Proceedings of the Seventh International Conference on Cognitive Modeling

Trieste, April 5-8, 2006

Edited by:

Danilo Fum

Fabio Del Missier

Andrea Stocco

Edizioni Goliardiche

Edizioni Goliardiche

sede legale
Via SS. Martiri, 18 - 34124 TRIESTE

sede editoriale
Via Aquileia, 64/A - 33050 Bagnaria Arsa (UD)
tel. +39 0432 99.63.32 - fax +39 0432 99.69.00
email: eg.books@libero.it

uffici editoriali periferici
Campobasso • Francavilla al Mare (CH) • Macerata
Palermo • Parma • Sora (FR) • Urbino • Vercelli

ISBN 88-7873-031-9

finito di stampare nel mese di marzo 2006
presso le officinegrafiche **estanmoda** - bagnaria arsa _ud_ italia
tel. +39 0432 99.63.32 - fax +39 0432 99.69.00

Cognitive modeling of computational and communications systems based on physics.

Carlos J. Martinez (jcmartinez@die.upm.es)
Dept. Electronic Engineering – ETSIT - Polytechnic University,
Paraninfo Ciudad Universitaria - Madrid 28046, Spain

Introduction

The network and computing evolution towards new high performance and processing paradigms is accompanied by new tools to model the new complex designs. Works from specification to implementation follow those needs.

In this line of work new methods for representing and modeling complex systems, we propose a physics-based model for cognitive and new computing and distributed paradigms. We present a formalization for a physics-based idea – originally used for modeling a mobile networks - where we treat data as the most "intelligent" parts and the classically though as intelligent processes as mere "collisions" of particles.

Among the wide variety of methodologies for modeling systems, from UML or Petri nets, to Markov Chains or Turing machines, we find that basing our representation in nature ended up in giving us several benefits. For instance, we found that comparing real systems to nature allows grasping the global picture quickly, sometimes the view let us pointing out systems parts that could be improved. The use for cognitive models and tools is proposed here; a close look to possible applications in GOMS, ACT-R, or SOAR will be the subject of future extended presentations.

An advantage of particles was the possibility to bring nature side-effects back to our design, converting them in new features or innovations to the system or introducing metaphor-based parts to complete the original system.

Initial experiences and tools

The particles idea for specifying and building protocol state-machines originated in late 2000 meanwhile planning performance evaluations of telephone IP based networks. At that time test tools required a programming effort in order to arrive to generate 21 calls per second towards a telecom 3G switch. By early 2003, we were using the particle model and a "particle engine" interpreter calling each phone in the testbed. Against the low expectations that an interpreted engine could suggest, we found that just an average server was able to generate 100 calls per second using a particles-view design and implementation.

The main program was using collisions - interactions or reactions - among particles as model code. It did not require a great depth of physics knowledge and using particles such as billiards balls was perfectly suitable metaphor for design.

Intuitive definition of the particles model

We introduce briefly the "particles" model, able to represent and implement systems based on physics. Given the complexity of cognitive systems, we looked at the most elemental and smallest systems in nature, and that led us to choose particle physics as our reference to model complex behaviors and systems. Looking at nature we asked several questions that served to give birth to this proposal:

1. Are actors more important than the information, really? Contrary to our macroscopic daily experience where a process has more information, and seems to be more "intelligent" than the information, we propose to treat the information as the model core instead of "mere" processes.

2. Can an effect be the consequence of only one cause? We think that statement as partially-true and consider that all effects are produced by joining two, or more, causes. The causes are in our approach the coincidence in space-time of two or more classically called elemental causes.

If we look at nature believing the messages –not the processors or brains - are the main "actors" and need to join two "causes" to produce an effect, we will be able to recognize these behaviors. It does not imply the postulates are absolutely true, but just a consistent view of reality.

A simple example

Let first see an example of particle interaction and relate it to a traditional system models. Consider the collision of an electron and a proton to produce a neutron and a photon (figure 1.a). The interaction between two particles in this way can serve for modeling a process with two inputs signals, and the resulting particles will correspond to two output signals (figure 1.b). Having found advantages using this kind of elemental particle physics we also experimented with bigger elements like molecules in chemical reactions.

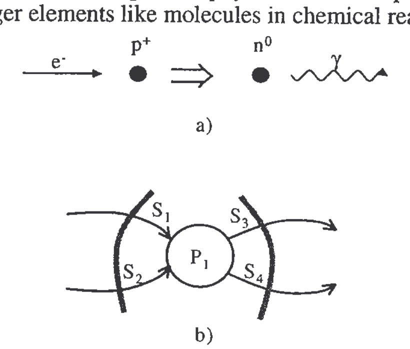

Figure 1.

The arriving events of signals s1 and s2 correspond to the electron and proton of figure 1.a, and the process P1 is related to its physic collision. This intuitive correspondence between particles and computing elements is formalized in the next section, but let us see other analogies below:

Potential use and meaning

Notice that figure 1 allows a radically different view of networks too, with messages (s1 s2) as the real actors and routers (p1) as simple colliding zones. Also computers can be seen as having intelligence not in the processes but in the signals or data. These behaviors seem appropriate for new paradigms we propose extending it to cognitive systems.

In cognitive models, the electron and proton could be knowledge - believes, states, goals... - that collide in order to produce new knowledge – neutron photon, actions, new believes, states...- A production rule will be mapped to just the collision zone and basic physics law of particle interactions. At the long run this could make cognitive systems as simply the coincidence in time of some knowledge like those electron and proton.

Applied to adaptable dynamic agents - able to reconfigure and react to environment – we see S2 (proton) as the agent implementation and S1(electron) as a message. Here P1 is just a collision in the particle-view, where in other models might seem P1 is a static agent itself, and S3 (neutron) can represent a new-agent with an "adapted" behavior.

We also found benefits in bringing particles side-effects back to cognitive models, for instance Sims and Gray (2004) showed a memory decay formula that can be approximated by particles. In nature massive particles use to "decay" in time to smaller - less energetic – particles, and we can store memory information in particles knowing that they "naturally" lose information and that it occurs as an stochastic process with an exponential like curve.

The particles view

We consider that our model states that the importance of a system is in the messages more than in the traditional processors. The application of this first postulate broke with the idea of intelligent engines exchanging simple data, and left us with "intelligent" particle/ knowledge being the real actors; they follow "nature rules" and collide in simple zones. Advantages for traditional and cognitive systems are:

- In networks, the model reduces the routers to junctions and gives all the intelligence to messages and packets. The routers/ processes are simple areas where the incoming particles/ messages collide. The behavior is now led by messages and rules, carrying the intelligence of the network.

- If the particles are the main actors they have more power on the overall behavior of collision zones, and therefore as there are no main processors, the system can potentially modify itself and adapt.

We want to build systems by treating the data as cognitive carriers and process as nothing else than space-time zones or collision areas. Other benefits and possibilities are:

We can use Particle-Dynamics applied in traditional models, ranging from data-flow to state-machines.

- It is applicable to many different levels of abstraction, from the visual tool representations in CaDaDis or Cogent tools to a pure definition of behaviors where the "zones" represent totally abstract concepts.

Formalizations and foundations

In order to provide a robust formalization and show how the Particle-Dynamics model can be applied to a great number of systems and cases, we worked to make the Particle-Dynamics match with the meta-model defined by Lee and Sangiovanni (1998). Since that meta-model was valid for the computational models ranging from discrete-time to Petri-nets, it shows the particle model is valid for many computational systems but a full demonstration is left for extended papers. Here we focus on introducing the notions of "Particles" and "Zones" briefly.

A *Particle Observation*, or measurement, e has a tag and a value, i.e. it is a member of the T x V set where T and V are the sets of all possible tags and values. The tags in T allows to model time, precedence relationships, and also other key properties when we add dimensions like space-time or object-layer in this Particle-Dynamic view. This e is equivalent to what the computational reference framework defines as an event.

A particle *Trajectory* is the set of space-time coordinates – not the observed values - along its life history. This life trajectory s can thus be defined as a projection over the T axis of a set of observations. It is the set of representations of a particle at different times and locations (since we added the space dimension). This corresponds to the T(s) defined in the framework denoting the set of tags in a signal s.

About *zones*, those will be defined in terms of tuples or vector of N paths, each of those vectors b belongs to the set of tuples S x S x... x S that represent the behavior of a process. We define a zone z to be an index of any subnet in the set B of behaviors P_1 P_2 ... P_M ... . Let us notice that among the set of all possible behaviors we could have as unique element the process representing the entire system. In Particle-view we represent a bigger zone that can have smaller zones and particle details inside it.

Acknowledgments

The first work on particles was funded by Interquanta.com in its project for next generation mobile during 2003-2004.

References

Lee E., Sangiovanni A. (1998) *Comparing Models of Computation*, IEEE Trans. on Computer-Aided Design Of Integrated Circuits And Systems,

Martinez C., Estebanez L.,(2006) *Modeling Nature-Inspired Systems based on physics*. Proceedings of the IASB

Bhattacharyya S., Murthy P., Lee E., (1999) *Synthesis Of Embedded Software from Synchronous Dataflow specifications*, Journal VLSI Signal Processing

Sims C., Gray W, (2004) Episodic versus Semantic Memory: An Exploration of Models of Memory Decay in the Serial Attention Paradigm.