Katharina Vock, and Georg Fodor. Emotional behavior arbitration for automation and robotic systems. *Proceedings of 2006 IEEE International Conference of Industrial Informatics, Singapore*, pages 423–428, 2006.

[Rus03] Gerhard Russ. *Situation-dependent Behavior in Building Automation*. PhD thesis, Faculty of Electrical Engineering and Information Technology, Vienna University of Technology, 2003.

[Sch96] Daniel L. Schacter. *Searching for Memory: The Brain, the Mind, and the Past*. Basic Books, 1996. ISBN 0-465-02502-1.

[ST94] Daniel L. Schacter and Endel Tulving. *Memory Systems*. The MIT Press, 1994. ISBN 0262193507.

[ST02] Mark Solms and Oliver Turnbull. *The Brain and the Inner World*. Karnac/Other Press, Cathy Miller Foreign Rights Agency, London, England, 2002. ISBN 1-59051-017-8.

[Tec05] Dan Tecuci. A generic episodic memory module. Technical report, University of Texas at Austin, 2005. [Online www.cs.utexas.edu/users/tecuci/proposal/proposal.ps; accessed 23-October-2006].

[Tul83] Endel Tulving. *Elements of Episodic Memory*. Oxford: Clarendon Press, 1983. ISBN 0198521022.