2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Cognitive Memory Systems In Consciousness And Memory Model

Zhongzhi Shi, Xiaofeng Wang, and Xi Yang

Abstract—**Memory is a fundamental component in human brain and plays very important roles for all mental processes. The analysis of memory systems through cognitive architectures can be performed at the computational, or functional level, on the basis of empirical data. In this paper we discuss memory systems in the extended Consciousness and Memory Model (CAM) The knowledge representations used in CAM for working memory, semantic memory, episodic memory and procedural memory are introduced. It will be explained how, in CAM, all of these knowledge types are represented in dynamic decription logic (DDL), a formal logic with the capability for description and reasoning regarding dynamic application domains characterized by actions.**

I. Introduction

A cognitive architecture is the essential structure and processes of cognition in the form of a broadly-scoped, domain-generic computational model, used for a broad, multiple-level, multiple-domain analysis of behavior [1]. We can benefit a great deal from utilizing cognitive architectures for functional comprehension. Cognitive modeling is a key scientific issue of intelligence science to use information theory and technology to model the human mind.

Memory is a fundamental component in human brain and plays very important roles for all mental processes. The analysis of memory systems through cognitive architectures can be performed at the computational, or functional level, on the basis of empirical data. In this paper we discuss memory systems in the extended Consciousness and Memory Model (CAM).

First of all we introduce prior approaches to model the human mind. One of the most impressive attempts to model the human mind is Anderson's ACT-R[2, 3], which has become a famous cognitive system and has been continuously developed for decades. Researchers have used ACT-R to model variety of psychological phenomena including memory, attention, reasoning, problem solving, and language processing.

John Laird, Allen Newell and Paul Rosenbloom developed a symbolic cognitive architecture named Soar in early 1980s[4, 5]. Soar is both a vision of what cognition is, and an implementation of this vision in the form of a software architecture. Soar has been used to develop many sophisticated agents, such as TAC-Air-Soar[6] which models fighter pilots in military training exercises.

PRODIGY is another cognitive system that was extensively developed from the middle 1980s to the late 1990s[7]. PRODIGY incorporates two kinds of rule based knowledge. One is domain rules that encode the conditions under which actions have certain effects. Another one is control rules that specify the conditions under which the architecture should select, reject, or prefer an operator, set of operator bindings, problem states or goals.

Ron Sun proposed CLARION model which specifies a four-way partitioning of the major memory systems (modules) is as follows: implicit procedural (action-centered) memory, explicit procedural (action-centered) memory, implicit declarative (non-action-centered) memory, and explicit declarative (non-action-centered) memory [1]. Declarative (non-action-centered) memory may be further partitioned into semantic memory and episodic memory. There is also a working memory in CLARION, which is for storing information temporarily for the purpose of facilitating action decision making, while activations of rules and chunks may be viewed as short-term memory. CLARION is based on the separation of the two dichotomies, as there are both implicit and explicit procedural memory, and both implicit and explicit declarative memory.

Pat Langley proposed a cognitive architecture named ICARUS[8]. ICARUS is neither the oldest nor most developed architecture, compared with some continually developing cognitive systems like ACT-R and Soar. However, ICARUS distinguishes itself via its concern with physical agents that operate in an external environment and focuses on the organizing, using and acquiring hierarchical knowledge structures. ICARUS also has some deficiencies, such as the lack of a sophisticated computational model to represent, manage and inference the knowledge in the cognitive system.

In 1998, Franklin and his colleagues developed a cognitive model called Intelligent Distribution Agent (IDA) which is implemented as a software agent[9].The IDA model implements and fleshes out Global Workspace theory [10,11], which suggests that conscious events involving widespread distribution of focal information need to recruit neuronal resources for problem solving. The neuronal resources enhanced IDA is renamed as Learning Intelligent Distribution Agent (LIDA). The LIDA architecture adds three fundamental, continuously active and learning mechanisms to the existing IDA system that underlie much of human learning: a)perceptual learning, the learning of new objects, categories, relations, etc., b) episodic learning of events, the

Zhongzhi Shi is with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (phone: 86-10-82610254; fax: 86-10-82610254; e-mail: shizz@ ics.ict.ac.cn).

Xiaofeng Wang is with Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: wangxiaofeng@ict.ac.cn).

Xi Yang is with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: yangx@ ics.ict.ac.cn).

This work was supported by the National Program on Key Basic Research Project (973) (No. 2013CB329502), National Natural Science Foundation of China (No. 61035003, 61202212, 61072085, 61202211), National High-tech R&D Program of China (863 Program) (No.2012AA011003), National Science and Technology Support Program(2012BA107B02).

what, where, and when, c) procedural learning, the learning of new actions and action sequences with which to accomplish new tasks[12]. The LIDA cognitive cycle consists of nine steps, that is, perception, percept, local associations, competition for consciousness, conscious broadcast, recruitment of resources, setting goal context hierarchy, action selection and action taken [13].

Ben Goertzel and his colleagues have recently proposed a cognitive architecture titled OpenCog Prime for robot and virtual embodied cognition [14]. They define a set of interacting components designed to give rise to human-equivalent artificial general intelligence (AGI) as an emergent phenomenon of the whole system.

Rosenbloom has proposed factor graphs as an uniform implementation level for cognitive architecture, and demonstrated that they can yield a state-of-the-art production match algorithm [15]. He and his colleagues have developed a Sigma cognitive architecture by generalized language of conditionals, which compiles down to factor graphs for processing via the summary product algorithm [16]

Chris Eliasmith and his team of Waterloo neuroscientists have developed an artificial brain called Spaun (Semantic Pointer Architecture Unified Network) which contains 2.5 million virtual neurons. It can perform at least eight different tasks, from simple ones like copying an image, to more complex ones similar to those found on IQ tests, such as finding the next number in a series [17].

Haikonen proposes a consciousness model based on a distributed representational economy, instantiated in a neural network architecture[26]. Symbolic processing ability is also enabled to handle the critical task of generating inner language. The consciousness in the proposed model is not a single functional model. It emerges when the cognitive processes of agent work in unison and focus their attention upon the same entity as perceived by the various sensory modalities[27].

Chella and colleagues work on an operational mobile robot which serves as a museum tour guide[28]. Three levels of representation are used, which are sub-conceptual area concerning low-level processing of sensor data, conceptual area organizing the lower level sensory data into conceptual categories and structured information and linguistic area performing first-order logic based calculus.

In our prior work, we have outlined a mind model named Consciousness And Memory Model (CAM) [18]. Here we present the Extended CAM, incorporating a variety of novel features. Compared with previous cognitive models, CAM has several important distinct characteristics, such as dynamic description logic(DDL)-based knowledge representation; unique and sophisticated computational models for perception, cognition, and motivation-model based consciousness.

The remainder of this paper is organized as follows. Section 2 gives the extended architecture of CAM. Section 3 will introduce dynamic description logic (DDL) which is used for knowledge representation in CAM. Sections 4 will discuss knowledge representation in working memory, semantic memory, episodic memory and procedural memory. Finally, the conclusions of this paper are drawn and future works are pointed out.

II. Extended Architecture of CAM

The extended architecture of CAM is illustrated in Figure 1 and organized into ten modules as follows:

1) Visual module: The visual module is the part of the central nervous system which gives organisms the ability to process visual detail, as well as enabling the formation of several non-image photo response functions. It detects and interprets information from visible light to build a representation of the surrounding environment. The visual system carries out a number of complex tasks, including the reception of light and the formation of monocular representations; the buildup of a binocular perception from a pair of two dimensional projections; the identification and categorization of visual objects; assessing distances to and between objects; and guiding body movements in relation to visual objects. From lateral geniculate nucleus (LGN) neuron send their signals to the primary visual cortex V1. About 90% of the outputs from the retina project to the LGN and then onward to V1. In the ventral pathway, many signals from V1 travel to ventral extrastriate area V2, V3 and V4 and onward to many areas of the temporal lobe.

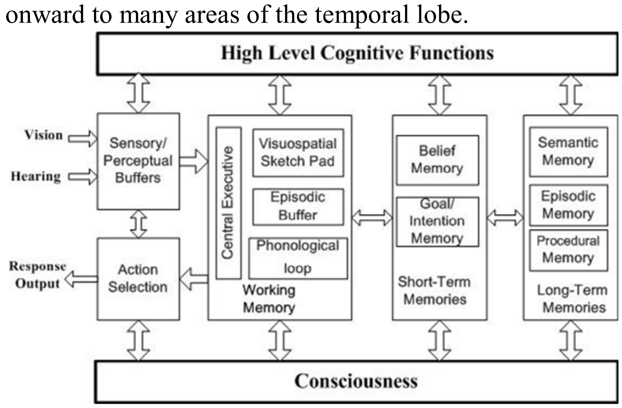

Fig. 1 Extended architecture of CAM

2) Aural module: The auditory module is comprised of many stages and pathways that range from ear, to the brainstem, to subcortical nuclei, and to cortex. The advent of neuroimaging techniques has provided a wealth of new data for understanding the cortical auditory system.

3) Sensory buffers: Each of the classical senses is believed to have a brief storage ability called a sensory buffer.

4) Working memory: It includes the central executive, visuospatial sketch pad, phonological loop and episodic buffer. The central executive is future directed and goal oriented in effective, flexible and adaptive. At the basic level the working memory is located in the prefrontal cortex.

5) Short-term memory: Stores the agent's beliefs, goals and intention contents which are change rapidly in response to environmental conditions and agent's agenda.

6) Long-term memory: Contains semantic, episodic and procedural knowledge which change gradually or not at all.

7) Action selection: It is the process of constructing a complex composite action from atomic actions to achieve a specific task. Action selection can be divided into two steps, first is atomic action selection, i.e., select related atomic action from action library. Then selected atomic actions are composed together using a planning strategy. One of action selection mechanism is based on a spiking basal ganglia model.

8) Response output: The motor hierarchy begins with general goals, influenced by emotional and motivational input from limbic regions. The primary cortical motor region directly generates muscle based control signals that realize a given internal movement command.

9) Consciousness: The primary focus is on global workspace theory, motivation model, attention, and the executive control system of the mind in CAM.

10) High level cognitive functions: It includes a class of high level cognitive functions which perform cognitive activities based on the basic cognitive functions supported by the memory and consciousness components of CAM.

III. Dynamic Description Logic

In CAM, knowledge representation is based on dynamic description logic (DDL) which is an extension of description logic[29]. It introduces the notion of action into the description logic system in order to support the representation and reasoning of dynamic knowledge[19].DDL represents and reasons dynamic knowledge based on the notion of action.

An atomic action in DDL is a tuple $\alpha \equiv \langle P, E \rangle$, where

(1) $\alpha \in N_A$ is the name of the atomic action;

(2) P is a set of formulas which specifies the precondition that should be satisfied when the action is executed;

(3) E is a set of formulas that specifies that effect of action execution;

Intuitively, an atomic action encodes the changes of the domain caused by the action. For example, the action of buying a movie ticket can be defined as:

$$\text{buyTicket}(a,t) \equiv (\{Person(a), Ticket(t), \neg own(a,t)\}, \{own(a,t)\}).$$

Beside the atomic action, the DDL allows to define the complex actions which are constructed based on atomic actions. The following are the rules to construct complex actions.

$$\pi, \pi' \rightarrow \alpha | \phi? | \pi \cup \pi' | \pi ; \pi' | \pi^*$$

where α is an atomic action and ϕ is a formula in DDL. The action in form of "ϕ?" is named as test action which tests whether ϕ is true. Action in form of "$\pi \cup \pi'$" is named as select action which represents a selection between π and π'. Action in form of "$\pi ; \pi'$" is named as sequence action which represents an action that sequentially executes action π and action π'. Action in form of "π^*" is named as iteration action. The iteration action represents an action that iteratively executes action π. The complex action can be used to represent program control flows. For example, the program fragment: " if ϕ then π else π'" can be rewrited into a complex action:

$$(\phi?;\pi) \cup ((\neg\phi)?;\pi').$$

Knowledge base in DDL can be divided into three components which are TBox, ABox and ActBox. The definitions of TBox and ABox are same as those in description logic. The ActBox is defined as a finite set of definitions on atom actions and complex actions.

IV. Memories

Memories are the core components of CAM. They consist of working memory, semantic memory, episodic memory and procedural memory. Different memory plays different role in the cognitive process. The following subsections will introduce the memory components in details.

Before delving into the details of memories components, we briefly introduce a CAM based cognitive process example, in order to illustrate the role, functions and the internal structure of the memory components. The example is about a CAM based cognitive process for detecting the video event based on the semantic trajectory[20], which is a high-level semantic description of the moving object's trajectory in the video. The process consists of three phases which are transforming low-level visual features to middle-level raw trajectory information, turning raw trajectory information to high-level semantic trajectory, and performing event reasoning with the assistance of semantic trajectory information and background knowledge to detect the event. Except Phonological loop component, all the memory components are involved in the event detection process. Figure 2 illustrates the general structure of CAM based event detection process.

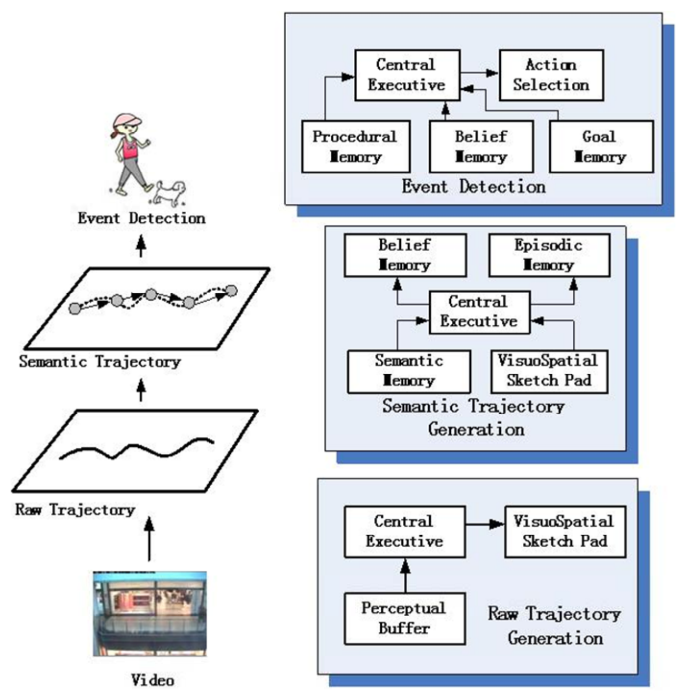

Fig. 2 General structure of CAM based event detection

As depicted in Figure 2, CAM based event detection process consists of three sub-process which are raw trajectory generation process, semantic trajectory generation process and event detection process. In raw trajectory generation process, the perceptual buffer load the video data from visual sensors, then the central executive component extracts the trajectory of interesting moving objects(here refers to human) from the video. The moving trajectory information is stored in the visuospatial sketch pad component. In semantic trajectory generation process, the central executive component generates the semantic trajectory based on the raw trajectory information stored in visuospatial sketch pad component. The semantic trajectory generation process is carried out under the guidance of the knowledge stored in semantic memory, such as, the knowledge of definitions of point of interests and the knowledge of definitions of key elements of semantic trajectory. The generated semantic trajectory information is stored in belief memory and episodic memory. The event detection process performs the event pattern matching to detect the events from the semantic trajectory. When a specific event is detected, a corresponding action is carried out. The procedural memory stores a set of actions to support the event based reaction in CAM. Moreover, the action taken to be executed should be consistent with the information stored in goal memory, which means the result of action execution should satisfy the intension in CAM and should have no contradiction with the goal.

A. Working Memory

The ability to mentally maintain information in an active and readily accessible state, while concurrently and selectively process new information is one of the greatest accomplishments of the human mind. Working memory provides temporary storage and manipulation for language comprehension, reasoning, problem solving, reading, planning, learning and abstraction.

As illustrated in Figure 1 and Figure 2, the working memory involves four subcomponents: central executive, visuospatial sketch pad, phonological loop and episodic buffer. The central executive is the core in working memory. It drives and coordinates other subcomponents in working memory to accomplish cognitive tasks. The visuospatial sketch pad holds the visual information about what the cognitive system had seen. The phonological loop deals with the sound or phonological information. The episodic buffer stores the linking information across domains to form integrated units of visual, spatial, and verbal information with time sequencing (or chronological ordering), such as the memory of a story or a movie scene. The episodic buffer is also assumed to have links to long-term memory and semantic meaning [21].

In the semantic event detection example, the central executive component includes a human classifier which recognizes human from video and a moving object tracker which can track the position of moving object in the video. For a video sequence loaded in the perceptual buffer, it first extracts the possible moving objects with some background modeling based method. Then the human classifier is applied to the possible moving objects to find out whether the object is a human. If the moving object is recognized as a human, then the object tracker is applied to the moving object. At last, the trajectory obtained by the tracker is stored in the visuospatial sketch pad. According to the example, we can see that the central executive component takes charge of the process of extracting raw trajectory from the video.

B. Semantic Memory

Semantic memory stores general facts which are represented as ontology. In philosophy, ontology is a theory about the nature of existence. In information science, ontology is a document or file that formally defines the relations among terms. The most typical kind of ontology for the semantic Web has a taxonomy and a set of inference rules.

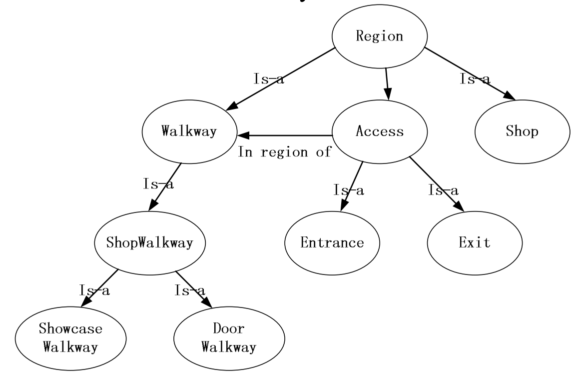

Fig. 3 Ontology fragment of interesting regions of a mall

In CAM, ontology specifies a conceptualization of a domain in terms of concepts, attributes, and relations in the domain. The concepts provide model entities of interest in the domain. They are typically organized into a taxonomy tree where each node represents a concept and each concept is a specialization of its parent. Each concept in a taxonomy is associated with a set of instances. By the taxonomy's definition, the instances of a concept are also instances of an ancestor concept. Each concept is associated with a set of attributes. In CAM, Logic language dynamic description logic (DDL) is used to define ontology.

For the example of CAM based event detection, the semantic memory stores the definitions of interesting regions of the scene which appears in the video and the elements of semantic trajectory that match with the interesting region of the scene. Figure 3 is the fragment of ontology which defines the interesting regions of a mall.

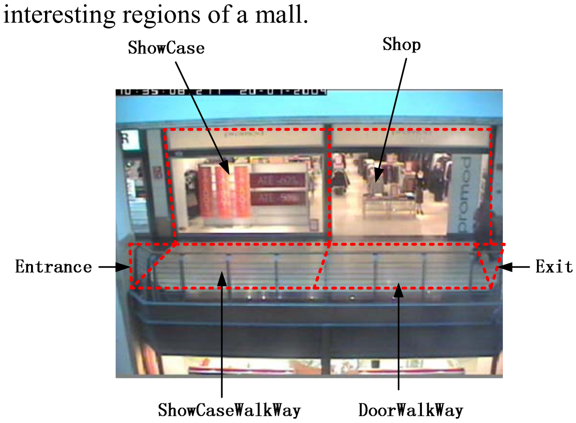

Fig. 4 Scene-concept map

For the example of CAM based event detection, the semantic memory stores the definitions of interesting regions of the scene which appears in the video and the elements of semantic trajectory that match with the interesting region of the scene. Figure 3 is the fragment of ontology which defines the interesting regions of a mall. In order to matching the defined concepts of interesting regions with the scene appearing in the video, a scene-concept map is constructed and stored in the visuospatial sketch pad. Figure 4 shows the scene-concept map.

With the region ontology and scene-concept map, the central executive component partitions the raw trajectory into trajectory segments according to the region. From each trajectory segment, the semantic stop points are generated according to the semantic trajectory elements defined in ontology. We will not go into the detail of semantic trajectory generation process, as here the semantic trajectory process is only an illustration to discuss the memory components in CAM. For those interesting readers, please refer to our paper[22]. Finally, the generated semantic trajectory information is stored in belief memory and episodic memory respectively, which means the generated semantic trajectory should be the description of current situation and also prepare to be the historical record in future.

C. Episodic Memory

Episodic memory is one part of long-term memory that involves the recollection of specific events, situations and experiences which are snapshots of working memory. Nuxoll and Laird demonstrated that an episodic memory can support an intelligent agent to own a multitude of cognitive capabilities [23].

In CAM the episode is an elementary unit that stores previous scene in episodic memory where an episode is divided into two levels: one is an abstract level in terms of logic, another is a primitive level shown in Figure 5. Among them, episode is represented in the form of logic symbol on the abstract level. The primitive level includes perception information correlated to abstract level of the described object. In order to represent and organize perception of the episode effectively, we adopt DDL to describe episode in abstract level and ontology in primitive level. Object data graph (ODG) is used to describe episode.

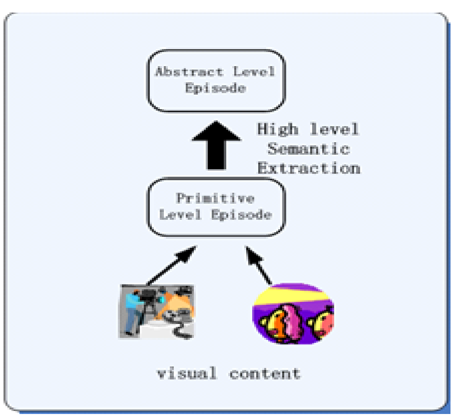

Fig. 5 Two levels of episode

Figure 6 depicts an ODG structure of film Waterloo Bridge where objects associate with other objects through URI in episode. Figure 6 shows us 3 objects: M2, W2, and film Waterloo Bridge. In addition, object W2 has worn a blue skirt. The film also associates with two main roles M1, W1 and among them W1has worn a white coat.

In Soar, the retrieval of episode is modeled as a case-based reasoning problem which finds solutions to problems according previous experience [5]. We follow this idea and build a case based system to retrieve the episode according to the cues. To simplify the system, we restrict the cue to be a transitional sequence like episode. Then, the retrieval of episode is modeled as problem of finding the episode that is most relevant to the cue. As the abstract level episode can represent the content of episode precisely, thus the matchmaking is only performed between cue and abstract level episode. In CAM, the transitional sequence is formally defined as a possible world sequence and whether the episode implies cue can be inferred by the DDL based tableau algorithms. A possible world sequence is a directed acyclic graph Seq=(W_p, E_p), where $w_i \in W_p$ represents a possible world, edge e_i=(w_i,w_j) represents an action α_l that is executed in w_i and lead the state transformed from w_i to possible world w_j.

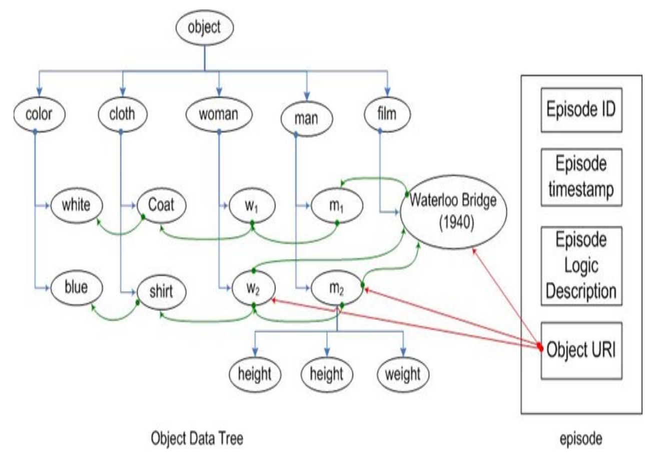

Fig. 6 Object data graph

From the DDL perspective, an episode e_p can imply a cue c if and only if $e_p \rightarrow c$ is a valid formula. Thus, we infer the implication relationship between episode and cue based on the process that can determine whether formula $e_p \rightarrow c$ is valid. The process that infers the implication between episode and cue is described in the algorithm 1.

Algorithm1 CueMatch(e,c)
Input: episode$e$,cue$c$
Output: whether $c \preceq_p e$ hold

if $length(e) < length(c)$ then
    return false;
end
n_e:=$first\_node(e)$;
n_c:=$first\_node(c)$;
if $MatchPossibleWorld(n_e; n_c)$ then
    α_e:= $Null$;
    α_c:=$action(n_c)$;
    if $\neg(Pre(\alpha_e) \rightarrow Pre(\alpha_c))$ *unsatisfiable according DDL tableau algorithm* then
        n_{tmp}:=n_e;
        while $next\_node(n_{tmp}) \neq Null$ do
            α_e:= (α_e; $action(n_{tmp})$) ;
            if $MatchAction(\alpha_e; \alpha_c)$ then
                Let sub_e be the sub sequence by removing $\alpha_e$from e ;

15 Let sub_c be the sub sequence by removing α_c from c ;
16 if *CueMatching*(sub_e; sub_c) then
17 return true;
18 end
19 end
20 n_{tmp}:=*next node*(n_{tmp}) ;
21 end
22 end
23 end
24 Remove n_e from e ;
25 return *CueMatching*(e; c) ;

Function MatchPossibleWorld(w_i,w_j)
Input: possible worlds w_i, w_j
output: whether $w_i \vDash w_j$ hold
1 f_w:=$Conj(w_i) \rightarrow Conj(w_j)$;
2 if $\neg f_w$ *is unsatisfiable according to DDL tableau algorithm* then
3 *return* true;
4 else
5 *return* false;
6 end

Function MatchAction(α_i,α_j)
input : action α_i, α_j
output: whether $\alpha_i \vDash \alpha_j$ hold
1 if α_i == *null or* α_j == *null* then
2 *return* false
3 end
4 f_{pre}:=$Conj(Pre(\alpha_i)) \rightarrow Conj(Pre(\alpha_j))$;
5 f_{eff}:=$Conj(Eff(\alpha_i)) \rightarrow Conj(Eff(\alpha_j))$;
6 if $\neg f_{pre}$ *and* $\neg f_{eff}$ *are unsatisfiable according to DDL Algorithm* then
7 *return* true ;
8 else
9 *return* false ;
10 end

In Algorithm 1, the *length*() function returns the length of a possible world sequence. The length of a possible world sequence is determined by the number of nodes contained in the sequence. Function *action*(n) returns the action that executed at possible world n in the possible world sequence. The *next_ node*(n) function returns the next node of n in the possible world sequence. The next node of n means the one that is reached by executing the action *action*(n) in the possible world sequence. The step 14 in Algorithm1is the procedure of making a composed action by connecting two action α_e and *action*(n_{tmp}) with sequential action constructor ";". In order to simplify the algorithm, we assume that the action (Null; α)==α. In algorithm 1, *MatchPossibleWorld*() and *MatchAction*(), there are some processes that check whether a formula's negative form is unsatisfiable according to the DDL tableau algorithm. This kind of process is used to ensure that the formula is valid which means the formula is always hold.

In the event detection example, the past semantic trajectory is stored as the episode. With the cue based episode retrieval algorithm, the semantic trajectories that match the certain pattern described as cue can be retrieved. The cue matching algorithm forms the core of the process to discover event from past trajectories.

D. Procedural Memory

Procedural memory is a type of long-term memory for the performance of particular types of action. Procedural memory stores knowledge about what to do and when to do it. In ACT-R, 4CAPS, SOAR etc., procedure knowledge is encoded as situation-action rules which provide an efficient and scalable representation.

In CAM procedural knowledge is represented in DDL. In order to be compatible with atomic actions described by Baader et al.'s formalism [23], we extend atomic action definitions of DDL($X^@$) to include occlusions and conditional post-conditions. With respect to a TBox T , an extended atomic action definition of DDL($X^@$) is of the form $\alpha \equiv (P, O, E)$, where [25]

– $\alpha \in N_A$;
– P is a finite set of ABox assertions for describing the pre-conditions;
– O is a finite set of occlusions, where each occlusion is of the form $A(p)$ or $R(p, q)$, with A a primitive concept name, R a role name, and $p, q \in N_I$;
– E is a finite set of conditional post-conditions, where each conditional postcondition is of the form ϕ/φ with ϕ an ABox assertion and φ a primitive literal.

In the above definition, the pre-conditions specify under which conditions the action is applicable. Each conditional post-condition ϕ/ψ says that, if ϕ is true before executing the action, then ψ should be true after the execution. The occlusions indicate those primitive literals that can change arbitrarily as while as the action is executed.

As a result, procedural knowledge can now be represented in *DDL($X^@$)* with extended atomic action definitions. For example, *BuyBookNotified*(*Tom,Kin*) might be described by the following extended atomic action definition:

BuyBookNotified(*Tom,Kin*)≡
({ *customer(Tom), book*(*KingLear*)}, { },
{ *instore*(*KingLear*)/*bought*(*Tom, KingLear*),
instore(*KingLear*)/¬*instore*(*KingLear*),
instore(*KingLear*)/*noti fy*(*Tom, Noti fyOrderSucceed*),
¬*instore*(*KingLear*)/*noti fy*(*Tom, Noti fyBookOutOf Stock*) })

where *notify* is a new introduced role name, and both *NotifyOrderSucceed* and *NotifyBookOutOfStock* are new introduced individual names. According to this description, if the book *KingLear* is in store before executing the action, then the formulas *bought*(*Tom, KingLear*), ¬ *instore*(*KingLear*) and *notify*(*Tom, Noti fyOrderSucceed*) will be true after the execution; otherwise, the formula *notify*(*Tom, NotifyBookOutOf Stock*)will be true after the execution, which means that Tom is notified that the book is out of stock.

In event detection example, the procedure memory stores a set of action to perform event based reaction. For example, action named *RecordShopping* may be a possible action in procedural memory, which is used to count the number of shopping events that happened in the video. The *RecordShoping* is defined as:

$$\begin{aligned}&RecordShopping(x?,y?,z?)\equiv\\&\quad(\{human(x?),trajectory(y?),\\&\quad hasTrajectory(x?,y?)\},\{\},\\&\quad\{ShoppingTrajectory(y?)/recordEvent(z?),\\&\quad\neg ShoppingTrajectory(y?)/\neg recordEvent(z?)\})\end{aligned}$$

$x?,y?,z?$ are three variables which can be replaced by the identity when the system performs reasoning process. The *RecordShopping* describes that if $x?$ is a human, $y?$ is a trajectory and $y?$ belongs to $x?$, then the action can be performed. If *RecordShopping* action is performed, then is $y?$ is a *ShoppingTrajectory* then *recordEvent*($z?$) become true, else the $\neg$ *recordEvent*($z?$) become true. When *recordEvent*($z?$) turn to true, then CAM system will record a shopping event, when$\neg$*recordEvent*($z?$) is true, then no event record action is performed.

In order to allow the information stored in goal memory to control the selection and execution of actions, we further extend *RecordShopping* by adding a *inShoppingRecord*($s?$) concept in the pre-condition of RecordShopping. Thus, if goal memory contains a $\neg$inShoppingRecord($s?$) formula, then the *RecordShopping* action won't be executed.

V. Conclusions

Memory is a fundamental component in human brain and plays very important roles for all mental processes. The analysis of memory systems through cognitive architectures can be performed at the computational, or functional level, on the basis of empirical data. We have introduced the extended CAM architecture, a model of the human mind. All knowledge contained in working memory, semantic memory, episodic memory and procedural memory are represented in dynamic description logic (DDL), which is a formal logic for description and reasoning about dynamic application domains characterized by actions.

In addition to its application to Neuro-Cognitive Robots, we are exploring the adaptation of CAM to brain-machine integration systems, which seek to combine biological intelligence with high computer performance to reach human-level artificial intelligence.

Acknowledgment

Authors would like to thank Prof. Liang Chang for his commences on dynamic discription logic.

References

[1] Ron Sun. Memory systems within a cognitive architecture. New Ideas in Psychology 30 (2012) 227–240.
[2] Anderson, J. R., and Lebiere, C. The Atomic Components of Thought. Mahwah, NJ: Lawrence Erlbaum, 1988.
[3] Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Qin Y. An Integrated Theory of the Mind. Psychological Review, volume 111, pp. 1036-1060, 2004.
[4] Laird, J. E., Newell, A., and Rosenbloom, P. S. Soar: An Architecture for General Intelligence. Artificial Intelligence, volume 33, pp. 1-64, 1987.
[5] Laird, J. E. The Soar Cognitive Architecture. The MIT Press, 2012.
[6] Tambe, M., Johnson, W. L. and Jones, et al. Intelligent Agents for Interactive Simulation Environments," AI Magazine, 16 (1): 15-40, 1995.
[7] Carbonell, J.G., Knoblock, C.A., and Minton, S. PRODIGY: an integrated architecture for planning and learning. Architectures for Intelligence, 1990.
[8] Pat Langley. Cognitive Architectures and General Intelligent Systems. AI Magazine, 27(2): 33-44, 2006.
[9] Franklin, S., A. Kelemen, and L. McCauley. IDA: A Cognitive Agent Architecture. In IEEE Conf on Systems, Man and Cybernetics. : IEEE Press,1998.
[10] Baars, Bernard J. A cognitive theory of consciousness. Cambridge: Cambridge University Press, 1988..
[11] Baars, B. J., and S. Franklin. How conscious experience and working memory interact. Trends in Cognitive Science 7:166-172, 2003.
[12] Ramamurthy, U., Baars, B., D'Mello, S. K., & Franklin, S. LIDA: A Working Model of Cognition. Proceedings of the 7th International Conference on Cognitive Modeling. Eds: Danilo Fum, Fabio Del Missier and Andrea Stocco; pp 244-249. Edizioni Goliardiche, Trieste, Italy, 2006.
[13] Friedlander, D., Franklin, S. LIDA and a Theory of Mind. Proceedings of the First AGI Conference, Pages 137-148, 2008.
[14] Ben Goertzel, Hugo de Garis, Ruiting Lian, etc. OpenCogBot: Achieving Generally Intelligent Virtual Agent Control and Humanoid Robotics via Cognitive Synergy. Progress of Advanced Intelligence, Vol. 2， 47-58， 2010.
[15] Rosenbloom, P. S. Towards a new cognitive hourglass: Uniform implementation of cognitive architecture via factor graphs. In Proceedings of the 9thInternational Conference on Cognitive Modeling, 2009.
[16] Rosenbloom, P. S. Extending mental imagery in Sigma. Proceedings of the 5th Conference on Artificial General Intelligence, 2012.
[17] Eliasmith, C., Stewart T. C., Choo X., Bekolay T., DeWolf T., Tang Y., Rasmussen, D. A large-scale model of the functioning brain. Science, Vol. 338, 1202-1205, 2012.
[18] Zhongzhi Shi. On Intelligence Science and Recent Progress. Invited Speaker, ICCI2006, Beijing, 2006.
[19] Zhongzhi Shi, Mingkai Dong, Yuncheng Jiang, and Haijun Zhang. A Logic Foundation for the Semantic Web. Science in China, Series F Information Sciences, 48(2): 161-178, 2005.
[20] Stefano S, Christine P, Luisa DM et al. A conceptual view on trajectories. Data Knowledge Engineering, 65(1):126–146, 2008.
[21] Baddeley A. The episodic buffer: a new component of working memory? Trends Cogn. Sci. (Regul. Ed.), 4 (11): 417–423, 2000.
[22] Xiaofeng Wang, Gang Li, Guang Jiang, Zhongzhi Shi, Semantic trajectory-based event detection and event pattern mining. Knowledge and Information Systems, November 2013, 37(2): 305-329.
[23] Nuxoll, A., Larid, J. E. Extending Cognitive Architecture with Episodic Memory. In Proceedings of the 22nd AAAI Conference on Artificial Intelligence, pp. 1560-1564, 2007.
[24] Baader, F., Lutz, C., Milicic, M., Sattler, U., Wolter., F. Integrating description logics and actionformalisms: first results. In: Proc. of the 12th Nat. Conf. on Artificial Intelligence (AAAI'05),pp. 572–577. AAAI Press / MIT Press, 2005.
[25] Liang Chang, Zhongzhi Shi, Tianlong Gu, Lingzhong Zhao, A Family of Dynamic Description Logics for Representing and Reasoning About Actions. J. Autom. Reasoning, 49(1): 1-52, 2012.
[26] Pentti O.Haikonen, The Cognitive Approach to Conscious Machines, Imprint Academic, UK, 2003
[27] Jansusz A. Starzyk, Dilip K. Prasad. A Computational Model of Machine Consciousness, International Journal of Machine Consciousness, 2010
[28] Chella, A. Frixone, M. Gaglio, S. A Cognitive Architecture for Robot Self-Consciousness. Artificial Intelligence in Medicine, 44, 147-154, 2008
[29] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, Peter F. Patel-Schneider, The Description Logic Handbook: Theory, Implementation and Application, Cambridge University Press, 2003.