A Comparison among Cognitive Architectures: A Theoretical Analysis

Danilo Lucentini
Ricardo Gudwin

DCA-FEEC-UNICAMP

danilolucentini@gmail.com
gudwin@dca.fee.unicamp.br
A **Cognitive Architecture** is a theoretical/computational model that aims at describing, as accurately as possible, the underlying infrastructure for an intelligent system, explicitly emphasizing its cognitive capabilities or cognitive functions.

The focus of this paper is to compare three of the most popular cognitive architectures: CLARION, LIDA, SOAR.

- Criteria for the choice of the architectures:
  - Executable code must be available for comparison
  - Existence of web-site, literature and instructional material
  - Existence of examples on how to use the architecture

The comparison is based on a chosen set of cognitive functions present in the human cognitive cycle: Input Data/Perception, Goals, Action Selection, Learning.
SOAR

Symbolic Long-Term Memories
- Procedural
- Semantic
- Episodic

Symbolic Short-Term Memory
- Clustering
- LT Visual Memory
- Perception
- ST Visual Imagery
- Action

Decision Procedure

Body
CLARION

Action-centred
- action-centred explicit representation
- action-centred implicit representation

Non-action-centred
- non action-centred explicit representation
- non action-centred implicit representation

Motivational
- goal structure
- drives

Meta-cognitive
- reinforcement
- goal setting
- filtering, selection, regulation
Methodology

- **Input Data/Perception**
  - How data is captured from the environment, stored and further processed. Two approaches: symbolic and sub-symbolic.

- **Goals or motivations**
  - These are factors which distinguish motivational agents from reactive agents.

- **Action Selection**
  - How to choose the best action in the short and long term.

- **Learning**
  - Which are the learning mechanisms and how do they help the agent in order to take an action.
In **SOAR**, all information obtained from the environment is stored in working memory in the form of WME (Working Memory Elements). The SOAR’s input/output mechanism requires WMEs to be generated externally, so it is mainly a symbolic architecture.

In **LIDA**, the sensory information is captured and stored in the Sensory Memory in a raw state. When a relevant stimulus is identified by the codelets, some nodes in PAM increase their activation level. LIDA is said to have a mixed kind of representation, but more biased to a symbolic approach.

**CLARION** has a mixed representation. Each module is divided into two parts: the top-level is responsible for the explicit knowledge (symbolic, using rules for example) and bottom-level for implicit knowledge (subsymbolic, using neural network for example).
In **SOAR**, there is no built-in motivational process. In other words, this issue must be solved outside SOAR, if necessary.

In **LIDA**, motivations are considered, in an embedded way, in the Behavior Network (an action-selection module).

**CLARION** has an exclusive module, named MS (Motivational Subsystem), in charge of handling goals and motivations. The output of this module is used as input for the ACS (Action-Centered Subsystem), such that the decision process might be influenced by current agent’s motivations. The MS is also divided in top-level (symbolic) and bottom level (sub-symbolic).
Action Selection

- **In SOAR**, actions are associated with operators and operators are associated with state. Operators are represented by rules, which verify a set of conditions and apply changes in the current state. The selected operator will be chosen according to its priority compared to the others.

- **In LIDA**, there are basically two modules responsible for action selection: PM and AS. PM will provide the available actions at the current state and the AS will select just one action using a modified version Mae’s Behavior Net.

- **CLARION** uses a probabilistic method to select the proper action. The ACS is CLARION’s module responsible for choosing action. At the top level, there are basically rules. The bottom level is composed of neural networks. The selected action will be chosen according a probabilistic distribution between the two levels.
A classical learning procedure in SOAR is chunking, and its function is to create new procedural knowledge. The basic idea is to summarize previously learned knowledge, creating a new rule.

There is a reinforcement learning module which is one of most recent modules in the architecture. The idea is to determine a positive or negative reinforcement.

SOAR also has two types of memories subject to learning: episodic and declarative.

In LIDA, there are three types of learning: perceptual, episodic and procedural and they are related to the consciousness mechanism in architecture.
Perceptual learning is related to object recognition, new categorizations, and new relationships. Episodic learning is the encoding of information into episodic memory, the associative, content-addressable, memory for events. Finally, procedural learning also uses the information provided by consciousness broadcast, which is applied on the scheme net implemented in procedural memory.

CLARION provides a two-way learning system mainly in ACS and NACS modules: bottom-up and top-down, so new items at the top level are learned from items at the bottom level and vice versa.

In addition, learning in the same level is also possible, for example: in the bottom level, a backpropagation network with Q-Learning algorithms can be used in order to adapt the neural network using reinforcement learning.
SOAR is a predominantly symbolic architecture, despite having some sub-symbolic modules. LIDA and CLARION, on the contrary, have a mixed approach that tries to combine the benefits of both paradigms.

CLARION has a unique mechanism for dealing with goals and motivations, allowing an agent to select new actions bringing some benefit only in the long term.

All the architectures provide learning processes. Nevertheless, they are quite simple in SOAR, providing less flexibility than in CLARION or LIDA.

In a future work, we intend to focus on more pragmatic issues related to the software implementation of the architectures.