*Intelligent Systems Conference 2018*
*6-7 September 2018 | London, UK*

# A multi-tiers AI and IoT architecture

*Francesco Rago*
*Megatris Comp. LLC*
*Sunnyvale, CA 94085*
*Francesco.rago@megatris.info*

***Abstract*** **— New intelligent technology solutions are an enabling opportunity for innovation in the Internet of Things (IoT). These challenges call for more intelligent computing models (Digital Agent, Deep Learning, Semantic Networks, …) that enable rapid innovation for applications and service delivery. Big Data is a consequence of IoT applications as they are a major source of data. The Internet of Things delivers fast-moving data from sensors and devices around the world. The challenge for many organizations is making sense of all that data. Digital Agents can be used as a framework for modeling, understanding, and reasoning about them. In order to improve the efficiency of processing it is important to understand how these applications and the corresponding big data processing systems are performed in cloud computing environments. Therefore we have implemented a set of measures to improve the architecture. A real case study is described.**

***Keywords—IoT; AI; AI Architecture; Digital Agent; Neural network; Semantic network; Rules; AI Architecture measurement;***

## I. Introduction

In a world where almost anyone and anything can be connected to the Internet, the exponential increase in the volume of information and connected devices creates a dilemma. In these cases new intelligent technology solutions are an enabling opportunity for innovation in Internet of Things (IoT). These challenges call for more intelligent computing models (Digital Agent, Deep Learning, Semantic Networks, …) that permits rapid innovation for applications and service delivery.

IoT applications are considered to be a major source of big data and are supported through clouds architectures where data is stored and processed.

The critical challenge is using this data when it is still in motion, extracting valuable information from it. Organizations are scrambling to apply tools and analytics to these streams of data before the data is stored for post-event analysis because it is necessary to detect patterns and anomalies while they are occurring, in motion, in order to have a considerable impact on the event outcome.

## II. Our research platform

For real-time decision making on data, we have developed an architecture triggered by events that process and provide deep analysis on their data. Event processing uses the following techniques to manage, and make sense of streaming data:

1) Assessment, applying transformations and rules to determine if further processing needs to occur or the data (or event) can be quickly discarded.

2) Analysis, time series, analysis generated by an event stream processing, can be continuously processed to understand real-time trends.

3) Correlation, event stream processing allows to connect to multiple streams of data and identify that series of events occurred.

Digital Agents, (DA) Neural Networks (NN) and other AI Tools are vital for modeling, understanding and reasoning about data.

A Digital Agent can be considered as the universal primitives of digital computation [1] while NN are connectionist computing systems vaguely inspired by the biological neural networks. Such systems learn tasks by considering examples, generally without task-specific programming.

Finally, we have used software metrics to expand our standards with new ones to try to define a rationale for software process design. As well know, a software metric is a measure of a degree to which a software system or process possesses some property. The property we had essentially in mind was performance to improve algorithms and to create recipe to use the different AI tools. Therefore, AI measurements are highly demanded, but such work is still in its infancy in industrial environments.

## III. Research Contribution

This research has explored the use of maps onto maps inspired by neuroscience. As is well known nervous system is a set of modules of neurons that is able to ignite a many layers cognition. Even if we used supervised neural network we were able to create a full circuit that from the analysis of IoT data generate to an answer through cognitive steps.

## *IV. Related Works on Cognitive Architectures*

A cognitive architecture is a generic computational model to study systems behavior and cognition. It provides agents with decision-making mechanisms.

Among the most known cognitive architectures we have: SOAR [2]; CLARION [3] and ACT-R [4].

SOAR is the most known and includes working and long-term memory, and learning mechanisms (chunking, reinforced knowledge, etc.).

John E. Laird proposed a standard cognitive architecture to provide the appropriate computational abstraction for defining a standard model, although the standard model is not itself such an architecture [6]. The standard model spans key aspects of structure, processing, memory,content learning and perception and motor.

We started our work from Laird one, even if we have modified the standard depending on neuroscience approaches developed these years.

The purpose of architectural processing is to support bounded rationality, not optimality. System behavior is driven by sequential action selection via a cognitive cycle.

We think that complex behavior arises from a sequence of independent cognitive cycles that operate in their local context, but separate architectural modules for global optimization and planning are necessary creating more independent layers in a logic of maps of maps as in natural brains.

Declarative and procedural long-term memories are contained in neural networks and not in symbols structures and associated quantitative metadata.

Global control is provided by a cognitive map realized by a neural network and procedural long-term memory is composed of rule-like conditions and actions. Rules are necessary to produce control actions for IoT environment.

Learning updates cognitive maps for specific contexts and it occurs online and incrementally, as a side effect.

-Perception and Motor Control is the key of a working IoT architecture. We can have many different such perception modules, each with input from a different modality and its own buffer. This is prepared "ad hoc" for each smart object or mobile connected device. Motor control converts internal states into external actions.

Neuroscience suggests connessionistic models as basis for cognitive architecture. It considers processing as the dynamic and graded evolution of activities in a neural net module. Each unit's activation depends on the connection strengths and activity of its neighbors, according to the activation function.

The consequence of all these assumptions is an hybrid connectionist architectures. Elements of classical symbolic processing are included in neural nets, Wermter and Sun [7]. We realized a collection of neural net modules that share data coded in activation patterns following Miikkulainen [8]. We used the best of the two worlds as the hybrid models combines both symbolic and sub-symbolic approaches with the rising paradigm represented by connessionistic machine learning methods, such as deep learning, which have found enormous practical success in limited domains, Kotseruba [9].

## V. AI ARCHITECTURE FOR IOT

We are going to describe now our DA based architecture. It is devoted to manage complex environment and it is structure to give a cognitive taste to old fashioned Command, Control and Communication systems.

### *A. Complex Enviromnent management*

AI systems permit the transition from Command, Control and Communication systems to Mission Management Systems where there is a requirement for always more unmanned management.

We have the following transitions:

- The Human System Interface is no more a cockpit Aide but it becomes a decision aide permitting operator empowerment;
- Sensing has an evolution from events collection to a Dynamical Threshold Management with a choice of phenomena of interest;
- Monitoring & Diagnosis process uses Models leaving event-action schema for a more complete semantic of events.
- Decisions are simulation and model based; statistics support Proactive Decisions.

### *B. A summary of the software architecture*

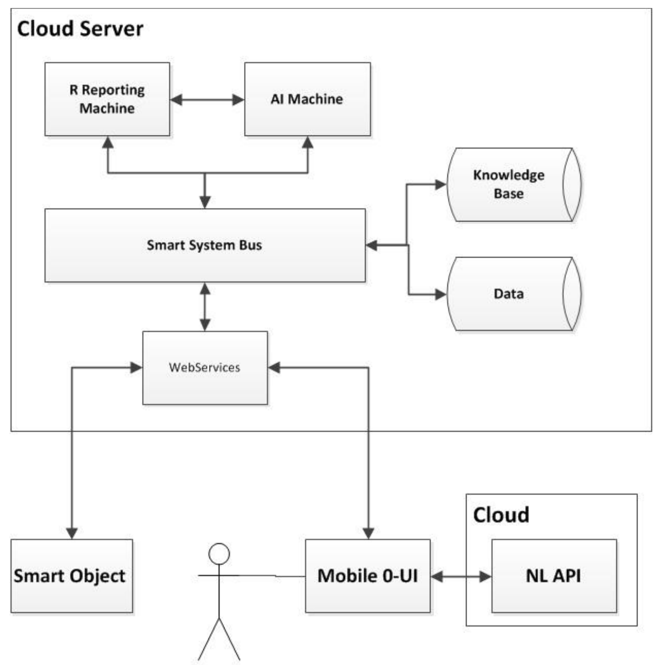


Fig. 1: Architecture flow

We give a rapid sketch of the architecture flow.

Smart objects and mobiles communicate with cloud servers using micro-services. Mobiles facilitate humans with an interface totally based onto Natural Language. The app user experience and user interface (UX/UI) is made to facilitate vocal and written communication.

Web services actuate requests to SSB (Smart System Bus). SSB drives the requests to the appropriate software cloud machines.

Modifying the Natural Language content is easy to associate cloud applications to a mobile or to any IoT device.

### C. Smart Systems Bus (SSB): Scope and Functions

SSB is a software system born in the new wave of IT Technologies related to Cloud Computing. It collects events from sensors and it is able to collect, cluster and support decisions, sometime in unmanned way.

The main functions of the SSB system are:

- Events management
- Communication management
- Actions management depending on Decision Models

SSB has architecture with one or more servers and controllers. Controller can be installed everywhere. Controllers can be any type of distributed hardware mobiles enclosed.

SSB can send feedback controls to the environment. It aggregates large numbers of sensors with a fusion of heterogeneous information from many sources. Many instrument resources have a continuous technological evolution and this characteristic causes continual re-invention of driver software. SSB middleware has as primary design goal to facilitate integration of instruments into current computing to leverage Cloud-based services.

## VI. AI Machine

An AI machine is based on Digital Agent as the universal primitives of digital computation [1]. All physically possible computation can be directly implemented using Digital Agents.

Message passing using types is the foundation of system communication. When a Digital Agent receives a message, it can concurrently:

- sends/receives messages to/from other Digital Agents;
- creates new Digital Agents to solve a problem;

Digital Agent Model can be used as a framework for modeling, understanding, and reasoning about, a wide range of concurrent systems.

Information integration needs to make use of the following information system principles:

- Persistence: Information is collected and indexed.
- Concurrency: Work proceeds interactively and concurrently, overlapping in time.
- Pluralism: Information is heterogeneous, overlapping and often inconsistent. There is no central arbiter of truth.
- Provenance: The provenance of information is carefully tracked and recorded.

### A. Digital Agents

Digital Agents are used as universal primitives of concurrent digital computation, have beliefs, desires and intentions. They have states which can be nested, so that, (for example), one agent is able to have beliefs about another agent's intentions. Agents communicate using asynchronous SSB standard messages architecture.

*a) Agent declarations and initialization*

Agents are declared via the agent keyword, followed by the agent's name (which must be unique), and the body of the agent. At startup, agents are invoked in the order in which they are declared; an agent terminates when it reaches the end of its code body or if terminated. Agents can communicate the results of their actions.

*b) Beliefs, desires and intentions*

Agents have explicit data structures corresponding to beliefs, desires, and intentions. These states can be nested, so that (for example) an agent can have beliefs about other agent's intentions.

Agent's beliefs are the information it has about contexts and about itself; these beliefs may be incorrect. An agent's intentions activate a course of action(s) as one's purpose or objective plan.

An agent's desires means a strong feeling of wanting to have something or wishing for something to happen but still it is not in an intention state.

The following is thus a legal modal expression:

believe AGE = 50 ;

intention (week = 1, heartrate = 80, … ,week = 12 ,

heartrate = 75);

*c) Context*

Context is a structure composed by entities and relationships. Such data can be of two types: unchangeable and changeable.

An agent is part of a context and can represent and analyze it as a dynamical systems. In general, the state of a dynamical system is a trace that summarizes all the information about the past behavior of the system.

Beliefs related to a context overload agent beliefs, for example:

On(table, floor); unchangeable

John(age, 15) ; changeable

*d) Communication*

A DA provides built-in communication primitives for sending and receiving messages. Their messages can be: pushed to mobile. They send orders and data, receive orders and data. Messages are event following SSB standard.

The effect of communication is to change the state of the recipient of the message or to activate actions. Note that message delivery is guaranteed, but it is asynchronous.

*e) Functions*

A DA can activate functions as precompiled or interpreted SSB routine. All functions have global scope, and can be invoked either by agents or by other functions.

*f) Learning*

A DA can learn from different sources: Smart Objects, Mobiles and internet. Data is appropriately used to extract meaning using neural networks or mapping data to a cognitive map. Neural network are the key tool to classify events a.

Neural networks can be applied to numerous situations where time series prediction is required. We can turn a temporal problem into a simple input output mapping by taking the time series data x(t) at k time-slices t, t–1, t–2, …, t–k+1 as the inputs, and the output is the prediction for x(t+1).

## VII. Agent's Cognitive architecture

Central to AI Machine is an extended semantic network. This is a knowledge base that contains a large number of entities of different type (class) and relationships between entities.

Each knowledge base is related to a context, an aggregate of entities, relationships and rules.

AI machine can be thought of as an architecture where there is no single algorithm that is responsible for intelligence. Rather, a large number of different specialized algorithms can be actives and these work closely together in cognitive synergy.

### *A. Computational Modelling for Digital Agent*

After the description of the basic characteristic of a Digital Agent, we will describe the internal structure of their architecture.

First of all each DA has multiple cognitive/perceptual layers with increasing degrees of abstractions.

We require that there be top-down and bottom-up mechanisms working together to connect the Cognitive representations to the perceptual data.

We require a Cognitive Layer (CL) and a Perceptual Layer (PL) with their own autonomous structures. The structure of CL reflects the Cognitive associations that we normally acquire through experience; and the structure of the perceptual layer reflects the historical series of input data. There is an autonomous memory for each layer where the associations or structures of the respective layers are stored.

### *B. Phenomenological Layer*

Phenomenological layer groups IoT data. The objective of PL is to organize data and to map to CL describing situations. During every system cycle new data are collected in PL.

*a) The agent's experiences in the context*

We assume that the continuous interaction of an agent and its environment is summarized by a discrete view-action-view sequence of the form:

$$v_0, a_0, v_1, a_1, ..., a_{n-1}, v_n. \qquad (1)$$

A view represents a sensory description associated with a context state.

An action denotes a sequence of one or more control laws [11] that take the agent from one state to the next. Distinctive states are the result of following control trajectory. The basin of attraction of the hill-climbing control laws absorbs accumulated error from each trajectory every time each action happens.

The sequence (1) is transformed into a set of schemas of the form $(v_i, s_i)$, $a_i$, $(v_{i+1}, s_{i+1})$, where $s_i$ is the state name associated with the environment state where view $v_i$ is observed. A schema represents a particular action execution of the agent in the environment. An action execution is characterized in terms of the distinctive states the agent was at before and after the action was performed.

If the states of the system to control obey to a SISO nonlinear system:

$$ds(t) / dt = f(s(t)) + Bu(t), \; y(t) = Cx(t) \qquad (2)$$

where s(t) is the state vector, y(t) is the system output e u(t) is the system input, we can achieve the desired control with a neural network. Using a recurrent neural network, we can construct a neural network system model:

$$ds(t) / dt = As(t) + WS(s(t)) + Bu(t), \; y(t) = Cs(t) \; (3)$$

where W is the connection matrix and S is the activation function [11].

### *C. Cognitive Layer*

Phenomenological states are subject to a transformation from Phenomelogical Entities into Cognitive Entities and Relationships. The CL has as main components a cognitive map which serves to acquire, code, store and recall about the relative locations and attributes of phenomena in their everyday knowledge environment. The term refers to a kind of neural network representing the agent knowledge.

Knowledge is updated whenever a relationship is found. The cognitive map updated by a neural network is a topological map, i.e., it represents only the connectivity between entities. Associations between entities can be represented by the variable $V_{i,j}$ , an association stored in the modifiable synapses of the neural network. Whereas a positive $V_{i,j}$ association means that knowledge entity j can be accessed from knowledge i. 0 means no relationship exists.

The prediction of neighboring place j, $p_j$, is permits to calculate activation $a_j$ by $a_j = p_i V_{i,j}$, and this activity indicates whether entity j is accessible from entity i.

### *D. Basic Flow*

During every system cycle new data is collected and transformed into Phenomelogical Entities, if appropriate. Data are contained in view and schemas structured depending on a template.

If we indicate $EAV_i$=[$element_i$,$attribute_i$,$value_i$] and $REL_i$=($element_i$, $relation\_name_k$, $element_j$ ) as elements of the views; we can execute rules:

$$\{EAV_i\}_{i\epsilon N} \text{ AND } \{REL_i\}_{i\epsilon N} \Rightarrow \{action_j\}_{j\epsilon N}$$

$$\text{AND } \{EAV_j\}_{j\epsilon N} \text{ AND } \{REL_j\}_{j\epsilon N}$$

and new states prepared using statistical tools of R machine, if appropriate.

There is a time cycle to transform Phenomelogical Entities into Cognitive Entities and Relationships. Neural networks recognize data status and define a possible update/evolution of CL.

The relation between schemas and contextual cognitive maps is recognized by a type of nonlinear mapping. The neural network model can be applied to many types of non-linear maps, if the pertinent variables were adjusted properly. This is a non-linear mapping between views whose data updates the specific contextual cognitive map while schemas are the input of cognitive maps to generate an output that will operate on PL.

If the CL is very dissimilar or juxtaposed after a predefined numbers of cycles (dissimilarity is measured on using a standard algorithm) this means a new concept is in the CL and problem solving results are evaluated using a fitness algorithm: more a logic element is able to fix problems, more it is fit. If it is not fit, new CL is changed with the old one.

## VIII. Architecture Evaluation

Following [12], a symbol system supports the acquisition, representation, storage, and manipulation of symbolic structures. Architecture is analogous to the hardware of a standard computer, while the symbols (which encode knowledge) correspond to software. The role of a general symbolic architecture is to support the encoding and use of diverse types of knowledge that are applicable to various goals and actions.

The basic functions performed by an architecture usually consist of the following (from Newell [13]):

- The fetch-execute cycle
- Assemble the operator and operands
- Apply the operator to the operands using architectural primitives
- Store results for later use
- Support access structures
- Input and output

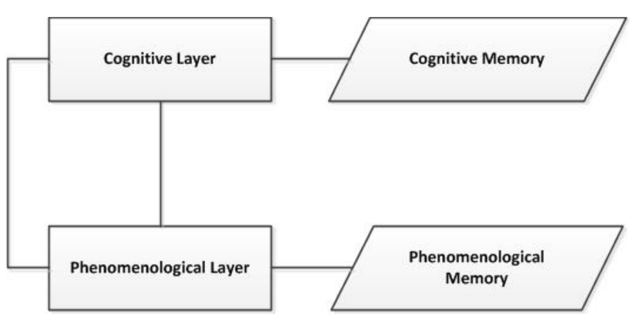


Fig. 2: DA internal data architecture.

Architectures are distinguished by their implementation of these functions, and the specific set of primitive operations supported.

We processed a problem of weather forecast with 65 DA executions on a ACPIx64 server using as database a MySql engine. We have the following average results:

Table I. measures on a case

| Activity | Unit duration (secs) | # | Duration (sec) |
|---|---|---|---|
| Execute cycle | | | 14 |
| Apply Operator | 0,24 | 20 | 4,8 |
| Rules | 0,32 | 24 | 7,68 |
| I/O | | | 1,52 |

A Digital Agent uses in average 14 seconds to terminates their execution. Operators containing neural networks activations and R routines use less time than the logical algorithm of rules. We have to balance rules that are easier to implement but slower in execution with the other operand that uses neural network.

## IX. Conclusion

This research has explored the use of maps onto maps on two layers. We trained supervised neural network because our assumption was that in nature the basic structure of a brain are coded in DNA. This is equivalent to create a structure using commands.

Metrics where used to value specific rules and operators performance. Next steps will be to create higher layer to permit a DA stream to improve cognitive levels of the systems.

## References

[1] Carl H, Bishop P, and Steiger R., "A Universal Modular Actor Formalism for Artificial Intelligence," IJCAI 1973.

[2] Lehman, J., et al.: A Gentle Introduction to SOAR, An Architecture for Human Cognition: 2006 Update. University of Michigan (2006) .

[3] Sun, R., "Desiderata for Cognitive Architectures," Philosophical Psychology 17(3), 341–373 (2004) .

[4] Anderson, J., et al.: An Integrated Theory of the Mind. Psychological Review 111(4), 1036–1060 (2004)

[5] Sun, R., "Cognition and Multiagent Interaction, From Cognitive Modeling to Social Simulation." In: Sun, R. (ed.) Rensselaer Polytechnic Institute, Cambridge U. Press, Cambridge (2005)

[6] Laird, J. E., Lebiere, C. & Rosenbloom, P. S. (2017). A Standard Model for the Mind: Toward a Common Computational Framework across.

[7] Wermter, S. and Sun, R. eds., 2000, *Hybrid Neural Symbolic Integration*, Berlin, Springer Verlag.

[8] Miikkulainen, R. and Dyer, M., 1991, "Natural Language Processing With Modular PDP Networks and Distributed Lexicon," *Cognitive Science*, 15: 343–399.

[9] I. Kotseruba, J. Tsotsos, "40 Years of Cognitive Architectures Core Cognitive Abilities and Practical Applications," 2016/10/27,arXiv preprint arXiv:1610.08602.

[10] P. Tabuada," Symbolic Control of Linear Systems Based onSymbolic Subsystems", IEEE Trans. On Automatic Control, Vol. 51, No. 6, JUNE 2006 1003.

[11] Y. Zhang ,"Adaptive Neural Network Based Control of Noncanonical Nonlinear Systems", IEEE Transactions on Neural Nnetwork Volume: 27 Issue: 9

[12] Wallace S.A., Laird J.E, “Toward a Methodology for AI Architecture Evaluation: Comparing Soar and CLIPS,”. In: Jennings N.R., Lespérance Y. (eds) Intelligent Agents VI. Agent Theories, Architectures, and Languages. ATAL 1999. Lecture Notes in Computer Science, vol 1757. Springer, Berlin, Heidelberg

[13] A . Newell, Unified Theories of Cognition, Harvard Press, Boston, MA. Shapiro, D., & Langley, P. (1990).

[14] Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics, AI Magazine 38(4). Robin Milner. Processes: A Mathematical Model of Computing Agents in Logic Colloquium 1973.

[15] M. E. Pollack, M. Ringuette. “Introducing the tileworld: Experimentally evaluating agent architectures”, In Proceedings of the Eighth National Conference on Artificial Intelligence, volume 1, pages 183–189. MIT Press, 1990.

[16] E. Gat. “Integrating planning and reacting in a heterogeneous asynchronous architecture for mobile robots”, In Proceedings Tenth National Conference on Artificial Intelligence, pages 809–15. AAAI Press, 1992.

[17] Jaeho Lee and Suk I. Yoo,”Reactive-system approaches to agent architectures”, in N.R. Jennings and Y. Lesp, Intelligent Agents VI, Proceedings of the Sixth International Workshop on Agent Theories, Architectures, and Languages (ATAL-99.

[18] A. Newell. Unified Theories of Cognition. Harvard University Press, Cambridge, MA, 1990. 11.

[19] Anderson, J.R., and Lebiere, C.,The Atomic Components of Thought, Lawrence Erlbaum Associates

[20] P. Langley,J.E. Laird, Cognitive Architectures: Research Issues and Challenges, (Technical Report), Institute for the Study of Learning and Expertise, Palo Alto, CA

[21] Controlling Physical Agents Through Reactive Logic Programming, Proceedingsof the Third international Conference on Autonomous Agents.