Cognitive architectures

Stefan Profanter

December 6, 2012

Hauptseminar Human-Robot Interaction

Wintersemester 2012/13

Technische Universität München

Lehrstuhl für Echtzeitsysteme und Robotik

Contents

1 Why do we need cognitive architectures?

A cognitive architecture is a *blueprint for an intelligent agent.* This blueprint consists of its representational assumptions, the characteristics of its memories, and the processes that operate on those memories. It is a model which tries to describe formally mental or cognitive capabilities (most often like a human) to implement those on a computer. Those models must be defined formally enough so that they can be a basis of a computer program. Here not only the behavior but also the structure of the modelled system/capability is described.
The main field of research is the *imitation* of the cognition *of animate being* - especially humans - including memory, speech, perception, problem solving, spiritual will or also attention.
Cognitive architectures can be seen as a restriction of artificial intelligence. Because in artificial intelligence, agents can use - in contrary to cognitive architectures - also strategies, which aren't used or applied by humans or animate beings.

Cognitive architectures are mainly used to understand how the environment of the system might be and how problems can be solved. Problem solving means, to find a way which goes from a predefined start state into a predefined goal state.
Such cognitive architectures allow the system to *act efficiently and autonomously*, to adapt to new situations and to improve itself. So cognitive systems exhibit effective behavior through perception, action, deliberation, communication and through interaction with the environment. The main characteristic of these systems is that they can act even in circumstances which weren't planned or weren't available when the system was designed.

2 Additional References

The main topics of this summary are explained in detail in "A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agent" published by David Vernon [VMS07, Ver12] on which this elaboration is based.

The Paper "The Past, Present, and Future of Cognitive Architectures" [TA10] gives you a quick *historical introduction* to cognitive architectures, starting with general problem solvers (GPS, 1963) trying to imitate the humans problem solving capability and explaining how GPS had influence on future cognitive architectures or how it is successfully used in human-like warfare simulations. Taatgen then describes some recent changes in cognitive modeling approaches, e.g. using neurons in neural networks or mapping components of cognitive architectures onto regions of the human brain.
He also states that "current models have an almost infinite freedom in specifying the initial knowledge and strategies of the model, allowing many different ways to model a single task". So the current problem is, that there are a lot of different cognitive architectures and models available. Taatgen says that in the future, researchers will try to solve this problem by simplifying models and merging ideas of different models.

In the paper by [DOP08] with the title 'Cognitive Architectures: Where do we go from here?' you can find an easy understandable introduction in the so called *artificial general intelligence* (AGI). This paper is from the year 2008 and is designed as a critical survey of the state of the art in cognitive architectures, e.g. SOAR, EPIC, ICARUS, ACT-R, ..., by this time.

Duch also tries to define some base properties which a cognitive architecture should have and how cognitive systems can be tested on how 'intelligent' they are. He also gives some ideas for developers and designers of cognitive architectures.

If you are interested in a more *practical related* reading about cognitive architectures you may find [SMV07] interesting; This paper describes the *iCub Cognitive Humanoid Robot*, which is an open-systems 53 degree of freedom, children sized robot. Here also the importance of humanoid embodiment for cognitive architectures is discussed. It contains also a digression concerning iCub's mechanical and electronic specifications.

A more general and future related definition of cognitive computer systems gives Brachman in his paper "Systems That Know What They're Doing" [Bra02]. He states that cognitive systems should, in addition to being able to reason, to learn from experience, to improve its performance with time, and to respond intelligently to things it is never encountered before, would also be able to *explain what it is doing* and why it is doing it.
This helps systems on finding unresolvable constraints with new tasks or deciding if they need more information for solving a specific task.
Brachman additionally proposes to use three types of processes operating most likely parallel to build a cognitive architecture. The *reactive* processes; these are things we do without thinking, e.g. simple reflexes or driving a car (can also contain newly learned "reflexes"). The *deliberative* processes, which make up the bulk of what we mean by the word "thinking", which decides for example, in which direction to go next, or planning a vacation. The third process - *reflective* - is the most important one and gives the system the ability to stop basic reasoning on a problem and step back and use an alternative approach.

On the page `http://cogarch.org` you get a Wikipedia-Style overview on general information and features of cognitive architectures and you will also find information on a lot of different individual architectures.
You can also find detailed information on different cognitive architectures in Table 1 and the corresponding cited papers.

3 Theory

David Vernon, a professor at the Technische Universität München, defines a cognitive system as an "autonomous system that can perceive its environment, learn from experience, anticipate the outcome of events, act to pursue goals, and adapt to changing circumstances." [Ver12].
A cognitive architecture is a *blueprint for intelligent agents*. It proposes computational processes that act, most often, like a person, or *acts intelligent* under some definition. The term architecture implies that not only the behavior, but also the structural properties should be modelled.
Cognitive architectures can be characterized by the following properties:

- Cognition is implemented *as a whole* (Unified theory of cognition), not just small aspects of cognitive behavior.

- The architecture often tries to reproduce the *time constraints* (reaction within a specific amount of time) of the modelled system.
- The system should be *robust in the face of error*, the unexpected and the unknown.
- A cognitive system should be able to *learn*.
- The system shouldn't depend on *parameter tuning*, it should be parameter-free (in contrast to Artificial neural networks).

The first one who mentioned cognitive architectures in connection with artificial intelligence was Allen Newell in the year 1990 where he describes the unified theory of cognition (UTC) [New90]. UTC describes how an intelligent system reacts to stimuli from the environment, how goal-directed behavior is exhibited, how goals are acquired rationally, how the knowledge is represented and how it can be expanded by learning.
The inspiration for such architectures can be traced back to Alan Turing (1950). Turing thought that the main barriers for computers of that time is speed and memory capacity. But history has shown, that the puzzle of human intelligence, creativity, and ingenuity is much more complex.

3.1 Different paradigms of cognition

There are two main approaches to model cognitive architectures: the *cognitivist* approach based on symbolic information processing and the *emergent systems* approach which groups together connectionist, dynamical and enactive systems [VMS07].
In the following, there is are mentioned different references to symbols. A symbol in the sense of artificial intelligence is a representation to describe concepts of different perception- and cognitive processes. Symbols are used when it is too complex to describe such concepts with numbers, bacause symbols use characters and words for description.

Cognitivist and emergent approaches can be compared on twelve distinct characteristics:

- **Computational Operation**

Cognitivist	Emergent
use rule base manipulation of symbol tokens, typically in a sequential manner	exploit processes of self-organization, self-production, self-maintenance, and self-development through a network of distributed, concurrent interacting components

- **Representational Framework**

Cognitivist	Emergent
use patterns of symbol tokens, designed by human, that refer to events in the external world	global system states are encoded in the dynamic organization of the system's distributed network of components

- **Semantic Grounding**

Cognitivist	Emergent
Symbolic representations are grounded through percept-symbol identification by either the designer or by learned association and can be interpreted directly by human	representations ground on autonomy-preserving anticipatory and adaptive skill construction. These representations are inaccessible for direct human interpretation

- **Temporal Constraints**

Cognitivist	Emergent
not necessarily entrained by events in the external world	operate synchronously in real-time with events in their environment

- **Inter-Agent Epistemology**

Cognitivist	Emergent
epistemology can be shared between agents because of their common view of reality	epistemology is the subjective outcome of a history of shared consensual experiences among phylogenetically compatible agents

- **Embodiment**

Cognitivist	Emergent
cognition is independent of the physical platform in which it is implemented	intrinsically embodied, the physical instantiation plays a direct constitutive role in the cognitive process

- **Perception**

Cognitivist	Emergent
perception provides an interface between the external world and the symbolic representation of that world	perception is a change in system state in response to environmental perturbations in order to maintain stability

- **Action**

Cognitivist	Emergent
actions are causal consequences of symbolic processing of internal representations	actions are perturbations of the environment by the system

- **Anticipation**

Cognitivist	Emergent
anticipation typically takes the form of planning using some form of procedural or probabilistic reasoning	anticipation requires the system to visit a number of states in its self-constructed perception-action state space without commiting to the associated actions

- **Adaptation**

Cognitivist	Emergent
adaptation usually implies the acquisition of new knowledge	entails a structural alteration or reorganization to effect a new set of dynamics

- **Motivation**

Cognitivist	Emergent
impinge on perception, action and adaption to resolve an impasse	enlarge the space of interaction

- **Relevance of Autonomy**

Cognitivist	Emergent
Autonomy is not necessarily implied	autonomy is crucial since cognition is the process whereby an autonomous system becomes viable and effective

3.1.1 Cognitivist Approach

In 1956 the science of cognition formed the term cognitivism. Cognitivism describes cognition based on computations defined over *internal symbolic representations* (which is the knowledge of the system). The information about the world is abstracted by perception, represented using some appropriate symbolic data-structure, reasoned about, and then used to plan and act in the world. The approach has also been labelled by many as the information processing (or symbol manipulation) approach to cognition. Cognitivism has been the predominant approach and to date it is still prevalent.
In cognitivist approaches, the symbolic representations are the descriptive product of a *human designer*. Therefore they can be accessed (extracted and embedded) directly and understood or interpreted by humans. This fact is often seen as a limitation factor of cognitivist systems: the representations are programmer dependent and constrain the system to an idealized description dependent on the cognitive requirements of human activity. It is possible to extend the predesigned knowledge with *machine learning algorithms*, probabilistic modeling or other techniques to deal with the uncertain, time-varying and incomplete nature of sensor data used by the representational framework. However this wouldn't eliminate the introduced limitation by the designers description.
In the cognitivist paradigm, the focus in a cognitive architecture is on the aspects of cognition that are constant over time and that are relatively independent of the task.
Therefore the strength of such systems is to *capture statistical regularities* for example in training data.

The main problems of cognitivist systems are the symbol grounding (symbols designed by human) and finding significant properties in a large dataset and then generalize to accommodate new data. Therefore such models have difficulties handling complex, noisy and dynamic environments. It is also very difficult to gather higher order capabilities such as creativity or learning. So these models are best suited for *well defined problem domains.*

3.1.2 Emergent Approaches

In Emergent approach, cognition is the process of *adapting to the environment.* Such systems does so through a process of self-organization through which it reacts on the environment in real-time to maintain its operability. In other words, the ultimate goal of an emergent cognitive system is to *maintain* its own *autonomy.* This process of making sense of its environmental interactions is one of the foundations of the enactive approach to cognition.
The cognitive agent constructs its internally represented reality (its world) as a result of its operation, its experiences in that world. Therefore for emergent systems, in contrary to cognitivist models, perception is the acquisition of sensor data. It isn't a process whereby the environment is abstracted and represented in a more or less isomorphic manner.
In contrast to the cognitivist approach, many emergent approaches assert that the primary model for cognitive learning is anticipative *skill construction* rather than knowledge acquisition.
Emergent systems have also the possibility to get familiar with and learn how to control the body it is embodied in. So the designer doesn't have to model each body-characteristic into the system.

Due to the mutual specification and co-development of emergent systems with the environment, it is very difficult for us humans to model such systems. Til now researchers only provided general modeling frameworks but no specific, fully defined model of cognition.

Connectionist Models Connectionist systems use statistical properties, rather than logical rules to process information to achieve effective behavior. Therefore such systems best capture the *statistical regularities* in training data. A prominent example for a connectionist model is the unsupervised neural training algorithm by Hebb (1949). Later on, connectionist systems were used to retrieve specific and general information from stored knowledge about specific instances. One of the key features of connectionist models is that the system becomes inseparable from its history of transformations and the task defined for the system. Furthermore connectionist models don't use a symbol representation of the knowledge, instead "meaning" is a description attributed by an *outside agent*.

In 1976 Grossberg introduced the Adaptive Resonance Theory (ART) [Gro76] which tries to imitate the brains information processing. ART describes *neural network* models used for pattern classification and prediction.

Nearly at the same time, in 1982, Kohonen's *self-organizing maps* (SOMs) were first mentioned [Koh82]. A SOM is a type of artificial neural network that is trained to produce a low-dimensional discretized representation of the input space.

Another popular representative of Connectionist systems is the Hopfield net [Hop82]. *Hopfield nets* serve as an associative memory and are for example widely used for pattern completion.

Dynamical Systems Models A dynamical system is a system of a large number of interacting components with a large number of degrees of freedom which needs external sources of energy to maintain structure or function. But only a small number of the system's degrees of freedom (DOF) contribute to its behavior. These DOF are called *order parameters*. Therefore dynamical systems are used to characterize the behavior of a high-dimensional system with a low-dimensional model which is one of the features that distinguishes dynamical systems from connectionist systems.

In contrary to connectionist systems which describe the dynamics in a very high-dimensional space, dynamical models describe the dynamics in a *low-dimensional space* where a small number of state variables capture the behavior of the system as a whole.

Enactive Systems Models Enactive systems take the emergent paradigm even further: properties of a cognitive entity are co-determined by the entity as it interacts with the environment in which it is embedded. Thus, *nothing is pre given* and hence there is no need for symbolic representations. The goal of enactive systems research is the complete treatment of the nature and development of autonomous, cognitive, social systems. The system builds its own understanding as it develops a cognitive understanding by co-determined *exploratory learning*. An enactive system generates its own models of how the world works and that the purpose of these models is to preserve the system's autonomy.

Emergent systems follow the principle that the perception of its body and the dimensionality and geometry of the space in which it is embedded can be deduced (learned or discovered) by the system from an analysis of the dependencies between *motoric commands* and *consequent sensory data*, without any knowledge or reference to an external model of the world or the physical structure of the organism [PON02][PONC03].

3.1.3 Hybrid Models

Hybrid models combine aspects of the emergent systems and cognitivist systems. The main idea behind hybrid models is to *avoid explicit programmer-based knowledge* in the creation of artificially intelligent systems and to use *perception-action behaviors* rather than the perceptual abstraction of representations. Typically, hybrid systems exploit symbolic knowledge to represent the agent's world

and logical rule-based systems to reason about this knowledge in order to achieve goals and select actions, while at the same time using emergent models of perception and action to explore the world and construct this knowledge. Thus, hybrid systems still use cognitivist representations and representational invariances but they are constructed by the system itself as it interacts with and explores the world rather than through a priori specification or programming.
Out of this, objects should be represented as invariant combinations of precepts and responses, where the object properties need to be learned through *object manipulation*. Thus the system's ability to interpret unknown objects is dependent on its ability to flexibly interact with it. This implicates that the internal representation of objects don't have any meaningful semantic meaning for humans. For object manipulation, the system must also be embodied in some kind (at least during the learning process).
The aim for developing such systems is, that they learn tasks which they *weren't explicitly designed for.*

4 Examples & Applications

A cognitive architecture defines how memories are stored and the processes that operate on those memories. In terms of cognitive models it defines the formalisms for knowledge representations and the learning mechanisms that acquire it.
For emergent systems it is much more complicated to predefine an exact structure because the designer doesn't know exactly what the system will learn in the future and how this knowledge is connected. Therefore architectures for emergent systems represent an initial point of departure for the base cognitive system and they provide the basis and mechanism for tis subsequent autonomous development, a development that may impact directly on the architecture itself.

Cognitive architectures can be used for a lot of different use cases. A cognitivist system, for example, has been used to develop a cognitive vision system which interprets a video sequence of traffic behavior and then generates a natural language description of the observed environment [Nag04].
Emergent connectionist models can be used, for example, to learn hand-eye coordination with Kohonen neural network [JV94, Mel88] or much simpler for face detection. As another example, a biologically inspired hybrid model is used in [MF03] for object segmentation, recognition, and localization capabiliteis without any prior knowledge through exploratory reaching and simple manipulation.

In the following paragraphs we will review some of the most important cognitive architectures of all three types: cognitivist, emergent and hybrid. For additional examples of cognitive architectures see Table 1 and the corresponding referenced resources for further reading.

4.1 The Soar Cognitive Architecture

The Soar (State, Operator and Result) system operates in a cyclic manner, with a *production cycle* and a *decision cycle*. In the first cycle, all productions that match the contents of declarative (working) memory fire. A production rule can also be seen as a current state, e.g. the current position in a maze. A production that fire (movement in the maze) may alter the current declarative memory (new position in maze) and cause other productions to fire. This loop is repeated until no more productions

Cognitivist	**Emergent**	**Hybrid**
Soar [Lai12, Ros93, JJP96]	**AAR** [Bro85]	HUMANOID [BMS+05]
EPIC [AM97]	Global Workspace [Sha05]	Cerebus [Hor01a, Hor01b]
ACT-R [And96]	I-C SDAL [CH00]	Cog: Theory of Mind [RCM+99]
ICARUS [LC06]	SASE [Juy04]	**Kismet** [Bre00, Bre03]
ADAPT [DDD04]	Darwin [KE08]	

Table 1: *Cognitive architecture examples. The emphasized ones are presented in Subsection 4. For additional information see given references. (Source: [VMS07])*

fire. At this point the decision cycle starts in which a single action from several possible actions is selected, based on the action's preferences.
If the cycle reaches an impasse, i.e., no action is available, a new state in a new problem state is set up. This process is called *subgoaling*, where the new goal resolves the impasse. Additionally a new production rule is created which summarizes the processing that occurred in solving the sub goal.
The subgoaling process is the only form of learning that occurs in Soar.

So the main properties of a Soar architecture are (see also Figure 1):

- Problem solving is realized as a search in problem spaces.
- Permanent knowledge is represented with production rules in production (long-term) memory. This knowledge may be in procedural, semantic and episodic form.
- Temporary knowledge (perception or previous results) is represented through objects stored in declarative (working) memory.
- A decision procedure decides what to do with the current knowledge (create new knowledge or do a specific action).
- New goals are created only if there is an impasse.

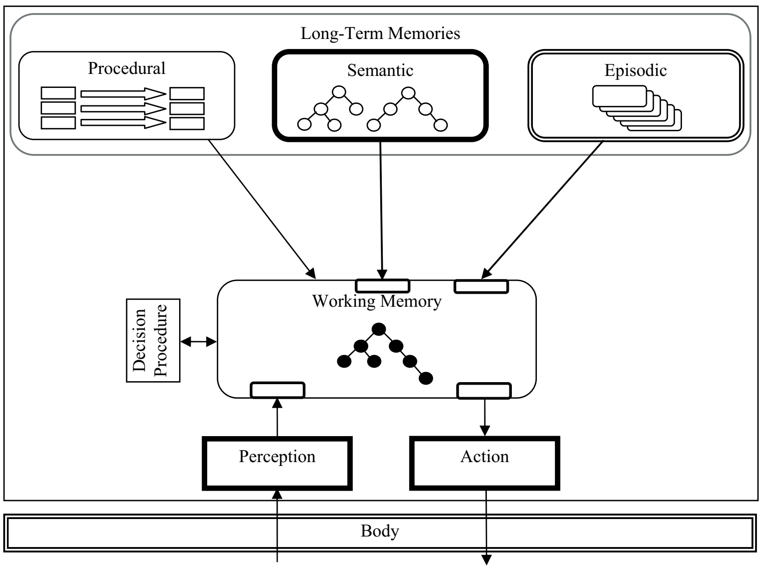

Figure 1: *Memory structures in Soar (Source: [JJP06])*

	production rule 1:
sp { propose*hello-world	propose to use operator 'hello-world'
(state <s> ^type state)	condition: if a state <s> exists in declarative directory, then fire rule (execute following actions)
-->	end of condition block, actions follow
(<s> ^operator <o> +)	action 1: propose to use operator <o> on current state
(<o> ^name hello-world)}	action 2: give <o> the name 'hello-world'
	production rule 2:
sp { apply*hello-world	Apply this rule if operator 'hello-world' should be executed
(state <s> ^operator <o>)	condition 1: an operator <o> has been selected
(<o> ^name hello-world)	condition 2: <o> has the name 'hello-world'
-->	end of condition block, actions follow
(write |Hello World|)	action 1: print 'Hello World' to console
(halt) }	action 2: stop problem solving process

Table 2: *Hello world program written in Soar programming language*

In Table 2 you can see an example program written in Soar. When loading this program, the two production rules are loaded into the production memory. Running an agent on this knowledge it produces 'Hello World'.
To run this program you need to download the Soar Suite from `http://sitemaker.umich.edu/soar/home`.

4.2 Adaptive Control of Thought-Rational (ACT-R)

The ACT-R cognitive architecture focuses on the *modular decomposition* of cognition and offers a theory of how these modules are integrated to produce coherent cognition. Each module processes a different kind of information (see Figure 2).
The *vision module* determines objects, the *manual module* is responsible for controlling the body (e.g. hands), the *declarative module* for retrieving information from long-term memory and a *goal module* for keeping track of the internal state when solving a problem. The fifth module, the production system in the center, coordinates the operation of the other four modules by using the module buffers to exchange information.
ACT-R operates in a *cyclic* manner where on each cycle the production system requests information from the modules by supplying constraints to it. The module places then a chunk which satisfies the given constraints in its buffer. The information of the buffers is then read, interpreted and new information may be requested or stored in those buffers.

Compared with Soar, this system has two bottlenecks:

- A buffer can only hold a single unit of knowledge (called 'chunk'). Therefore only one memory can be retrieved the same time.
- Only one production can fire in a cycle, compared with Soar where multiple productions could fire the same time.

Declarative knowledge effectively encodes things in the environment, while procedural knowledge encodes observed transformations. A central feature of the ACT-R cognitive architecture is that these two types of knowledge are tuned in specific application by encoding the statistics of knowledge. Thus, ACT-R learns sub symbolic information by *adjusting or tuning* the *knowledge parameters*. This sub-symbolic learning distinguishes ACT-R from the symbolic (production-rule) learning of Soar.

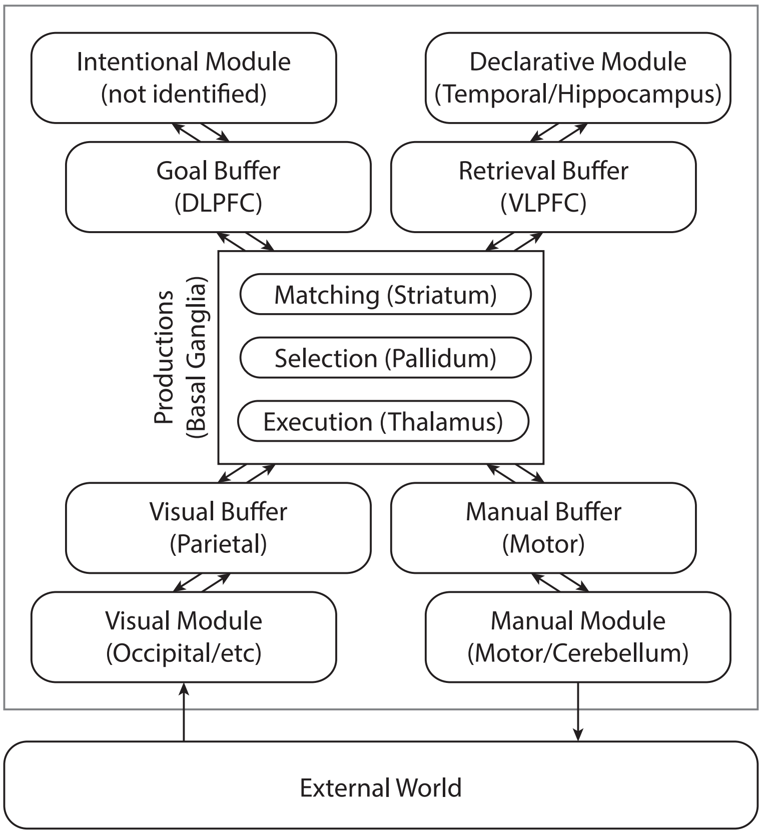

Figure 2: *The organization of information in ACT-R 5.0. Information in the buffers associated with modules is responded to and changed by production rules. The names in parentheses indicate the corresponding part of the human brain. DLPFC = dorsolateral prefrontal cortex; VLPFC = ventrolateral prefrontal cortex. (Source: [ABB+04])*

The Navy Center for Applied Research in Artificial Intelligence (NCARAI) in Washington DC performs state-of-the-art research on cognitive robotics and other AI relevant topics.
They are using ACT-R/E as a cognitive basis for their robot Octavia. ACT-R/E is built upon ACT-R with additional visual, auditory, motor and spatial modules to interact with the world [1]. In the video "Robotic Secrets Revealed, Episode 002" (on their videos page [2]) they show a very advanced example of human robot interaction:
In the first scene, the robot Octavia is presented with some items to identify. Tony, their human trainer, then leaves the room with the comment that they will do movement tests next, but have to wait for Laura. Then Laura comes in, says that there is a problem with other robots so the movement test can't take place and leaves the room. When Tony returns and wants to do movement tests,

[1] `http://www.nrl.navy.mil/aic/iss/aas/CognitiveRobots.php`
[2] `http://www.nrl.navy.mil/aic/iss/aas/CognitiveRobotsVideos.php`

Octavia is confused and tells Tony what it learned from Laura.
According to the explanation after the video, ACT-R/E was able to understand the statements given by human, learn from them and communicate the new facts to human. Additionally ACT-R/E allows Octavia to act and behave like human beings.

4.3 Autonomous Agent Robotics (AARs)

The main idea behind AARs and behavior-based systems is to avoid a decomposition of the system into functional components by using *subsumption*. Therefore this type or architectures is also often called 'Subsumption architecture' [3].
Subsumption means that at the bottom are simple whole systems that can act effectively in simple circumstances, layers of more sophisticated systems are added incrementally, each layer subsuming the layers beneath it.
This allows to break complicated intelligent behaviors into many simple behavior modules. The simple modules are organized into *layers*, each layer implementing a particular goal of the robot. This means for instance that a decision to move made by the Explore layer (see Figure 3) must take into account the decision of the 'Follow Light' layer directly below it, as well as the lowest layer labeled Avoid Obstacle.
Each layer can access all of the sensor data and generates commands for the actuators. And each separate tasks can suppress (or overrule) inputs or inhibit outputs. This way, the lowest layers can work like fast-adapting mechanisms (reflexes), while the higher layers work to achieve the overall goal.

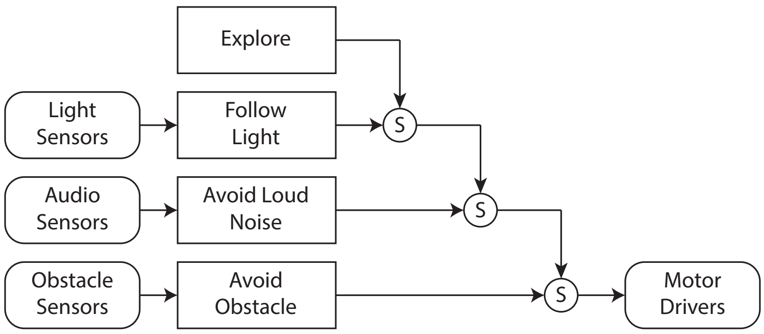

Figure 3: *Subsumption architecture. At the bottom is the lowest layer. Incrementally added layers subsume the layer beneath it. (Source:* `http://www.beam-wiki.org`*)*

4.4 Kismet

Kismet is a robotic head (see Figure 4) which can express a lot of different human like emotions. It has 21 degrees of freedom (DOF) for controlling the head orientation (3 DOF), the gaze (3 DOF) and its facial features (15 DOF). It additionally has a wide-angle binocular vision system and two microphones. It was designed to engage people in natural expressive face-to-face interaction.

[3] `http://ai.eecs.umich.edu/cogarch3/Brooks/Brooks.html`

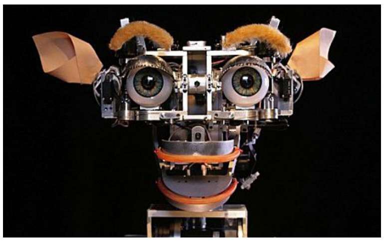

Figure 4: *The kismet robot head. (Source:* `http://programm.ard.de`*)*

Kismet has two types of motivations: *drives* and *emotions*. The drive subsystem regulates Kismet's social, stimulation and fatigue related needs. Like in an animal that has a level of hunger, each drive becomes more intense until it is satiated. These drives affect Kismet's emotion system, which contains anger, disgust, fear, joy, sorrow, surprise, boredom, interest, and calm. These emotional states can activate behaviors. For example, the fear emotion can induce the escape behavior.

Kismet's cognitive architecture consists of five modules: a perceptual system, an emotion system, a behavior system, a drive system and a motor system (see Figure 5).

External events, such as visual and auditory stimuli, are sensed by the robot and are filtered by a number of feature extractors (e.g. color, motion, pitch, etc.). This information together with affective input from the emotion system, input from the drive system and the behavior system, these features are bound by *releaser processes* that encode the robot's current set of beliefs about the internal and external state of the robot and its relation to the world. When the activation level of a releaser exceeds a given threshold (based on the perceptual, affective, drive, and behavioral inputs) it is output to the emotion system for appraisal.
The *appraisal process* tags the releaser output with pre-specified affective information on their arousal (how much it stimulates the system), valence (how much it is favored), and stance (how approachable it is). These are then filtered by 'emotion elicitors' to map each AVS (arousal, valence, stance) triple onto the individual emotions. A single emotion is then selected by a winner-take-all *arbitration process*, and output to the behavior system and the motor system to evoke the appropriate expression and posture.

Kismet is a hybrid system because it uses cognitivist rule-based schemas to determine the most suitable behavior an emotion, but allows the system behavior to emerge from the dynamic interaction between its subsystems.

5 Conclusion & Outlook

The presented types of models (cognitive, emergent, hybrid) have their own strengths and weaknesses, their proponents and critics, and they stand at different stages of scientific maturity. The arguments in favor of dynamical and enactive systems are compelling but current capabilities of cognitive systems are actually more advanced. Emergent systems, for example, have their strength in capturing the context-specificity of human performance and handling many pieces of low-level information simultaneously. But their main shortcoming is the *difficulty in realizing higher-order cognitive* functions.

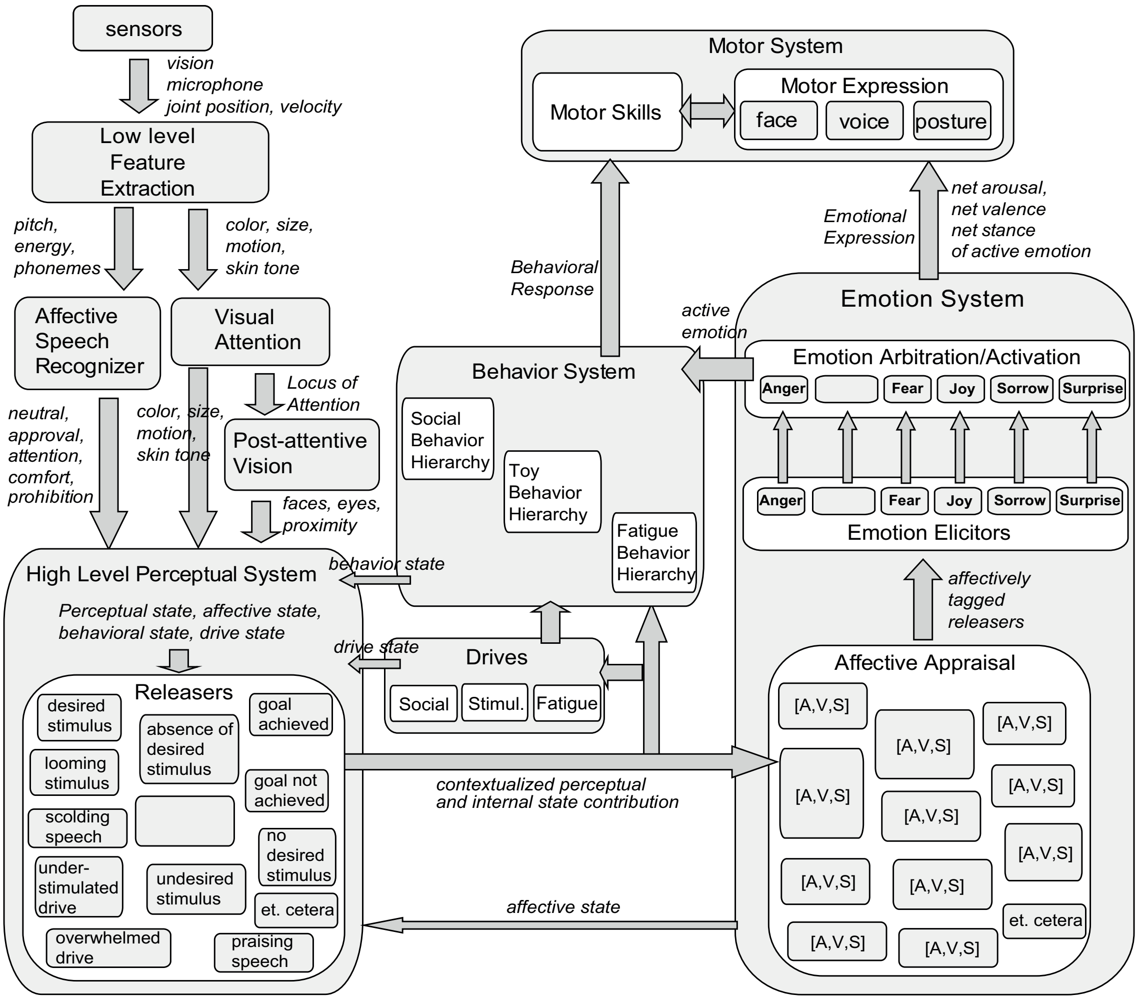

Figure 5: *An overview of Kismet's cognitive architecture. (Source: [Bre03])*

This suggests that hybrid approaches (synthesis of cognitivist and emergent approaches) may be the way forward, because they offer the best of both worlds - the adaptability and flexibility of emergent systems and the advanced starting point of cognitivist systems.

Designing a cognitive system and defining its architecture is a challenging discipline with several questions: Which physical embodiment (cognitivist systems don't depend on it, emergent systems, by definition, require embodiment) and what degree of autonomy is required? Which symbol-structure is needed for the knowledge representation? What is the goal specification? The given questions are only a rough overview, if you go more in detail, there will arise a lot more questions.

We have shown this freedom in design by presenting some examples for cognitive architectures, which differ strongly. Some are under active scientific development, others are popular and useful, but constraints are not well-presented.

Developing cognitive architectures, which behave similar to humans is a complex, not yet solved task which requires different approaches from all sides.

DARPA (Defense Advanced Research Projects Agency), for example, tried to launch the "Biologically inspired cognitive architectures" project in 2005 which was designed to create the next generation of Cognitive architectures that could be used to create embodied computational architectures of human intelligence. But the implementation of the previously designed ideas was too ambitious for this time and the project was therefore cancelled in 2007.
This DARPA project shows that it is worth to search for new approaches, even if they are ahead of the times.

Brachman presents the problem that machines don't know what they are doing, in a nice way [Bra02]:

> How many times have you watched something happen on your PC and just wished you could ask the stupid thing what it was doing and why it was doing it? How often have you wished that a system wouldn't simply remake the same mistake that it made the last time, but instead would learn by itself-or at least allow you to teach it how to avoid the problem? Despite our temptation to anthropomorphize and say that a system that is just sitting there doing nothing visible is "thinking," we all know that it's not. Indeed, if today's systems were people, we would say that they were totally clueless. The sad part is that even if my PC had a clue, it wouldn't know what to do with it.

Therefore, as already mentioned in Section 2, one of the new approaches in cognitive architectures will be, that cognitive systems are able to explain what they are doing and why they are doing it this specific way. Consequently, a cognitive system would be able to view a problem in several different ways and to look at different alternative ways of tackling it.

Beneath the design of intelligent systems we will also need a method (similar to the Robot@Home challenge) to test the system's intelligence and determine its cognitive abilities (e.g. 'clean up the house', 'do what this animal can do', ... [4]).
This testing and benchmarking is necessary to compare different architectures and to determine their completeness and performance.

[DOP08] mentioned already, that so far, cognitive architectures are used in very few real-world applications. They are mainly used in research and in small demo projects. So the next step with respect to real world applications, will be to extend those small demos to large scale applications or implementing cognitive abilities in everyday items, e.g. the current smartphones boom would be a great opportunity, to reach a mass of people and make cognitive architectures even more interesting for developers.

References

[ABB+04] Anderson, J. R., D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere and Y. Qin: *An integrated theory of the mind.* Psychological Review, 111:1036–1060, 2004.

[AM97] Anderson, John and Michael Matessa: *An Overview of the EPIC Architecture for Cognition and Performance With Application to Human-Computer Interaction.* Human-Computer Interaction, 12(4):391–438, 1997.

[4] `http://www.eucognition.org/eucog-wiki/Challenges_for_Artificial_Cognitive_Systems`

[And96] Anderson, John R.: *ACT: A simple theory of complex cognition.* American Psychologist, 51(4):355–365, 1996.

[BMS+05] Burghart, C., R. Mikut, R. Stiefelhagen, T. Asfour, H. Holzapfel, P. Steinhaus and R. Dillmann: *A cognitive architecture for a humanoid robot: a first approach.* In *5TH IEEE-RAS International Conference On Humanoid Robots*, pages 357–362, 2005.

[Bra02] Brachman, R.J: *Systems that know what they're doing.* IEEE Intelligent Systems, 17(6):67–71, 2002.

[Bre00] Breazeal, Cynthia L.: *Sociable machines: expressive social exchange between humans and robots.* PhD thesis, Massachusetts Institute of Technology, Massachusetts, 2000.

[Bre03] Breazeal, Cynthia: *Emotion and sociable humanoid robots.* International Journal of Human-Computer Studies, 59(1-2):119–155, 2003.

[Bro85] Brooks, Rodney A.: *A Robust Layered Control System For a Mobile Robot*, 1985.

[CH00] Christensen, W. D. and C. A. Hooker: *An interactivist-constructivist approach to intelligence: Self-directed anticipative learning.* Philosophical Psychology, 13(1):5–45, 2000.

[DDD04] D. Paul Benjamin, Damian Lyons and Deryle Lonsdale: *ADAPT: A cognitive architecture for robotics.* In *in Proceedings of the International Conference of Cognitive Modeling*, 2004.

[DOP08] Duch, Włodzisław, Richard J. Oentaryo and Michel Pasquier: *Cognitive Architectures: Where do we go from here?* Proceedings of the 2008 conference on Artificial General Intelligence 2008, pages 122–136, 2008.

[Gro76] Grossberg, S.: *Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors.* Biological Cybernetics, 23(3):121–134, 1976.

[Hop82] Hopfield, J. J.: *Neural Networks and Physical Systems with Emergent Collective Computational Abilities.* Proceedings of the National Academy of Sciences, 79(8):2554–2558, 1982.

[Hor01a] Horswill, I.: *Tagged behavior based systems: integrating cognition with embodied activity.* IEEE Intelligent Systems, 16(5):30–37, 2001.

[Hor01b] Horswill, Ian: *Cerebus: A Higher-Order Behavior-Based System.* AI Magazine, 2001.

[JJP96] Jill Fain Lehman, John Laird and Paul Rosenbloom: *A gentle introduction to Soar, an architecture for human cognition.* In *In S. Sternberg & D. Scarborough (Eds), Invitation to Cognitive Science.* MIT Press, 1996.

[JJP06] Jill Fain Lehman, John Laird and Paul Rosenbloom: *A gentle introduction to Soar, an architecture for human cognition.* PhD thesis, University of Michigan EECS Department, Michigan, 2006.

[Juy04] Juyang Weng: *Developmental robotics: Theory and experiments.* International Journal of Humanoid Robotics, 1:2004, 2004.

[JV94] Jones, Marggie and David Vernon: *Using neural networks to learn hand-eye coordination.* Neural Computing & Applications, 2(1):2–12, 1994.

[KE08] Krichmar, Jeffrey L. and Gerald M. Edelman: *Design Principles and Constraints Underlying the Construction of Brain-Based Devices.* In Springer-Verlag (editor): *Neural Information Processing*, volume 4985, pages 157–166. Springer-Verlag, Berlin, 2008.

[Koh82] Kohonen, Teuvo: *Self-organized formation of topologically correct feature maps.* Biological Cybernetics, 43(1):59–69, 1982.

[Lai12] Laird, John: *The Soar cognitive architecture.* MIT Press, Cambridge and Mass. and and London and England, 2012.

[LC06] Langley, Pat and Dongkyu Choi: *A Unied Cognitive Architecture for Physical Agents.* In *Twenty-first National Conference on Artificial Intelligence (AAAI-06) : Eighteenth Innovative Applications of Artificial Intelligence Conference (IAAI-06)*, Menlo Park and Calif, 2006. AAAI Press.

[Mel88] Mel, Bartlett W.: *MURPHY: A Robot that Learns by Doing.* In Anderson, Dana Z. (editor): *Proc. First IEEE Conf. on Neural Information Processing Systems*, pages 544–553, Piscataway and NJ, 1988. IEEE Service Center.

[MF03] Metta, Giorgio and Paul Fitzpatrick: *Early integration of vision and manipulation.* ADAPTIVE BEHAVIOR, 11:109–128, 2003.

[Nag04] Nagel, Hans-Hellmut: *Steps toward a cognitive vision system.* AI Magazine, 25(2):31–50, 2004.

[New90] Newell, Allen: *Unified theories of cognition.* Harvard University Press, Cambridge and Mass, 1 edition, 1990.

[PON02] Philipona, D., J. K. O'Regan and J. Nadal: *Is there something out there? Infering space from sensorimotor dependencies.* Neural Computation, 15:2003, 2002.

[PONC03] Philipona, D., J. K. O'Regan, J. Nadal and D. Coenen: *Perception of the Structure of the Physical World Using Unknown Multimodal Sensors and Effectors.* In *Advances in Neural Information Processing Systems*, pages 945–952. MIT Press, 2003.

[RCM+99] Rodney A. Brooks, Cynthia Breazeal, Matthew Marjanović, Brian Scassellati and Matthew M. Williamson: *The Cog project: Building a humanoid robot.* In Springer-Verlag (editor): *Computation for metaphors, analogy, and agents*, pages 52–87. Springer, Berlin, 1999.

[Ros93] Rosenbloom, Paul S.: *The Soar papers: Research on integrated intelligence.* MIT Press, Cambridge and Mass. [u.a.], 1993.

[Sha05] Shanahan, Murray: *A cognitive architecture that combines internal simulation with a global workspace*, 2005.

[SMV07] Sandini, Giulio, Giorgio Metta and David Vernon: *The iCub Cognitive Humanoid Robot: An Open-System Research Platform for Enactive Cognition.* In Lungarella, Max, Fumiya Iida, Josh Bongard and Rolf Pfeifer (editors): *50 Years of Artificial Intelligence*, volume 4850 of *Lecture Notes in Computer Science*, pages 358–369. Springer Berlin Heidelberg, Berlin and Heidelberg, 2007.

[TA10] Taatgen, Niels and John R. Anderson: *The Past, Present, and Future of Cognitive Architectures.* Topics in Cognitive Science, 2(4):693–704, 2010.

[Ver12] Vernon, David: *Cognitive System.* Encyclopedia of Computer Vision, 2012.

[VMS07] Vernon, David, Giorgio Metta and Giulio Sandini: *A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents.* IEEE Transactions on Evolutionary Computation, 11(2):151–180, 2007.