Globes, AEDL, and a Particle-Oriented Model

Testing ways to use AEDL concepts as a universal or standard way to represent other simulations.

The Particle-Oriented (P-O) model provides a mechanism for mapping concepts between the world of physics and computer science, being based in particle-like objects (review in panel#2).

**Ex. Defining and Setting Oscillation parameters with model-particles.**

In this example a typical Globes code is seen with the data and functions modeled as particles and its reactions-zones respectively:

- The `glbAllocParams` process of requesting a part of memory modelled as the collision of a request particle with the memory (composite of particles).

- The allocation process (with access a memory of complex mem particle composite) let escape the variable “values” that will hold the params.

- The particle “values” continues moving and passes through the zone called “glbDefineParams” where it collides with particles “th12” … copying its info.

**Mapping between Globes AEDL and A Particle-Oriented modeling**

Martinez, UPM Madrid University & supported by [http://Intequanta.biz](http://Intequanta.biz)

GLOBES code:

```c
values = glbAllocParams();
```

```c
glbDefineParams(values, th12, th13, th23, delta, sdm, ldm);
```

```c
... glbFreeParams(values); 
```

Backwards mapping: particles interactions into computer processes

**Information Systems as Particles Reactions**

- The first computing models using particles collisions are due to E. Fredkin & T. Toffoli in 1981 [FT1] for reversible gates, followed by “The Chemical Reactor” works at MIT [CR2].
- We focus in the mathematical model for generic computing, as it shows an Equivalence between Information-Particles Objects (P-O) and classical data, reviewed here [1][2]:
  - Model anything (from Finite State Machines (FSM) to Petri-nets (PN) with particle reactions
  - Applicable from design to operations as if the systems were composed of modeling particles.
  - The Maths show the bidirectional mapping between the P-O model and the classical models [1]
- Why using particles instead of others (Object Oriented, State Machines, …)?
  - It introduced Space-Time coordinates into processes and data, aiming to help visualization and also to being able to leverage modeling-geometry and symmetries in order to simplify designs.
  - It maps to UML, also visual, but P-O allows to use of geometry and even create model-conservation laws to simplify designs (as momentum-conservation help calculus in physics).
  - A “particle reactor”, as a compiler or interpreter, handles communicating messages, data parsing & composition, statistical multiplexing,… allowing time to designers for focusing in key functionalities.

The modelling particles can be seen as existing in a many-fold space or as a composite of onedimensional modelling particles, each one representing an axis’ component.

Mathematically 1 particle in a N dimension many-fold is equivalent to 1 object which is the composition of N fundamental particles. A P-O reactor should be able to work with both extremes as well as intermediate views:

---

**PARTICLE MODEL**

- **Particle Observation, or measurement,**
  - Extends what the computational reference framework defines as an event,
  - Corresponds to messages, goes beyond a signal and contains all messages fields.

- **An observed history** is the set of observations of particle values at different times
  - equivalently to a signal or a flow of messages, or conversations between nodes

- **A particle Trajectory or path** is the set of space-time coordinates
  - It corresponds to timestamps in computing and networks, – not values - along its life history,
  - extended to allow the use of geo coordinates

- **A zone,** defined in terms of collisions of particles,
  - Equivalent to process in computing
  - in networks, it is equivalent to nodes, and some special cases of links, networks and subnets

---

--[FT1] Conservative Logic, E. Fredkin, T. Toffoli, 1981,
Mapping between AEDL and Particle-Objects

In P-O [ND3] the model-particles represented literally the QM maths and its conversions between states in terms of probabilities to detect the amplitudes of each component. The 3 flavours and mass states were mapped to modelling-particles in a Hilbert space.

Learning from Globes AEDL, the next step seems to add also in P-O the equivalent of channels and rules, as more practical concept.

In representation and simulations, a goal is to define all reactions in a rule-like language (x y -> u v) using P-O maths for generating code. At this point only the graph trees are automatically generated as in this example of MINOS.glb file [MN4] in the GLOBES site:

Looking at easing up the efforts on experiment simulation design and communication, the proposal is to display the AEDL representation of experiments in a graphical way: the present automation was developed under Interquanta.Com’s PHY2006 project.

"Mapping between Globes AEDL and A Particle-Oriented modeling", Globes Heidelberg, Jan2007 Martinez, J Pg.3
When/Where

- POSTER:
- Poster file as sent to J.Kopp pre-presentation