Architectural Factors for Intelligence in Autonomous Systems

Ignacio López and **Ricardo Sanz** and **Carlos Hernández**
Autonomous Systems Laboratory
Universidad Politécnica de Madrid
José Gutiérrez Abascal, 2
28006 Madrid, Spain

Abstract

The question of measuring intelligence in artifacts and, in particular, evaluating architectures for it, requires not just an understanding of the very nature of intelligence —quite elusive objective, indeed— but an appropriate stance for evaluation. In this paper we argue that it is not just the case that architectures provide intelligence, but that they really provide a substrate for intelligent behavior in the execution of a particular task. The *measuring-intelligence-for-autonomy* position becomes maximally relevant in the context of the increased uncertainty levels that the upcoming challenging applications are posing to cognitive architectures. This fits our understanding of intelligence as the capability of *maximizing information utility*.

The Meaning of Intelligence

The meaning of the concept of intelligence is a long-discussed topic, and we can safely say that a universal agreement has not yet been achieved. However, a short, selected collection can be found in (Meystel 2000). We can quote one of the more complete and practical, by James Albus:

> Intelligence – an ability of a system to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioral subgoals that support the system's ultimate goal.

It is important to observe that intelligence is conceived within the context of a system in an environment, rather than as a context-free characteristic. Also, system goals and uncertainty are involved.

The rest of the definitions in this same document included highlights on other aspects in relation with intelligence: ability to solve new problems (new to the system), acquire knowledge, to think and reason, to learn and to adapt effectively to the environment, either by the system transforming itself or the environment. We can see that these aspects — developed perception, knowledge, adaptivity, *etc.*— could be factors leading to the "appropriate action increasing the probability of success," introduced in Albus' definition.

An Overview of the Operation of a General, Autonomous System

From the interpretations of intelligence mentioned before, we may conclude that intelligence cannot be conceived out of the context of a system with certain degree of autonomy. Intelligence manifests itself as cognitive autonomy, *i.e.* the capability of a system able to *think by itself* to reach a goal. That lead us to ponder a perhaps more general question: How does an autonomous system work in the general case?

Trying to abstract the concept of autonomous system from any specific implementation, we may find that an autonomous system is, in the most general case, a parallel, distributed system. This means that the operation of the system may be composed of multiple simpler operations, being carried out in physically different parts. These parts show some kind of cohesion which justifies them being part of a unique whole.

In relation to Albus' definition quoted above, we may assume this cohesion to be the foundation for achieving a unified "system's ultimate goal." Each of the system's elementary parts would operate towards a subgoal.

The function of a part of the system is to achieve the corresponding subgoal. For this, each function will carry out more or less developed afferent, efferent and deliberative operations, suitably coordinated between themselves, and with the rest of the parts of the system.

Operations manifest themselves in changes in system resources. The resources of the system allocated to a certain function change their state accordingly. In many cases, these resources are part of the physical substrate on which the system is grounded.

Conceptual and Grounded Operation

We may assume that, in the most general case, a cognitive autonomous system operation can be analysed[1] at two levels. The first, which we may call *physical*, answers to physical laws: gravity, magnetism, *etc.* Indeed, an important part of the system's operation is its physical action on the environment; for example a robot picking up objects, or a mobile robot exploring new territory. This kind of operation can be

[1] It can also be realised at two implementation levels, but this is not necessary for the analysis that follows.

observed by measuring a certain amount of *variables*, representing speed, temperature, force, *etc.* We shall call this kind of variables *physical variables*, after (Klir 1969).

The other kind of operation in a general autonomous system is *conceptual*. A *conceptual variable* is a specific resource of the system whose state represents the state of a different part of the universe (Klir 1969). For example, the area of memory used in an image file may represent a landscape, *encoded* (Newell 1990) in the state of its own bits.

Indeed, an autonomous system may have the capacity of operating with conceptual variables, using them for representing objects in their environment, for simulating the effect of its own action over them, or for inferring new objects among other examples. This type of operation can be called *conceptual operation*. We may observe that it will always be grounded on the *physical operation* of its resources (Landauer 1992).

Organisation of a General Autonomous System

The previous reflections are sufficiently general to apply to most systems. They identify key points of their operation from two points of view: their dynamic functional organization and the nature of their operation. However, this dynamic portraits should be complemented with a static sketch of what a system *is* and of the nature of the parts of which its formed.

We may part from a basic fact about systems: the behaviour of a system is derived from its properties. Indeed, a loaded beam deforms more, less, elastically or plastically —or breaks— depending on its properties of elasticity. The behaviour of a complex system, as many autonomous systems may become, may derive from large, heterogeneous sets of properties.

If we would experiment with such a system in order to analyze its behaviour, we would eventually distinguish three kinds (Klir 1969). A first type of behaviour would be composed by patterns of action exhibited by the system for short intervals of time, maybe shorter than our observations. We shall call this *temporary behaviour*.

We would probably observe that some patterns of actions exhibited by the system always hold. That is, they would have been observed in all experiments. They constitute the *permanent behaviour* of the system. However, there will also exist some patterns of action which would hold for long periods of time, but eventually change, perhaps in answer to significant changes in the environmental conditions of the experiment. These constitute the *relatively permanent behaviour*.

Returning to the concept of system properties introduced above, we may say that the three kinds of behaviour derive from three categories of properties. Temporary behaviour will derive from a set of properties called *program*. Permanent behaviour from *real structure*, and relatively permanent behaviour from *hypothetic structure*[2].

[2]We use the term hypotetic stucture following Klir; another posibility was to call it *dynamic stucture*

As we may observe, the most variable properties of a system form its program, and the most static ones its structure. The architecture of a system is the definition of its properties; in other words, the definition of its structure and program.

Setting apart for a moment the question of intelligence, we may observe that the architecture of a system may strongly affect its level of autonomy. Designing a system with larger program will enable it to operate in fast–evolving environments. A highly-structural system could not be able to adapt to the environment fast enough. On the other side, a system with a large structure would be less adaptive, but easier to model and control to ensure cohesion and coherent behaviour.

Intelligence in a General, Autonomous System

Most definitions in (Meystel 2000), address three aspects normally associated with intelligence and intelligent behaviour: novelty in the state of the environment or in the problem to be solved by the system, uncertainty regarding what is going to happen, and dealing with trying situations. Generally, novelty and 'trying situations' may be be indistinguishable to the system.

Novelty and trying situations can be regarded as two independent forces against successful system action which, if overcome, suggest intelligence. We call them 'independent' because they cannot be controlled directly by the system. We shall now see how they affect the system and their relation with architecture and intelligence.

Bearing the previous operational and architectural portraits of general autonomous intelligent systems in mind, we can observe that a novel or trying situation would put the system in a scenario to which its properties would not be adjusted to.

This mismatch can influence the system causing parts of it to change. Normally, the most variable part, its program, will change first. Then, the hypothetic structure. If the mismatch would be sufficiently intense, the real structure of the system might be affected and degraded. Note that the real structure of a system refers to the intrinsic properties of a system, including its *ultimate goal*, mentioned in the first definition of intelligence. Changes in the real structure stand for loss of system cohesion.

If the system is sufficiently autonomous, this progression of changes induced by the state of the environment will be stopped before reaching the real structure, and compensated for. The system will either adapt its hypothetic structure and program to the new environment, or it will change the environment (or both.) This may imply the need of reflexive intelligence on some part of the system.

When confronted to a novel or trying situation, a system will not be able, in general, to remain unaffected. However, the influence of the environment should be minimized by the designer, or confined to the most temporary of its properties. Architecturally, the system must be prepared to support a certain amount of change induced by the environment, and also be prepared to be able to reconfigure and adapt, without perturbing its real structure. This implies that an autonomous system must have a necessary degree of program and hypothetical structure. Robust but static alternatives may exist but they are difficult to find in the general case (consider, for example, the case of H_∞ control).

We may observe that the direct influence of the environment on a system comes from the coupling of physical variables of the system with physical variables of the environment[3]. Indeed, environmental temperature may affect the operating temperature of a computer; gravity may affect the works of a mechanical device. Systems are not perfectly isolated from their environment. This influence cannot be eliminated, although adequate system design can sometimes damp it and orient it advantageously.

Of course, the capacity of an intelligent system to overcome a novel or trying situation does not only lie in its independence from the environment. There also exists a factor of capacity: action *power*. Changing an unfavourable environment may involve physical action, which requires a certain amount of power. It is not only a matter of quantity. Also, the system must have the capacity to realize those changes, in the sense of having the adequate actuators, or resources in general.

This factor of capacity also applies to conceptual operation. The system must have the capacity to understand the situation, decide a course of action, and execute the specific actions. The system must know how to adapt itself appropriately to increase its probability of success. Adequate resources without knowledge would prove useless.

The capacity to understand and learn are explictly mentioned in the definitions in (Meystel 2000). We would like to briefly discuss a problem related to two other points included: 'adapting effectively to the environment' and 'apply knowledge.'

As derives from the previous sections, knowledge, deliberative processes and many aspects of perception in systems are examples of conceptual variables and conceptual operation. Physical action must correspond to conceptual processes. Executing a plan (conceptual) generated by a planning algorithm (conceptual process) means that the system executes physical action corresponding to it.

We shall call the process of making physical variables in the system correspond to their conceptual counterparts *grounding*. In a system formed by a controller and a DC motor, for example, the system may plan to make the input voltage to the motor 3V. The plan could be represented by a conceptual variable ̀input voltage' set to the value of '3V.' The variable would be grounded when the actual input voltage (physical variable) is set to 3V. Achieving this correspondence, however, may not be straightforward in real cases. In fact, it will frequently happen that grounding a conceptual variable precisely is not possible. As if, in the previous example, the input voltage could not be risen to more than 2.8V for any reason (insufficient power, perturbances, *etc.*)

The causes for this can be many. We may mention some. First, we must take into account that conceptual variables are normally implemented in specially dedicated resources, whose physical constraints are deliberately few: hard disks and RAM memory modules for example. This means that the environmental conditions can rarely prevent them from being set to a particular state: for example storing any combination of 1s and 0s. Their representational power is therefore little constrained by physical laws. However, making a physical variable adopt any value represented in its conceptual counterpart may imply impossible amounts of power, and be subject to noise from multiple sources.

A second cause may be the actual nature of conceptual variables. A conceptual variable may easily refer to abstract concepts and objects. In fact, a problem solving algorithm may produce an abstract solution which has to be translated into a set of specific, physical values to be implemented. Usually, the equivalence between the abstract solution and the physical counterpart is not unique or direct. Establishing a specific method for grounding an abstract object may involve making arbitrary intermediate decisions to solve ambiguities. The efficiency trade-offs of possible alternatives may not be easily established.

In summary, *grounding* is not straightforward for many reasons[4]. Inadequate grounding may lead to unforeseen and undesired behaviour. Proper grounding may sometimes be impossible due to practical reasons. A meta-knowledge is therefore necessary: knowledge on how to apply knowledge.

Observable Degrees of Intelligence

After these considerations, we can reflect on the original definition of intelligence by James Albus, perhaps being the most demanding: 'to act appropriately in an uncertain environment.' Acting appropriately involves perceiving, thinking, and conveying the thought appropriately to the environment. As we have seen, this involves physical, conceptual and grounding capacities, apart from a degree of architectural suitability.

From the point of view of the artificiality (Simon 1990) "acting appropriately" means serve a purpose of the maker, *i.e.* provide some degree of utility. In the case of intelligent systems this is done exploiting information stored in and/or obtained from the agent's environment. We can see different degrees of intelligence in the varying forms that this information is used to shape agent's behavior —temporary, permanent or relatively permanent— to fulfill the objectives: changing structures, groundings, *etc.*

In a very deep sense, the different observable degrees of intelligence —*i.e.* how information may affect behavior— may not be conmensurable with each other. This is a well known problem with human intelligence measures like Binet's test. The different aspects of intelligence (Gardner

[3]This need not happen at the conceptual level and for some researchers this lack of *situatedness* is a major impediment to real intelligence.

[4]The relation conceptual-physical is just an instance of a general class of *realisation* relations, for example over virtual machines. The main difference regarding physical realisation is the practical impossibility of environmental decoupling of the physical level in contrast with the conceptual one (this decoupling is indeed not desired if the system purpose is to somewhat change the environment).

1985) will require different kinds of measures unless we're able to find a raw measure that is common to all them (a kind of "conceptual´´ power as sought in the research on general intelligence). The efforts in this line have always somewhat reverted to measures of information —in Shannon's of Fisher's sense— not being very useful as evaluations of architectural aspects. This problem is not new and is the same problem found in the evaluation of computing power: megahertzs or megabytes do not convey the relevant information in most cases. That's the reason for focusing on benchmarks, *i.e.* external measures more than internal ones.

This implies than in a practical sense, intelligence can only be measured externally in the interacion of the agent with a surounding world to fuulfill a particular task (this, indeed, is Albus' *appropriate action*). We like to use the more sucint definition of intelligence: *intelligence is the capability of maximising information utility*.

Evaluating Architectures for Intelligence: Questions

Which functions/characteristics turn an architecture into an architecture supporting intelligence?

Let's describe four aspects that are of major relevance for architectures supporting intelligence as needed by an autonomous system:

- Scale and scalability; both conceptual, physical and even temporal. As a general rule, larger resources imply larger power, provided adequate control and integration. While physical scalability might not be always possible or adviceable, conceptual scalability, understood as ability to grow, enables growth in afferent and deliberative capacities (including meta-knowledge).
- Restricted dependence from the environment. As we have seen, this damps the effect of the environment on the system, which on one hand makes the system less vulnerable, and on the other enhances the possible action of the system over itself (adaptivity.)
- Adequate degrees of structures —real and hypothetical— and program. Excessive real structure prevents adaptivity, excessive program favours dependence from the environment. Hypothetic structure provides global adaptation to operating conditions.
- Conceptual-Grounded coherence. As we have seen, grounding is a key factor for effective action. This depends on a certain degree of compatibility between physical and conceptual variables and processes.

What evaluation methods are needed for different types of cognitive architectures?

Neutral —*i.e.* architecture independent— evaluation methods will mostly be in the form of benchmarks if they are expected to provide some degree of information about the performance of a particular system in a particular task.

More analytic measures will depend of the formalization of specific aspects of the architectures (e.g. real structure resilience or program adaptivity to certain class of environmental changes and disturbances).

How can we determine what architectures to use for different tasks or environments? Are there any trade-offs involved?

(Sanz, Matía, & Galán 2000) develops the idea that system, task and environment are coupled so that it is impossible to define system autonomy without considering all three factors. (Hui-Min Huang 2004) develops this idea in the specific area of unmanned systems. In this text autonomy is measured by a three–dimensional vector of the following components: environmental difficulty, mission complexity and human interface.

These two approaches show a unified conceptualization of system, task and environment. Developing adequate metrics, determining two of the three factors would enable calculating the third. At the moment, this can only be achieved in restricted domains as considered in (Hui-Min Huang 2004), and only after assuming a certain number of hypotheses for simplification.

Some of the systemic notions proposed here enable to obtain some direct conclusions about the relation architecture-environment, as it has been briefly discussed above.

Trade-offs derive from the fact that greater power and larger resources, a positive factor, derives in higher complexity and more difficult system cohesion. Traditionally, uncertainty and environmental dependence has been damped by designing controlled environments for the systems to operate in. The less controlled the environment, the wider the uncertainty: speed of evolution, events, *etc.*

Concluding Remarks

Intelligent system architectures have been designed with very different purposes in mind and this has led to the practical impossibility of architecture comparison across application domains. From a general perspective, however, there are two possible strategies for intelligent system architecture evaluation:

Benchmarking: It will provide performance information just about specific tasks and has the added inconvenience of requiring an extant cognitive system to be able to evaluate.

Formalisation: Formalising core mental properties will render neutral, domain independent measures that do not requiere extant systems, *i.e.* may be used in analysis and design phases.

This last strategy seems the most desirable but has a major drawback: formalization is always hard and at the end it may finish in so fine grained concepts and associated measures that they would be mostly worthless for design.

Acknowledgements

The authors would like to acknowledge the support of the Spanish Ministry of Education and Science through the DPI C3 grant and the support of the European Commission through the IST ICEA grant.

References

Gardner, H. 1985. *Frames of Mind: The Theory of Multiple Intelligences*. Basic Books.

Hui-Min Huang, E. 2004. Autonomy levels for unmanned systems (alfus) framework. volume i: Terminology. version 1.1. Technical Report NIST Special Publication 1011, National Institute for Standards and Technology (NIST).

Klir, G. J. 1969. *An Approach to General Systems Theory*. Van Nostrand Reinhold.

Landauer, R. 1992. Information is physical. In *Workshop on Physics and Computation, PhysComp '92.*, 1–4.

Meystel, A. 2000. Measuring performance of systems with autonomy: Metrics for intelligence of constructed systems. White Paper for the Workshop on Performance Metrics for Intelligent Systems. NIST, Gaithesburg, Maryland, August 14-16, 2000.

Newell, A. 1990. *Unified Theories of Cognition*. Harvard University Press.

Sanz, R.; Matía, F.; and Galán, S. 2000. Fridges, elephants, and the meaning of autonomy and intelligence. In *Proceedings of the 15th IEEE lnternational Symposium on Intelligent Control (ISIC 2000)*, 217–222. IEEE. http://ieeexplore.ieee.org/iel5/7088/19098/00882926.pdf.

Simon, H. A. 1990. *The Sciences of the Artificial*. MIT Press.