Systems, models and self-awareness
Toward a SysML model of consciousness

Ricardo Sanz
Carlos Hernández

Autonomous Systems Laboratory
Universidad Politécnica de Madrid

www.aslab.org

Content

- Systems engineering for conscious machines
- On Model-based minds
- On Model-based engineering
- The ASys vision of self-aware machines
- The Systems Modelling Language
- Towards a SysML model of consciousness

Systems engineering for conscious machines

The problems of Machine Consciousness

- We all know the many **problems of building conscious machines**.
- Let's summarise them in two big ones:

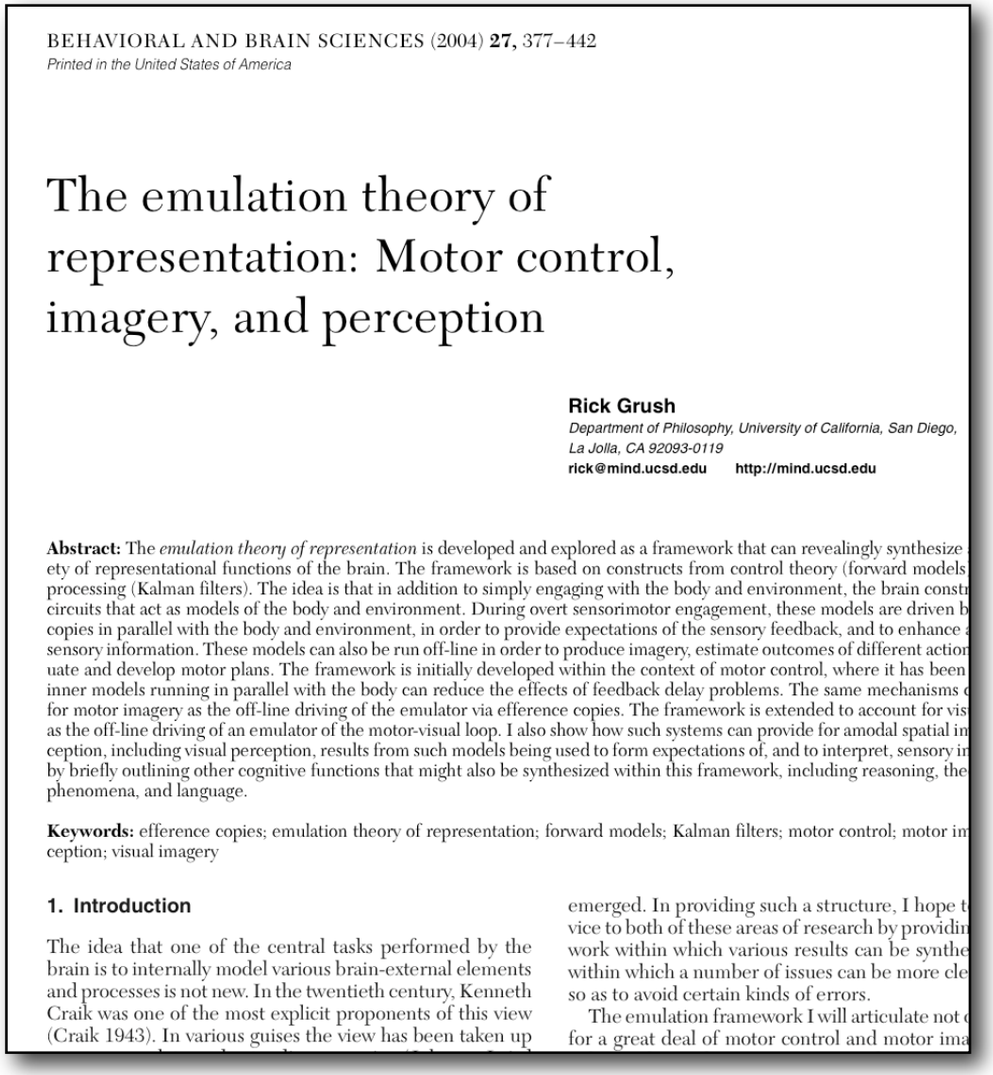
1 We don't know/agree on **what "consciousness" is**

2 We don't know **how to build the thing** (in part because of 1)

Building conscious machines

- To construct conscious machines we can use two major strategies (as described by Holland some time ago): direct and incremental
  - **Direct** approaches: try to build a machine following a particular theory of consciousness (e.g. GWT).
  - **Incremental** approaches: try to add competencies one-at-a-time up to reaching the level of consciousness.
- No matter what is the approach, MC will suffer a receding-horizon phenomenon similar to AI (there is not a ultimate test procedure)

Building conscious machines: the idea



BEHAVIORAL AND BRAIN SCIENCES (2004) **27**, 377–442
Printed in the United States of America

The emulation theory of representation: Motor control, imagery, and perception

Rick Grush
Department of Philosophy, University of California, San Diego, La Jolla, CA 92093-0119
rick@mind.ucsd.edu **http://mind.ucsd.edu**

Abstract: The *emulation theory of representation* is developed and explored as a framework that can revealingly synthesize ety of representational functions of the brain. The framework is based on constructs from control theory (forward models processing (Kalman filters). The idea is that in addition to simply engaging with the body and environment, the brain constr circuits that act as models of the body and environment. During overt sensorimotor engagement, these models are driven b copies in parallel with the body and environment, in order to provide expectations of the sensory feedback, and to enhance sensory information. These models can also be run off-line in order to produce imagery, estimate outcomes of different actio uate and develop motor plans. The framework is initially developed within the context of motor control, where it has been inner models running in parallel with the body can reduce the effects of feedback delay problems. The same mechanisms for motor imagery as the off-line driving of the emulator via efference copies. The framework is extended to account for vis as the off-line driving of an emulator of the motor-visual loop. I also show how such systems can provide for amodal spatial in ception, including visual perception, results from such models being used to form expectations of, and to interpret, sensory in by briefly outlining other cognitive functions that might also be synthesized within this framework, including reasoning, the phenomena, and language.

Keywords: efference copies; emulation theory of representation; forward models; Kalman filters; motor control; motor in ception; visual imagery

1. Introduction

The idea that one of the central tasks performed by the brain is to internally model various brain-external elements and processes is not new. In the twentieth century, Kenneth Craik was one of the most explicit proponents of this view (Craik 1943). In various guises the view has been taken up

emerged. In providing such a structure, I hope t vice to both of these areas of research by providin work within which various results can be synthe within which a number of issues can be more cle so as to avoid certain kinds of errors.

The emulation framework I will articulate not for a great deal of motor control and motor ima

```
/*
 * Assume there are three processes: Pa, Pb, and Pc.  Only Pa can output
 * the letter A, Pb B, and Pc C.
 * Utilizing only semaphores (and no other variables) the processes are
 * synchronized so that the output satisfies the following conditions:
 */

import Utilities.*;
import Synchronization.*;

class Pa extends ABCs implements Runnable {  // extend ABCs to
                                              // access semaphore sum
   public void run () {
      while (true) { nap(1+(int)(random(500)));
         System.out.print("A"); System.out.flush();
         V(sum);
      }
   }
}

class Pb e

   public
      whil
         P
         S
         V
      }
   }
}
```

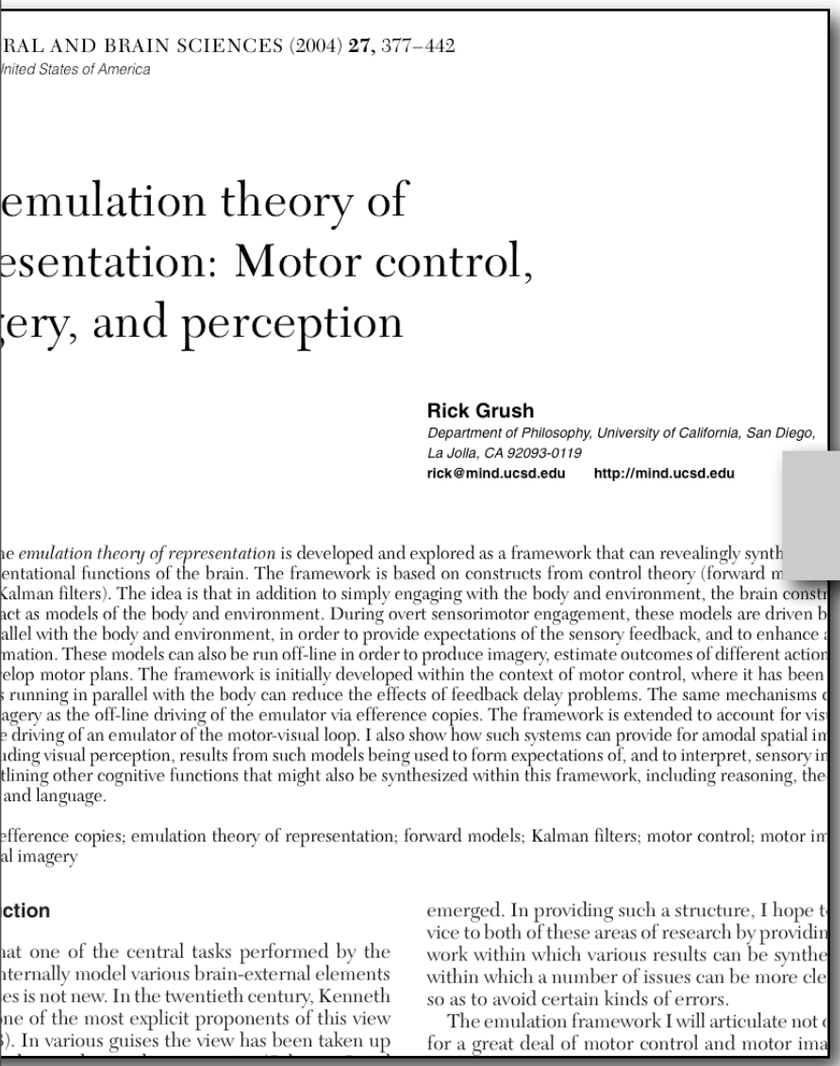
```
class ABCs extends MyObject {

   protected static final BinarySemaphore B       // these semaphores
      = new BinarySemaphore(0);                   // are static
   protected static final BinarySemaphore C       // so subclasses
      = new BinarySemaphore(1);                   // Pa, Pb,
   protected static final CountingSemaphore sum   // and Pc share
      = new CountingSemaphore(0);                 // them

   public static void main(String[] args) {

      Thread pa = new Thread(new Pa());
      Thread pb = new Thread(new Pb());
      Thread pc = new Thread(new Pc());
      pa.start(); pb.start(); pc.start();
      nap(9000);
      pa.stop(); pb.stop(); pc.stop();
      System.out.println();
      System.out.println("B=" + B + ", C=" + C + ", sum=" + sum);
      System.exit(0);
   }
}
```

Building conscious machines: the reality



RAL AND BRAIN SCIENCES (2004) **27**, 377–442
nited States of America

emulation theory of
esentation: Motor control,
gery, and perception

Rick Grush
Department of Philosophy, University of California, San Diego,
La Jolla, CA 92093-0119
rick@mind.ucsd.edu **http://mind.ucsd.edu**

he *emulation theory of representation* is developed and explored as a framework that can revealingly synth
entational functions of the brain. The framework is based on constructs from control theory (forward m
Kalman filters). The idea is that in addition to simply engaging with the body and environment, the brain consti
act as models of the body and environment. During overt sensorimotor engagement, these models are driven b
allel with the body and environment, in order to provide expectations of the sensory feedback, and to enhance
mation. These models can also be run off-line in order to produce imagery, estimate outcomes of different action
elop motor plans. The framework is initially developed within the context of motor control, where it has been
running in parallel with the body can reduce the effects of feedback delay problems. The same mechanisms
agery as the off-line driving of the emulator via efference copies. The framework is extended to account for vis
e driving of an emulator of the motor-visual loop. I also show how such systems can provide for amodal spatial in
uding visual perception, results from such models being used to form expectations of, and to interpret, sensory in
tlining other cognitive functions that might also be synthesized within this framework, including reasoning, the
and language.

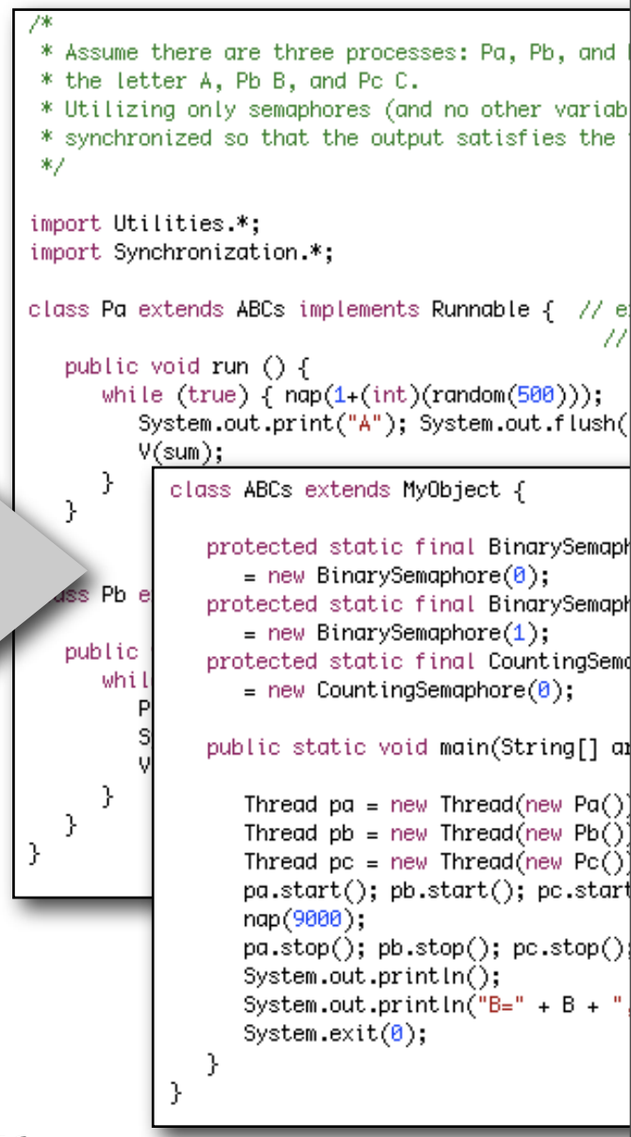
efference copies; emulation theory of representation; forward models; Kalman filters; motor control; motor im
al imagery

ction

at one of the central tasks performed by the
nternally model various brain-external elements
es is not new. In the twentieth century, Kenneth
ne of the most explicit proponents of this view
). In various guises the view has been taken up

emerged. In providing such a structure, I hope t
vice to both of these areas of research by providin
work within which various results can be synthe
within which a number of issues can be more cle
so as to avoid certain kinds of errors.

The emulation framework I will articulate not
for a great deal of motor control and motor ima



```
/*
 * Assume there are three processes: Pa, Pb, and
 * the letter A, Pb B, and Pc C.
 * Utilizing only semaphores (and no other variab
 * synchronized so that the output satisfies the
 */

import Utilities.*;
import Synchronization.*;

class Pa extends ABCs implements Runnable {  // e
                                                //
   public void run () {
      while (true) { nap(1+(int)(random(500)));
         System.out.print("A"); System.out.flush(
         V(sum);
      }
   }

ass Pb e

   public
      whil
         P
         S
         V
      }
   }
}
```

```
class ABCs extends MyObject {

   protected static final BinarySemaph
      = new BinarySemaphore(0);
   protected static final BinarySemaph
      = new BinarySemaphore(1);
   protected static final CountingSemo
      = new CountingSemaphore(0);

   public static void main(String[] ar

      Thread pa = new Thread(new Pa()
      Thread pb = new Thread(new Pb()
      Thread pc = new Thread(new Pc()
      pa.start(); pb.start(); pc.start
      nap(9000);
      pa.stop(); pb.stop(); pc.stop()
      System.out.println();
      System.out.println("B=" + B + "
      System.exit(0);
   }
}
```

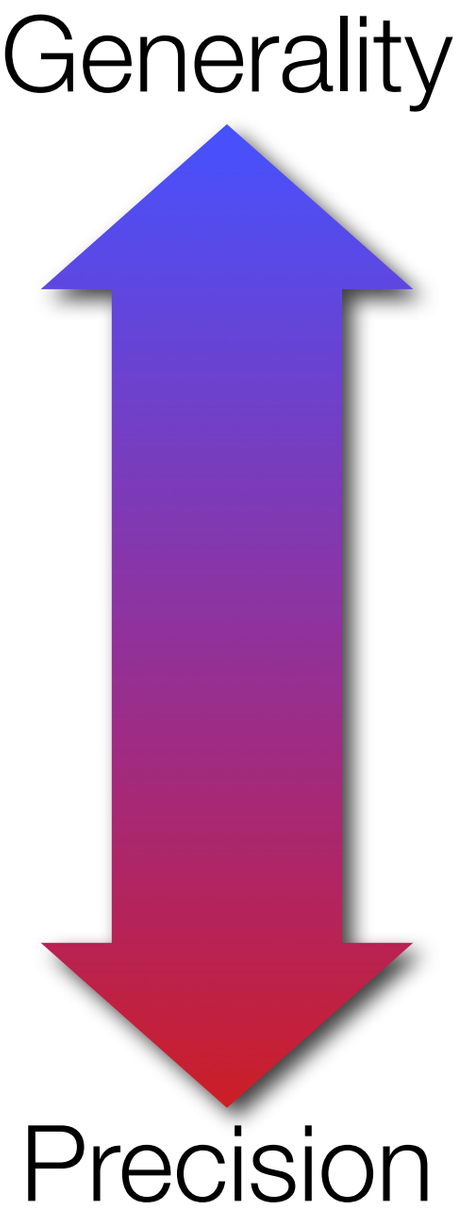
Magic of the consciousness engineer

Towards a Positive Theory of Consciousness (I)

- There are too many basic understandings of what consciousness is: process, function, module, property, emergent phenomenon, quantum state, etc.
- The problem can be traced back to a common architectural reverse-engineering problem: the **extraction of function from non-formal description of architectures**.
- A rapid scan of the literature on the topic leaves the following impressions:
  - Most theories that target the whole thing seem just literature (or plain rubbish, or love & hate manifestations)
  - Positive theories are mostly seem as partial and naïve

Towards a Positive Theory of Consciousness (II)

- We need a unified theory of “consciousness”:
  - That targets the whole thing (i.e. even qualia)
  - That is widely agreed across disciplines
  - That is expressible in different abstraction levels to be at the same time
    - general and
    - precise and
    - verifiable



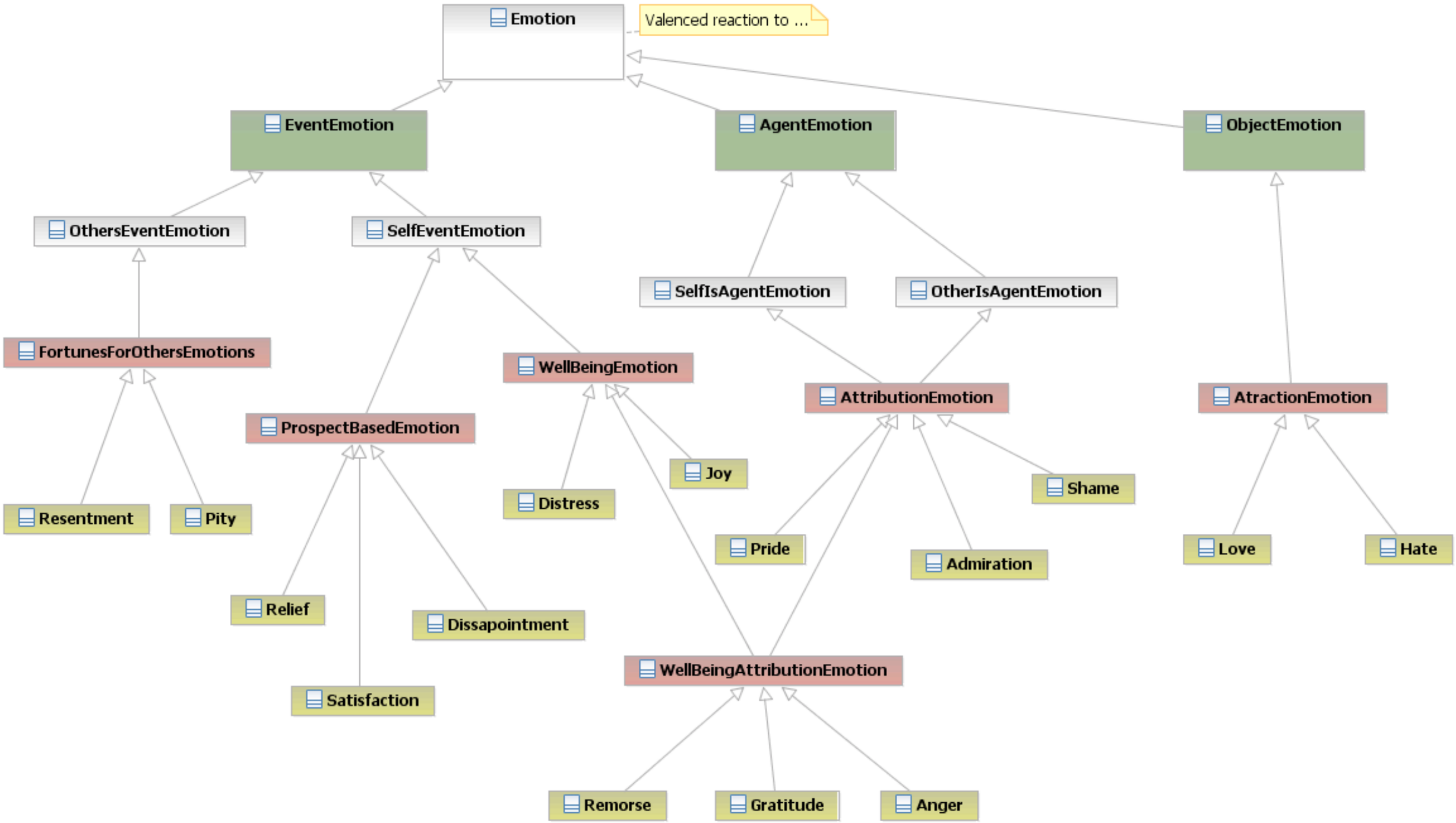
Towards a Positive Theory of Consciousness (III)

- We must **express the theory in a formal enough language** as to minimise the problem of multiple interpretations
- Some approaches to “formal theories” are available but are far from being accepted as targeting the whole (e.g. Tononi’s)
- Some other approaches are just incomprehensible (e.g. Zelzenikar’s).
- There’s even some researchers that think that formalisation of consciousness is untenable
- But ... **this is a necessary step in science**.

Modes of expression

- Verbal-Linguistic
  - Sharing words, writings, documents
  - Why are we stuck here?
- Logical-Mathematical
  - Equations, drawings, MatLab models
  - Engineering/scientific disciplinary
- Graphic-Visual
  - Pictures, charts, drawings
  - Used by many people!
- There are other methods too…

Ortony et al. emotions



Rationale for a Systems Approach to Consciousness

- The recognition of the enormous complexity of the issue of building conscious machines points into a direction that can help solve both problems.

- Complex systems engineering is addressed using the methods of the so-called discipline of ”Systems Engineering”

- This sits in the middle of technical and managerial issues, addressing the multifaceted problems of large-scale, complex systems engineering

- In this work we propose the use of recent systems engineering modelling methods to address the complexities of MC theorising

Core Objective

- The core objective of this approach is the development of **Reference Models of Consciousness**

- Aim:

  - Consolidate a unified vision on consciousness functions and mechanisms

  - To organise knowledge about consciousness components into standardised, reusable and extensible models

  - To develop methods for (re-)using this knowledge in support of the construction of conscious systems

The Nature of Models

"Modeling, in the broadest sense, is the cost-effective use of something in place of something else for some cognitive purpose.

...

A model represents reality for the given purpose; the model is an abstraction of reality in the sense that it cannot represent all aspects of reality. This allows us to deal with the world in a simplified manner, avoiding the complexity, danger and irreversibility of reality."

[Rothemberg-1989]

Reference Models of Consciousness Approach

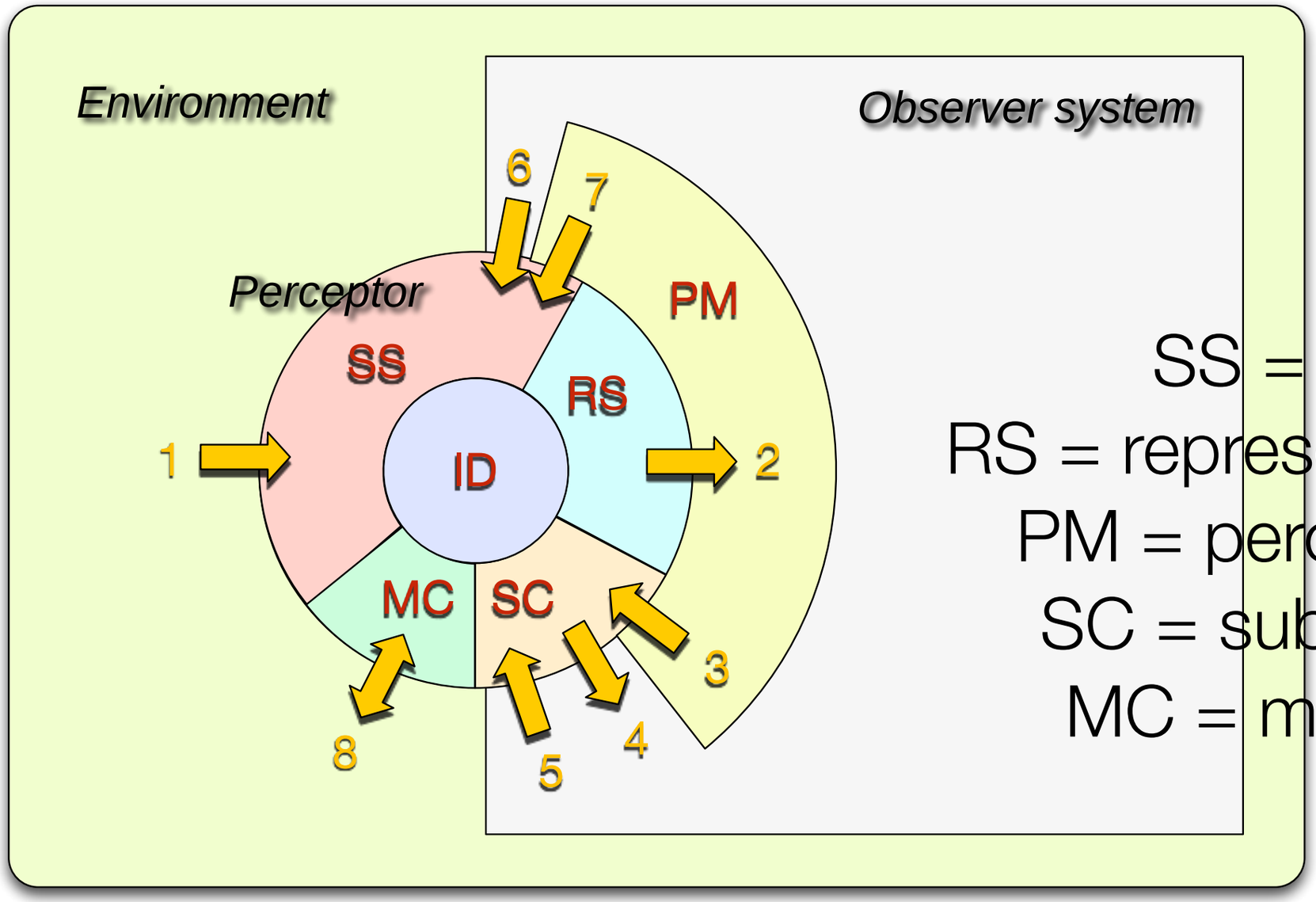
- Method:
  - Define an ontology for describing consciousness components and systems
  - Define reference models as standardised elements that include knowledge and information about the form, function and behaviour of components
  - Formalise the ontology and reference models in SysML
  - Create semantic mappings between SysML, cognitive science, systems biology and engineering tools

Targetting the two problems at once

- This approach can not only address problem 1 (**what "consciousness" is**) but can also help with problem 2 (**how to build the thing**)
- This can be possible by the unification of
  - the emerging model-based theory of consciousness and
  - the modern model-based practice of embedded systems construction
- Let's see some of the bricks of this approach

On Model-based minds

The question of perception

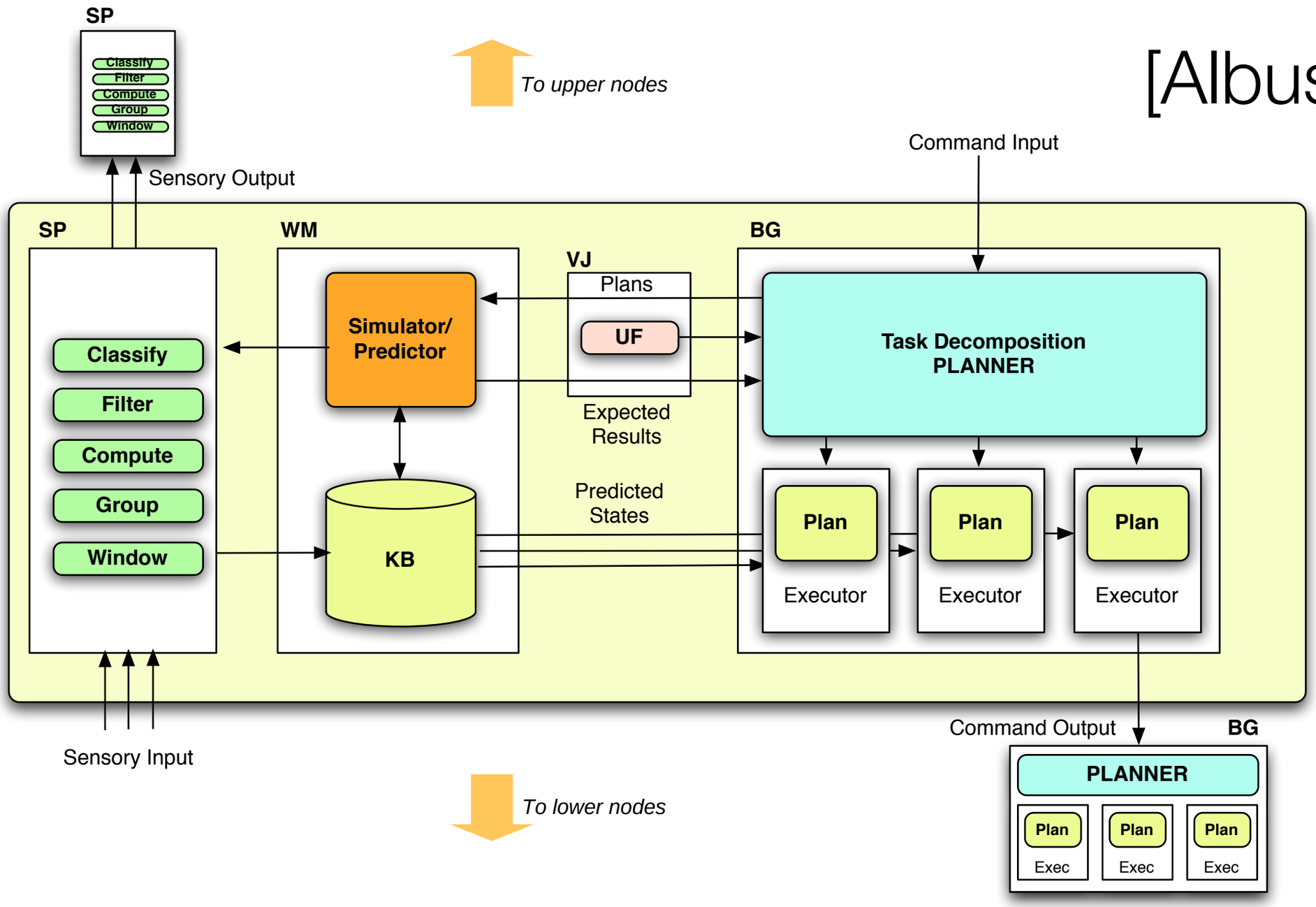


SS = sensory system
RS = representation system
PM = perceptual memory
SC = substratal coupling
MC = marginal coupling

[López 2007]

The question of action

[Albus 1999]



The question of Knowledge?

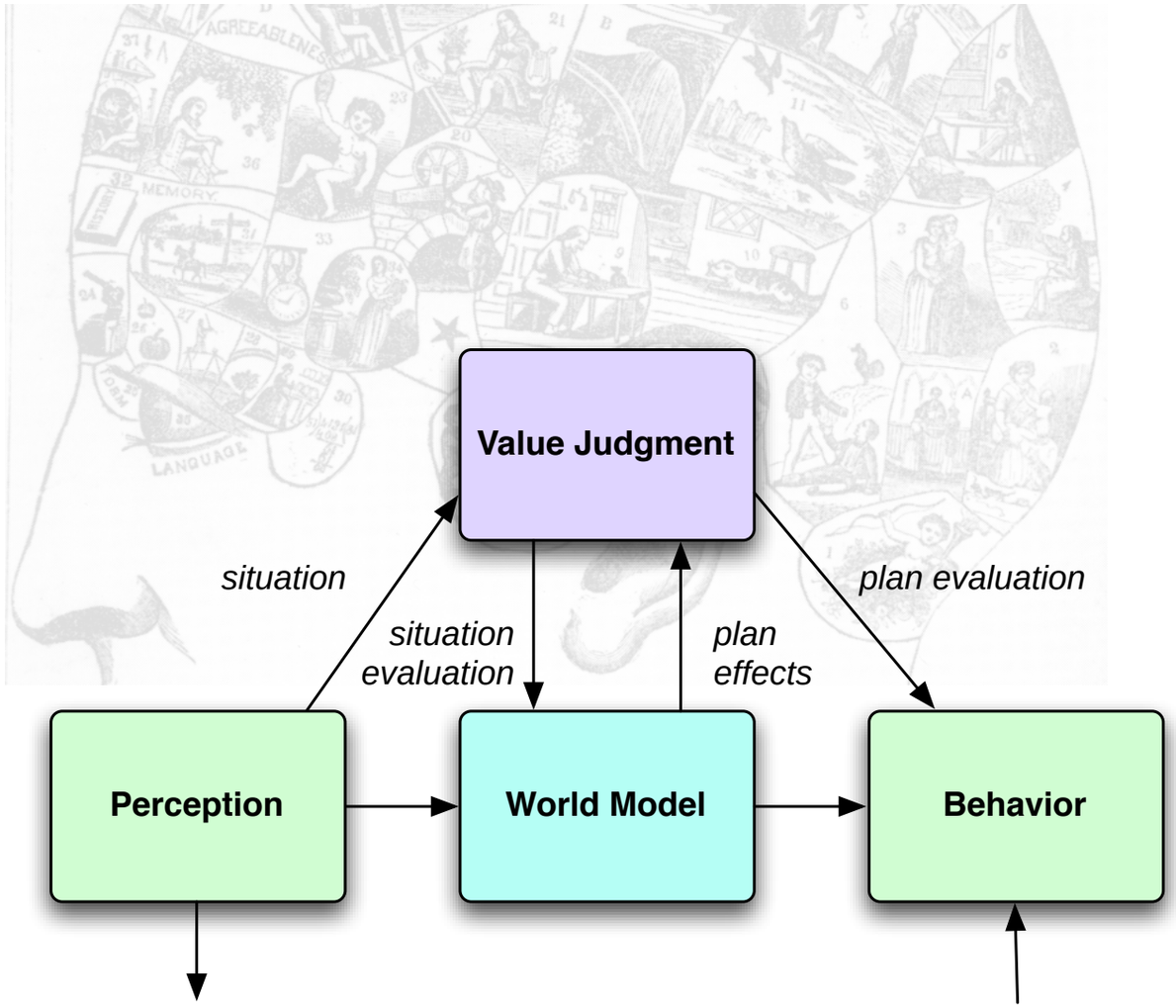
- Knowledge is composed by ***executable dynamic models***
  - Models about some (partial) reality in/out the agent
  - that may be executed
    - Over a physical ***execution engine*** (*e.g.* cerebellum)
    - Over a virtual machine running over a physical execution engine or another virtual machine
  - May be degenerated (e.g. a simple static value)
  - May be embodied (aka "*precompiled*" with the execution engine)
  - May be shared among heterogeneous engines

The end of the questioning

- This can be summarised in a single sentence:

Minds are model-based controllers

- They can be direct/inverse,
implicit/explicit,
static/dynamic,
isolated/coupled,
“genetic”/”memetic”/learned,
homogeneous/heterogeneous,
postdictive/predictive,
etc.



On Model-based systems engineering

What is Systems Engineering?

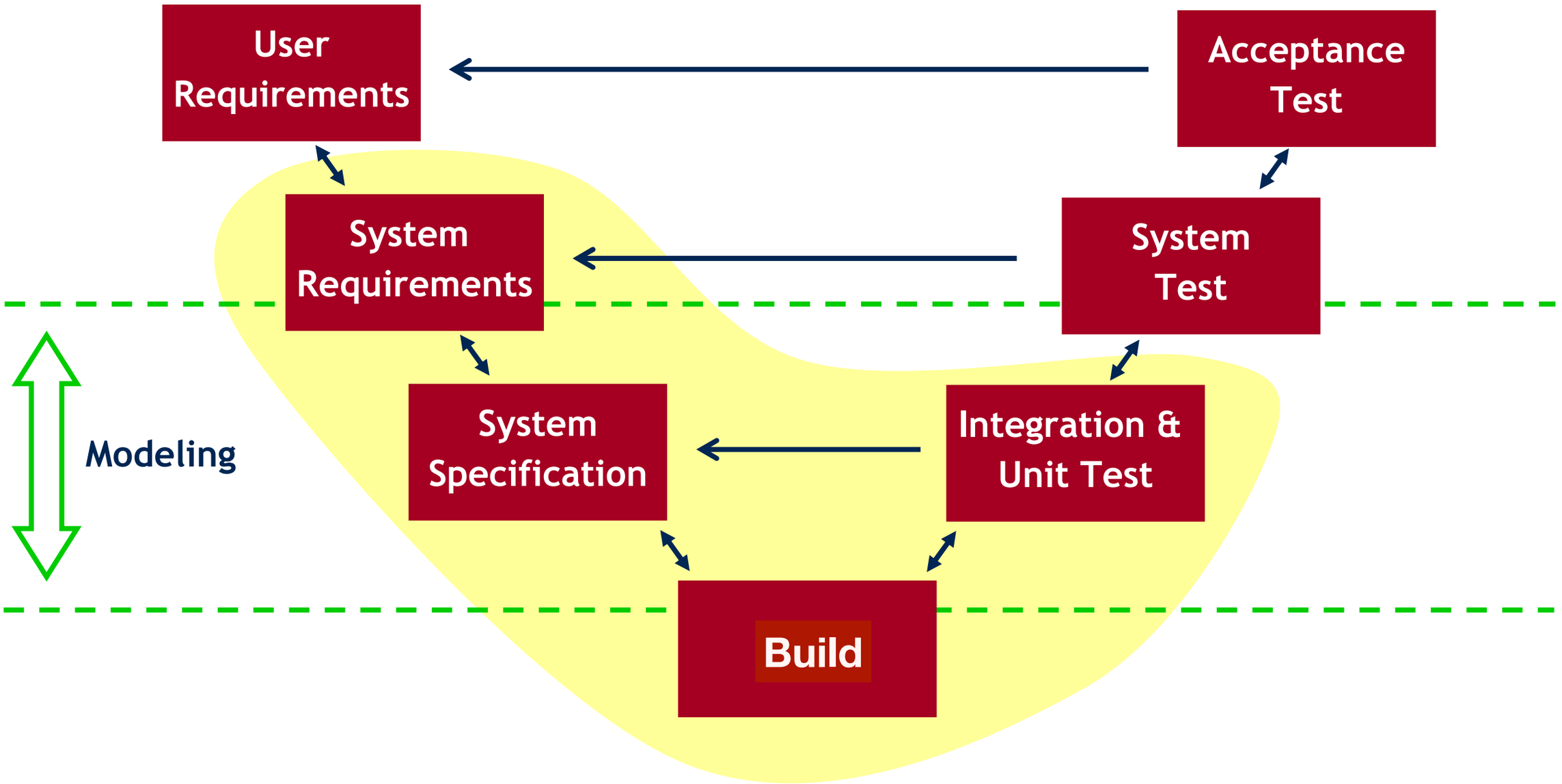
Systems Engineering is an interdisciplinary approach and means to enable the realisation of successful systems.

It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem.

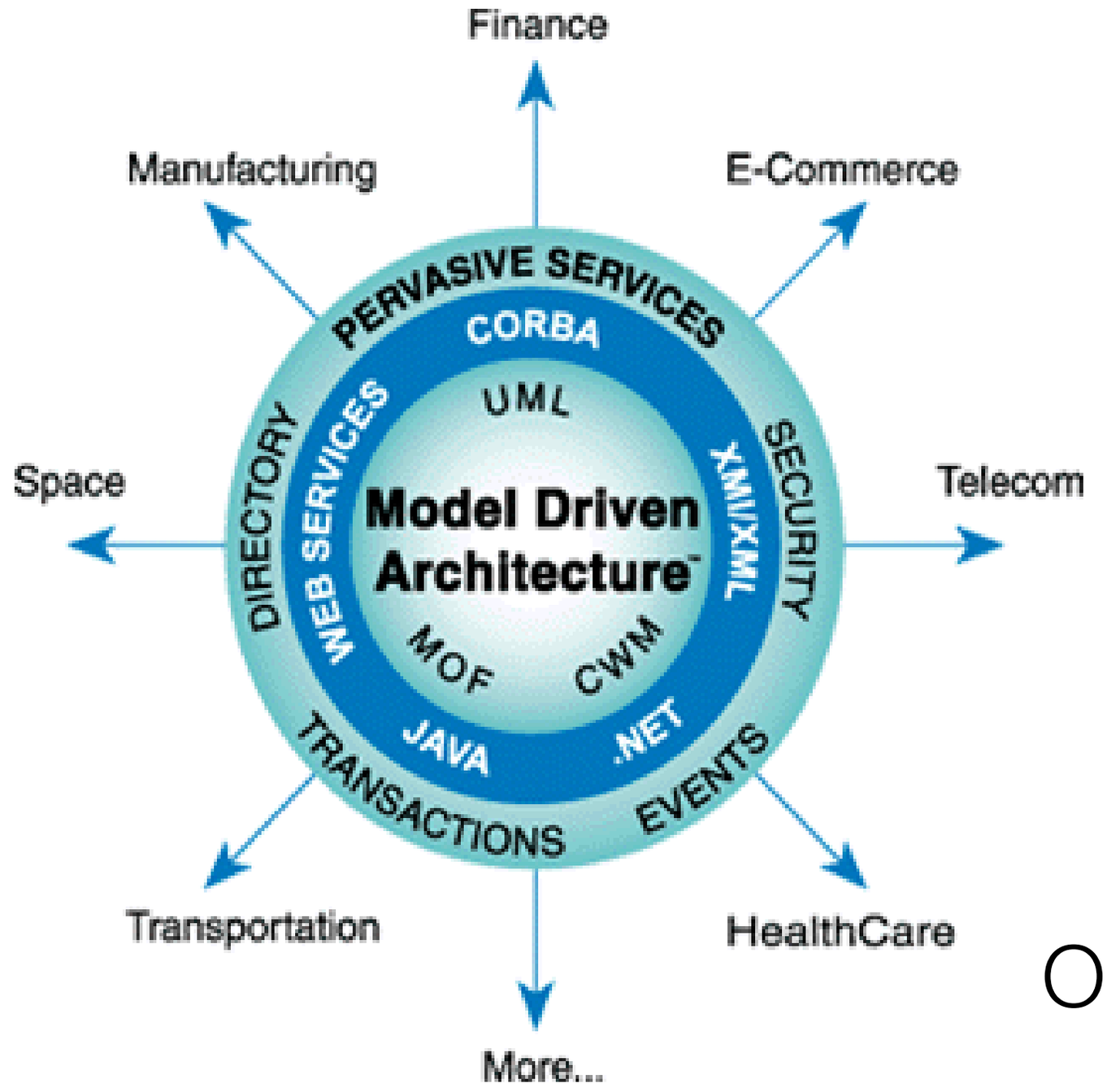
Does this match our needs?

- Talking about “customers” and “needs” may seem far-off for research on machine consciousness.
- In this field we can identify three classes of global research objectives for MC:
  - Building machines like us (the C3PO drive)
  - Understanding biological consciousness
  - Building better machines (the misgone HAL dream)
- All them specify needs for our realisations and SE is a systematic way of addressing those needs (aka requirements)

The V-model of Development

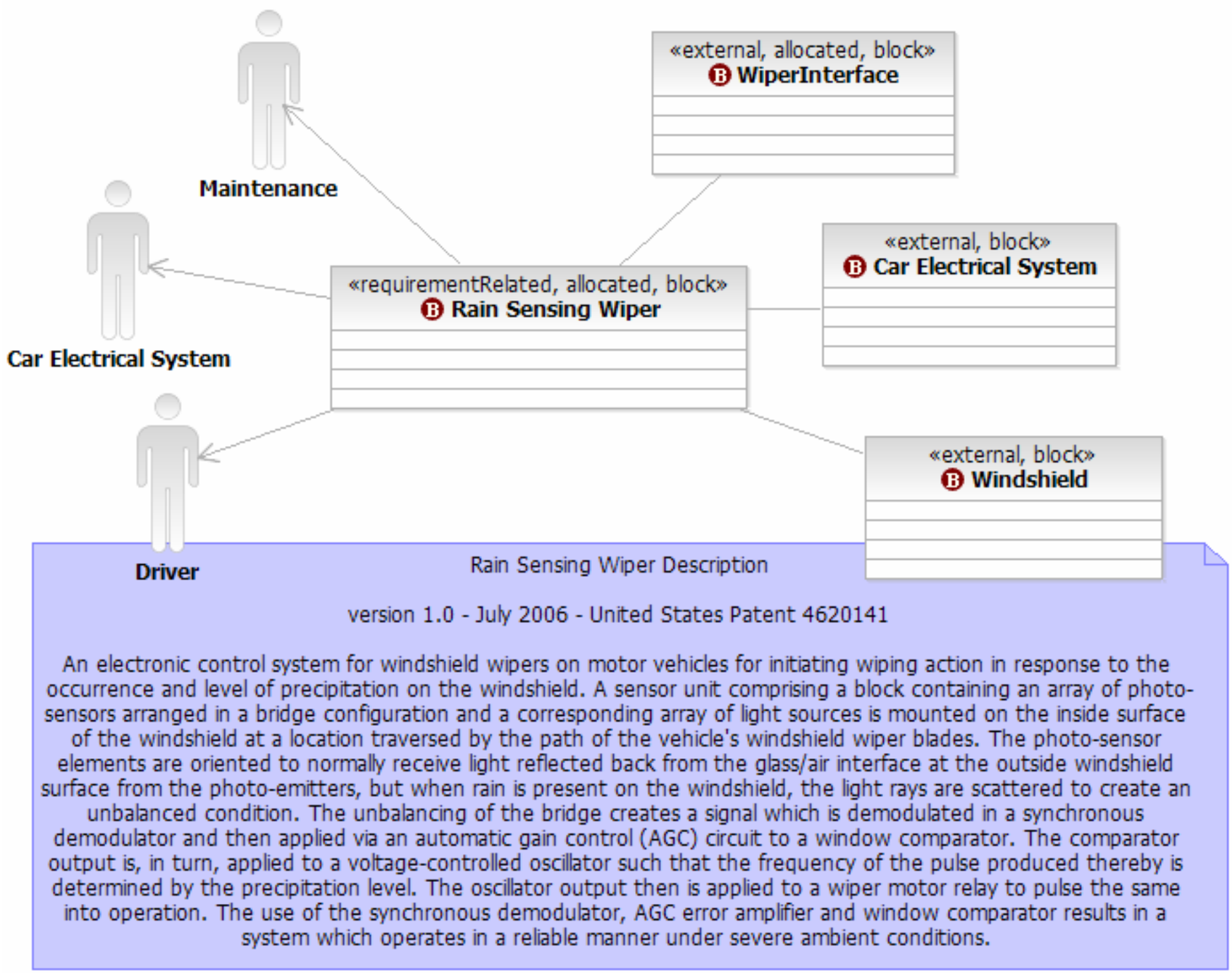


A Focus on Software and Architecture

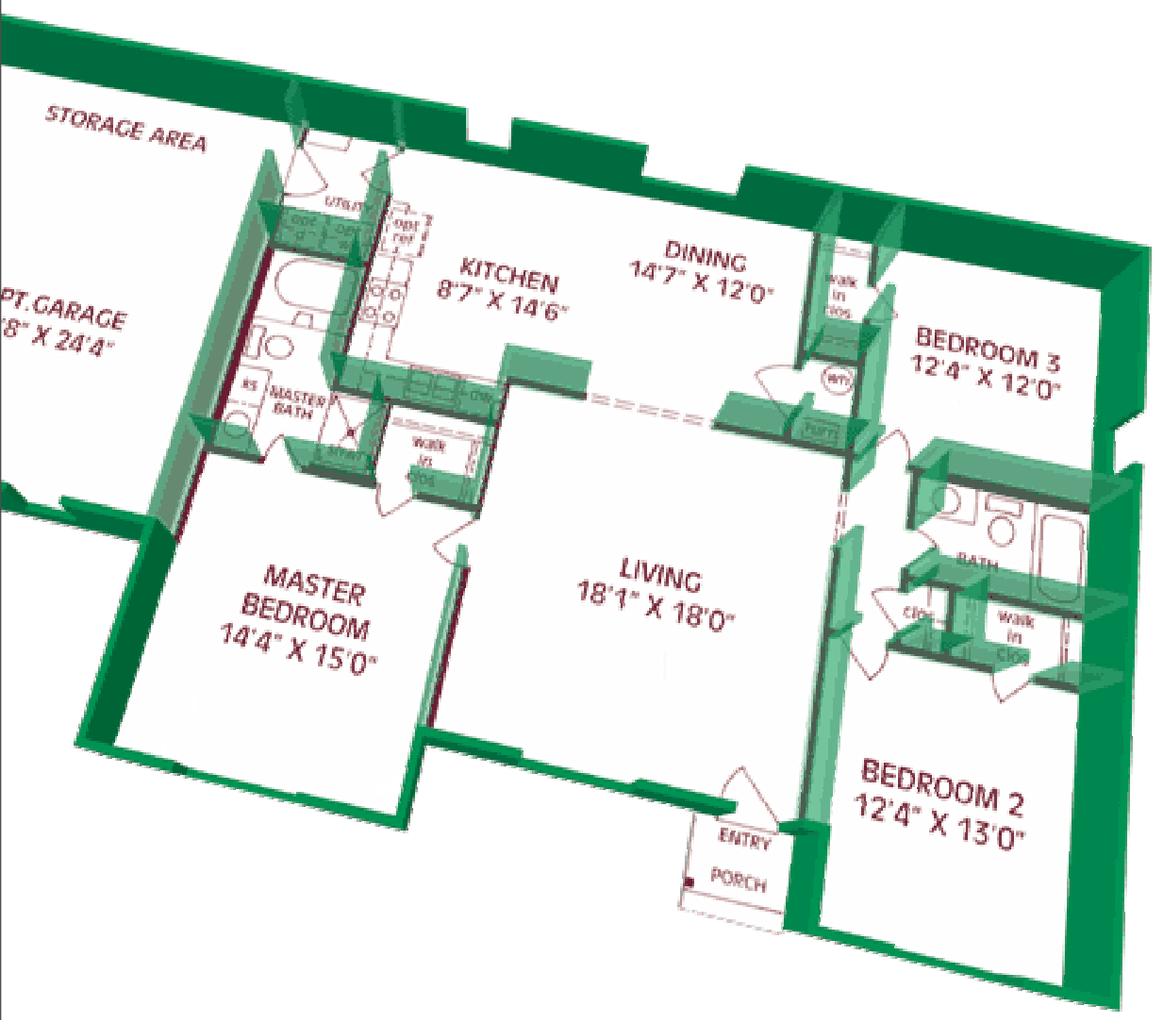


OMG's **MDA**

Context diagram for autonomous wiper



From Model to System



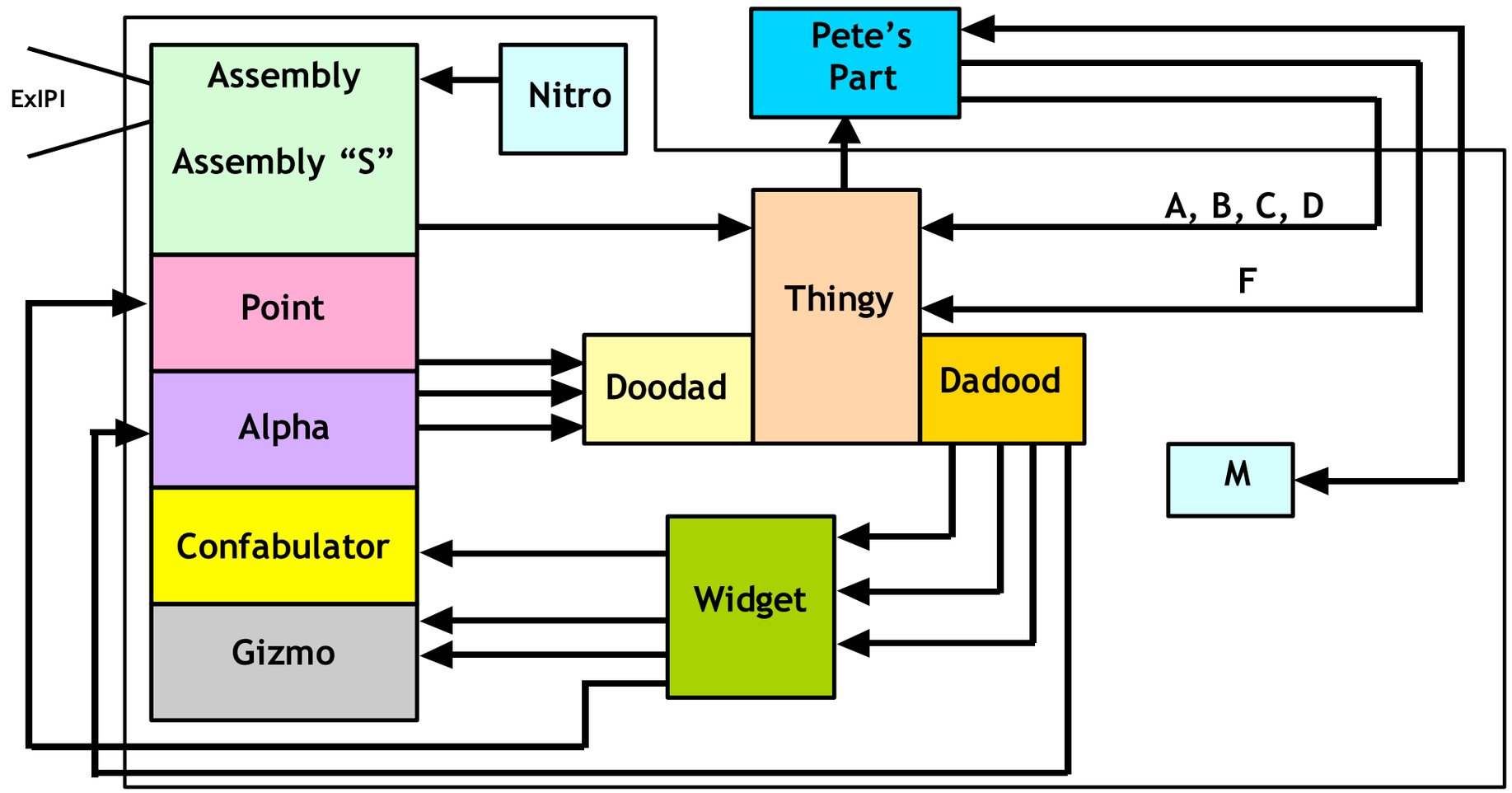
A Text Model

of complex systems by means of scalable design patterns. This approach is specially well captured in the multiresolutional approach fostered by the control design pattern that Meystel calls the *elementary loop of functioning* (Meystel, 2003). Of importance in relation with the ASys theory of meaning is the incorporation of value judgment mechanisms over this elementary loop (see Figure 7).

The elementary loop of functioning, when applied hierarchically, generates a multiresolutional ladder of meanings specifically focused on the controllable subspace of each control level. This approach partitions both the problem of meaning generation and the problem of action determination, leading to hierarchical control structures that have interesting properties of self-similarity.

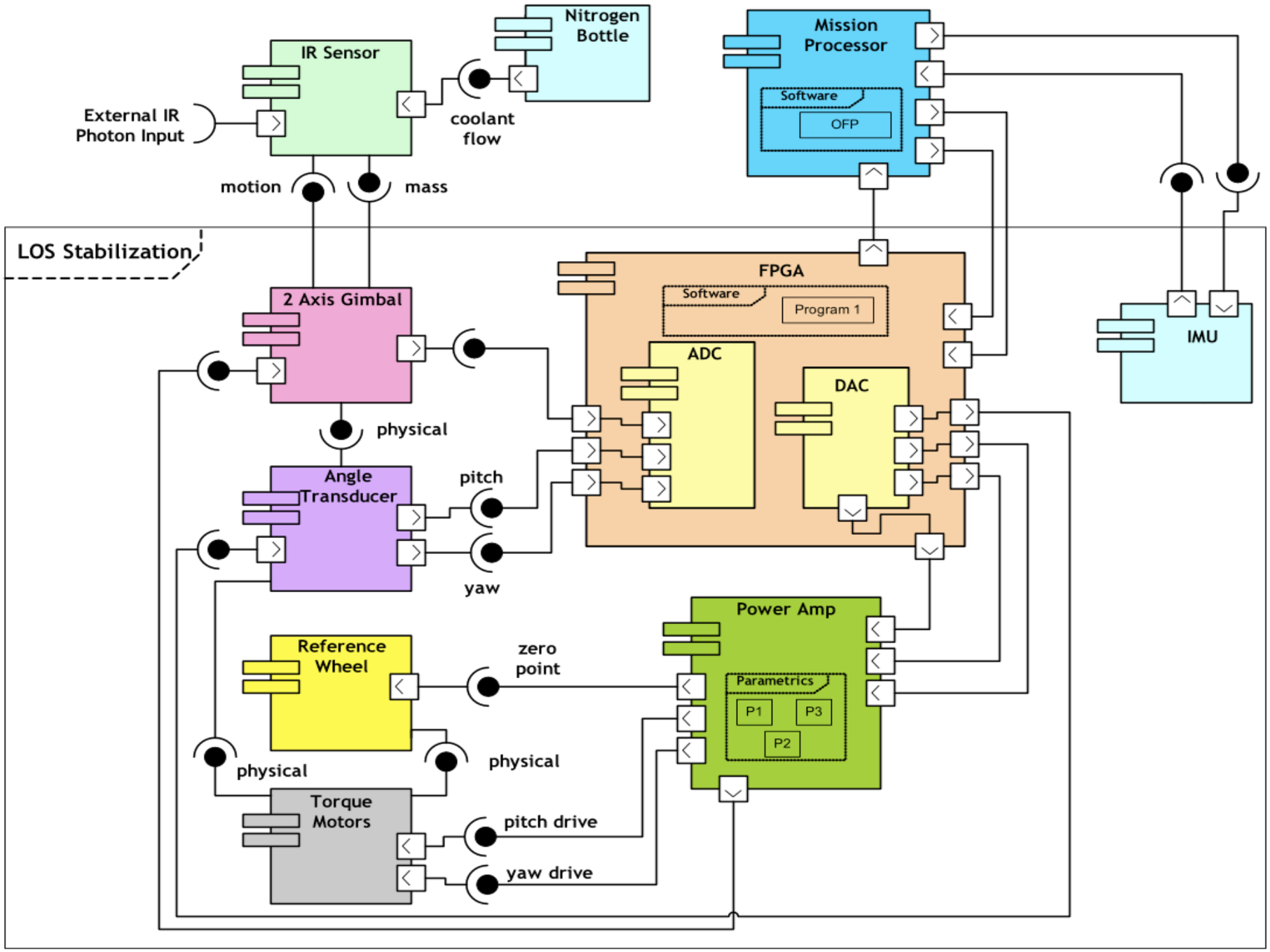
This core design pattern approach is extended in the concept of a control node of the RCS control architecture (Albus, 1992) (see Figure 7). Beyond the model of the world and the sensing and acting units, this architecture considers the existence of a value judgment unit that evaluate both static states and dynamic states derived from hypothetical plan execution.

A (soft) block model

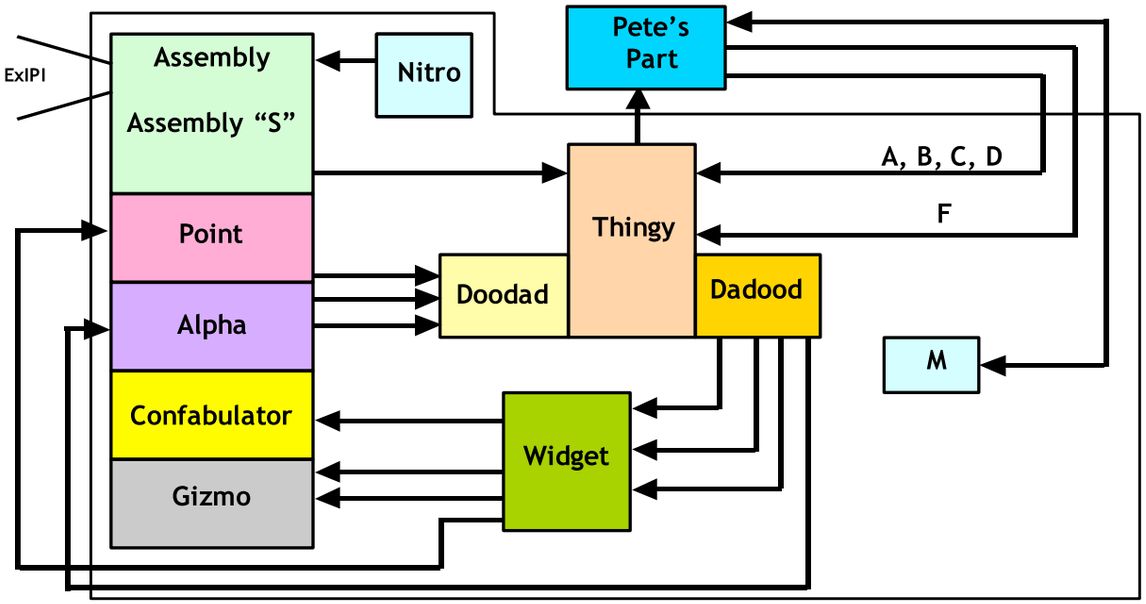


"The System"

A semi-formal block model

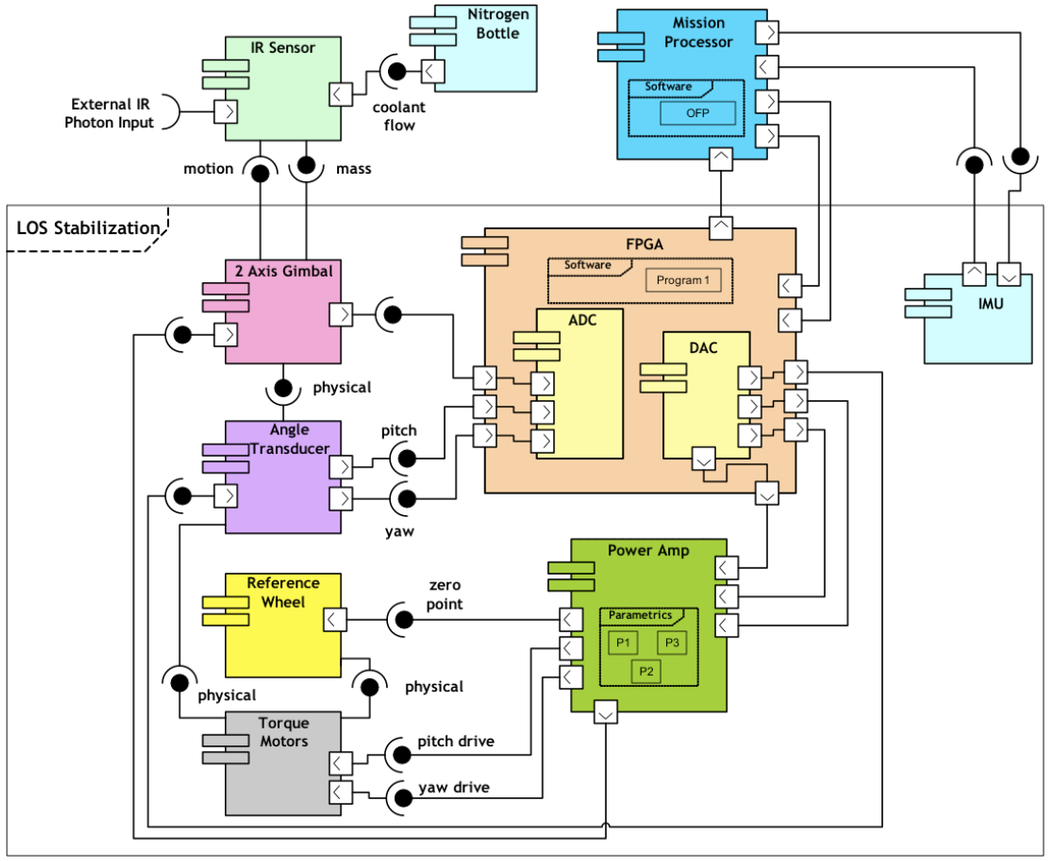


Compare and Contrast



"The System"

Block diagram with random or inconsistent meaning to symbols



Component diagram with well-defined syntax and grammar

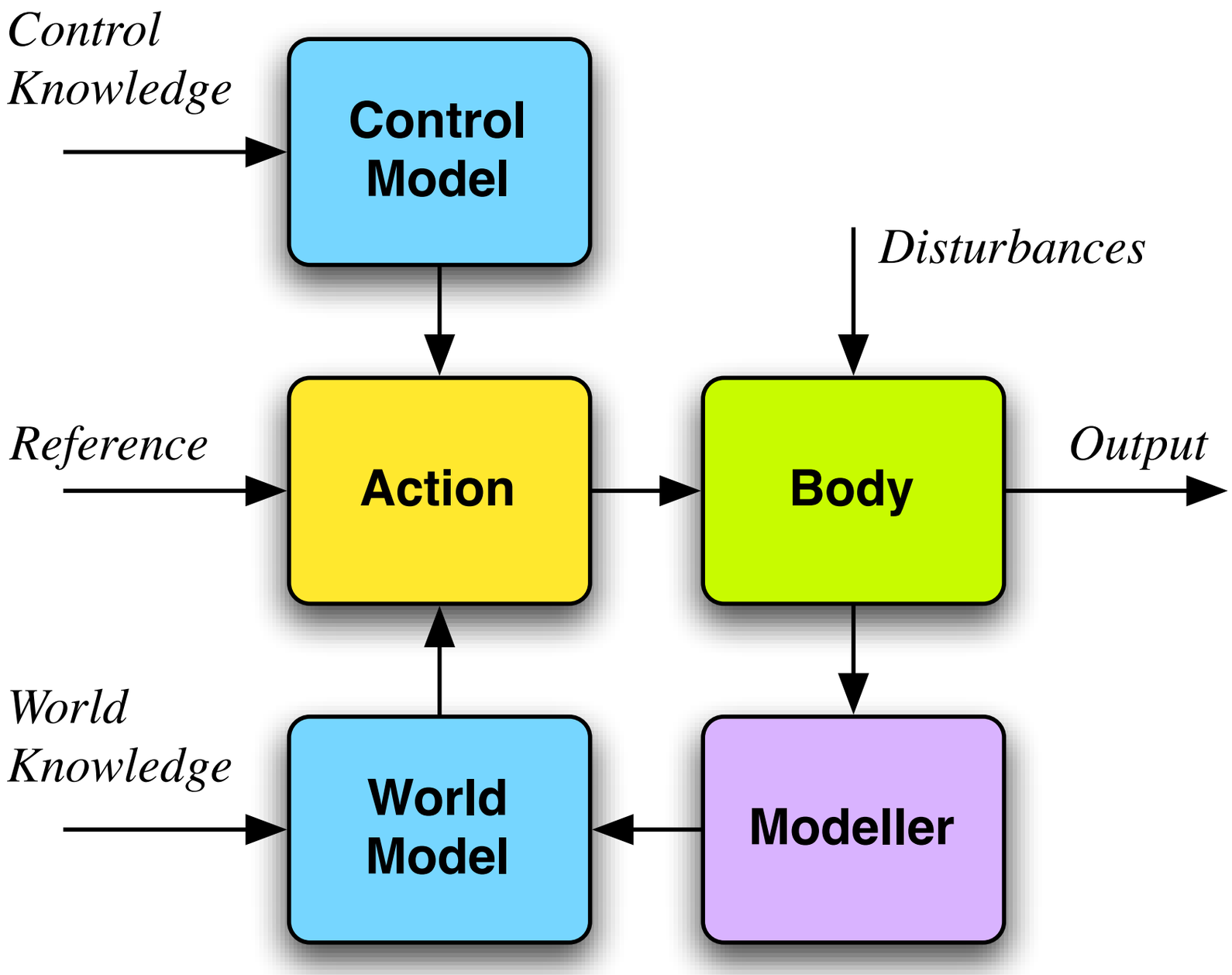
Communication is key !

The engineer (scientist) must use a consistent, well-defined, and well-understood language to communicate the requirements and design to other stakeholders (engineers or not), otherwise the product will be questionable, founder, fail, or be a full disaster.

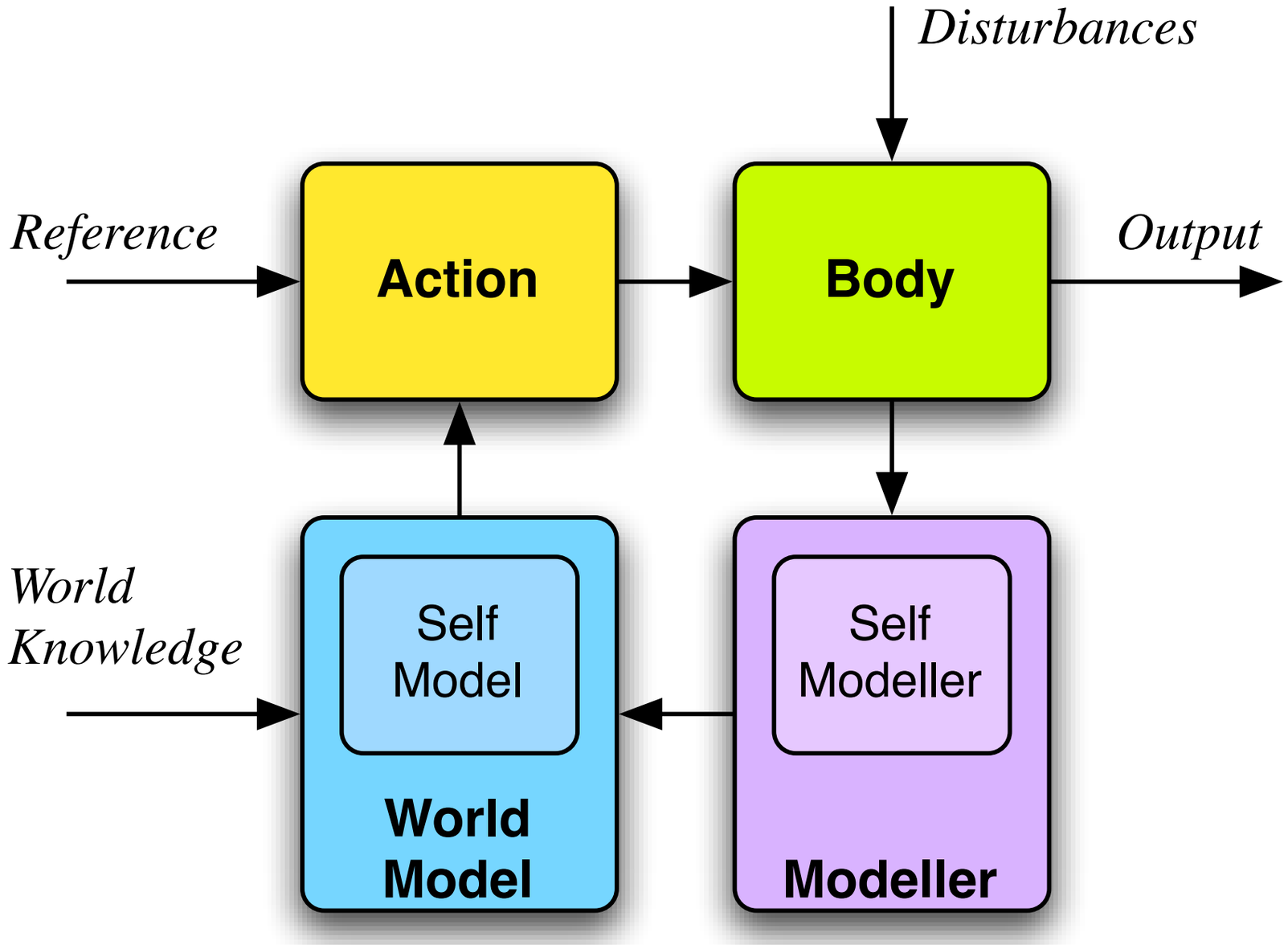
For the systems engineer, that language is currently SysML.

The ASys vision of self-aware machines

Expliciting Models



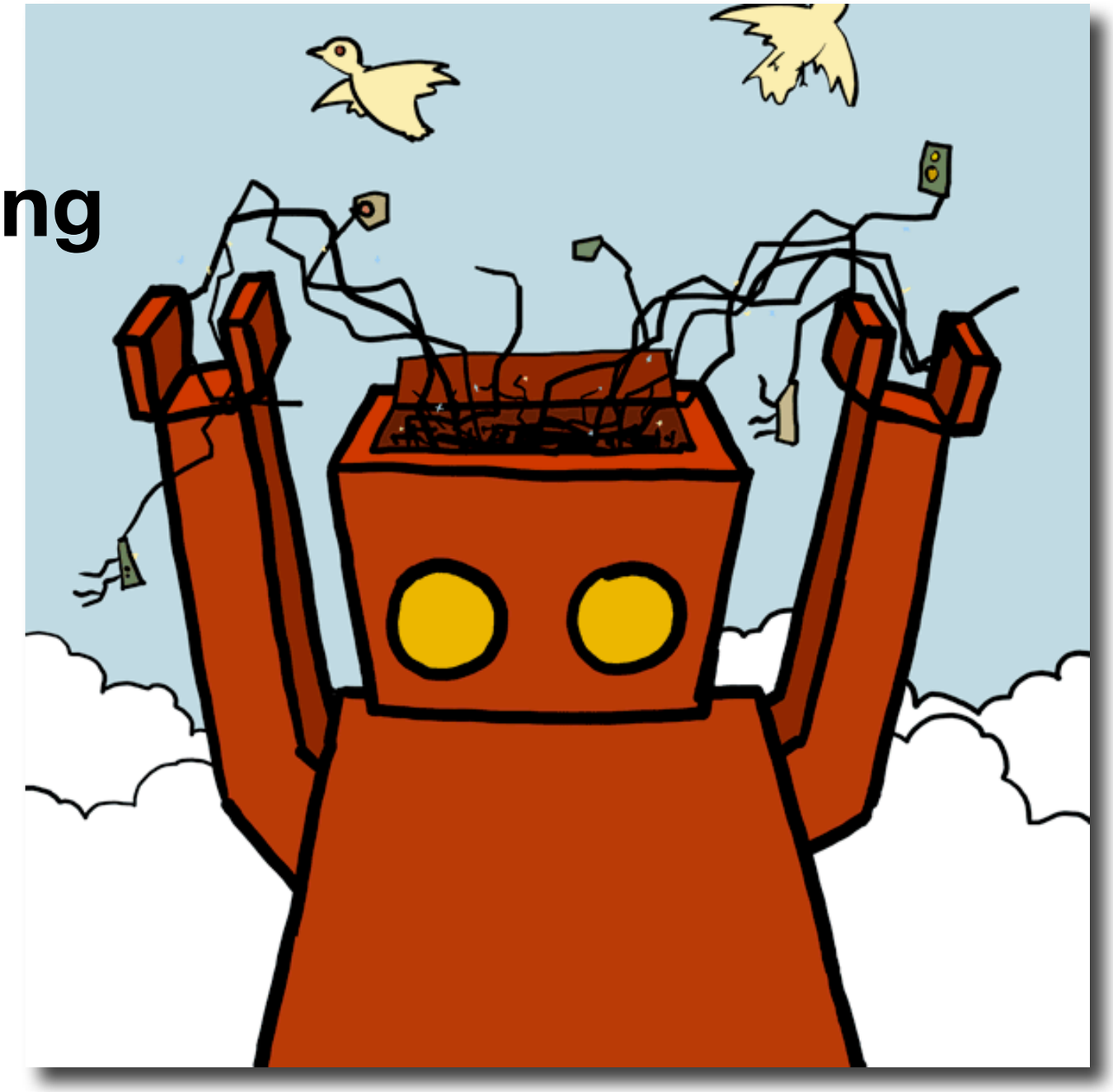
The ‘Self’ Step



Self = Access to Self-implementation

By means of **model-integrated engineering** of the system itself

Use the very engineering models as run-time **self-models**



Principles

- **Model-based cognition**: A cognitive system exploits models of other systems in their interaction with them.
- **Model isomorphism**. An embodied, situated, cognitive system is as good performer as its models are.
- **Anticipatory behavior**. Except in degenerate cases, maximal timely performance is achieved using predictive models.

Principles

- **Unified cognitive action generation.** Generate action based on an integrated, scalable, unified model of task, environment and self in search for global performance maximisation.

- **Model-driven perception.** Perception is realised as the continuous update of the integrated models used by the agent in a model-based cognitive control architecture by means of real-time sensorial information.

Principles

- **System awareness.** An aware system is continuously perceiving and generating meaning -future value- from the continuously updated models.

- **System self-awareness/consciousness.** A conscious system is continuously generating meanings from continuously updated self-models in a model-based cognitive control architecture.

The Systems Modelling Language

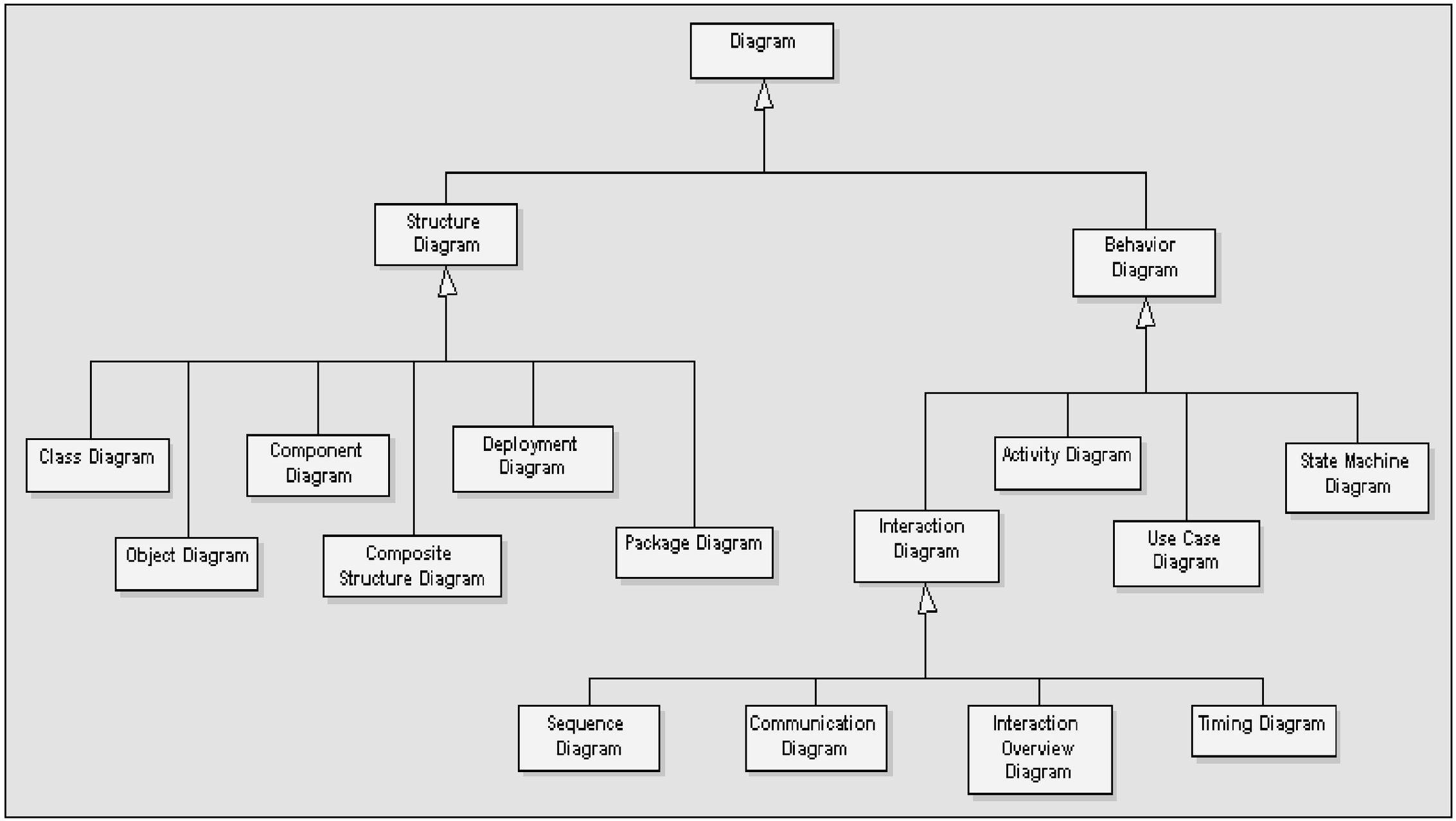
What is the Systems Modeling Language ?

- A model language for holistic system representation and systems engineering
- A graphical/textual semi-formal modeling language addressing the issues of the Systems Engineering RFP developed by the OMG, INCOSE, and AP233
- Can be seen as a UML Profile that represents a subset of UML 2 with extensions
- Supports the specification, analysis, design, verification and validation of systems that include hardware, software, data, personnel, procedures, and facilities

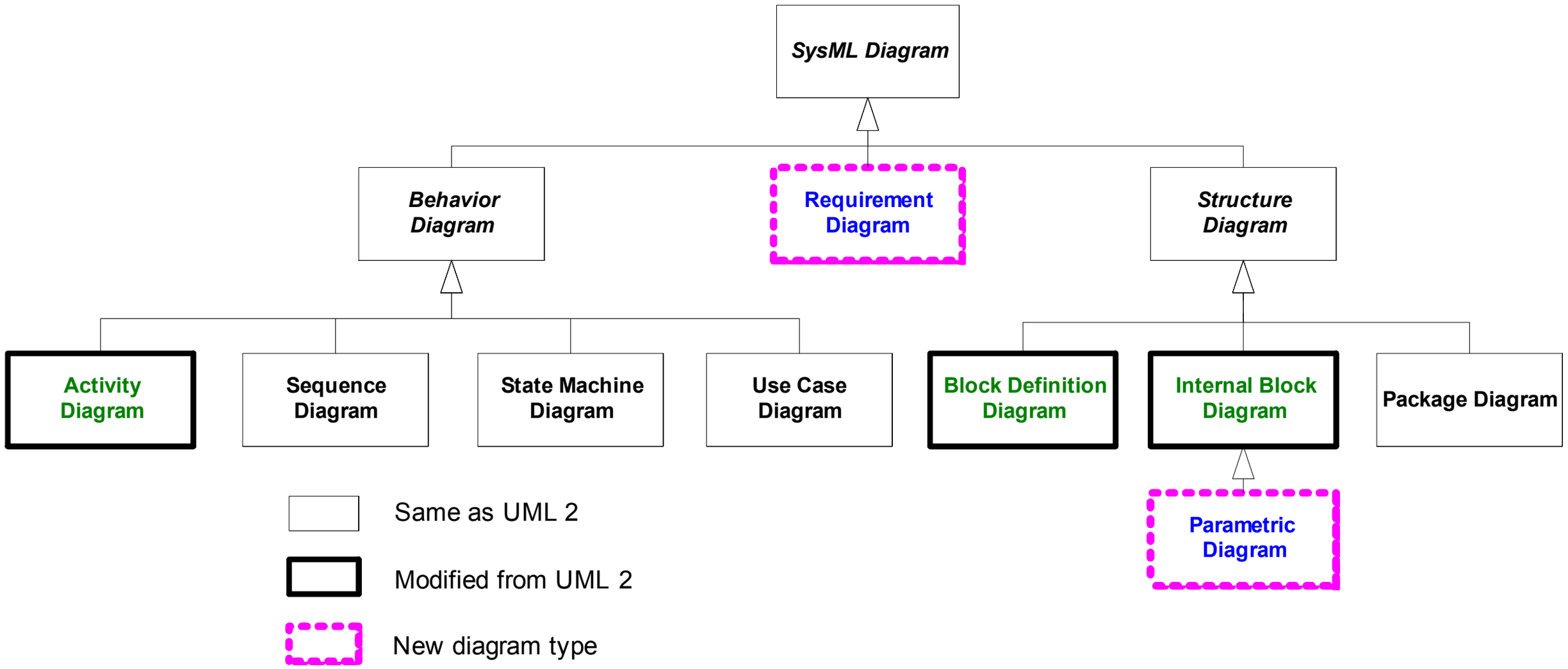
Some aspects of SysML

- SysML is broader than sofware-centric modelling languages
- Can capture all salient aspects of complex system design
- Is quite intuitive for system engineers
- Supporting proven systems engineering concepts like requirements, hierarchical block structuring and parametrics
- The language has been designed to not be a barrier to traditional system engineering methods

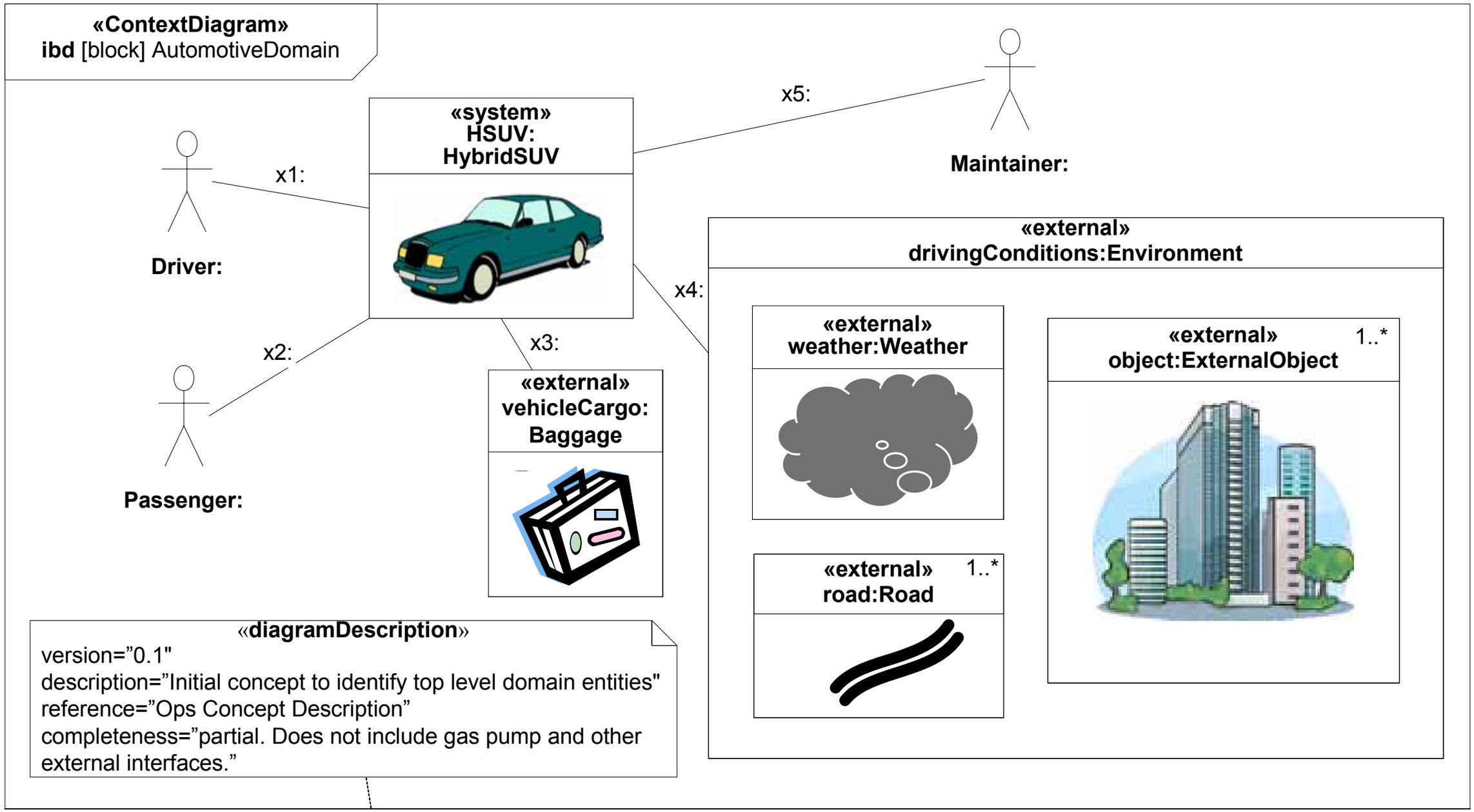
UML 2 Diagrams



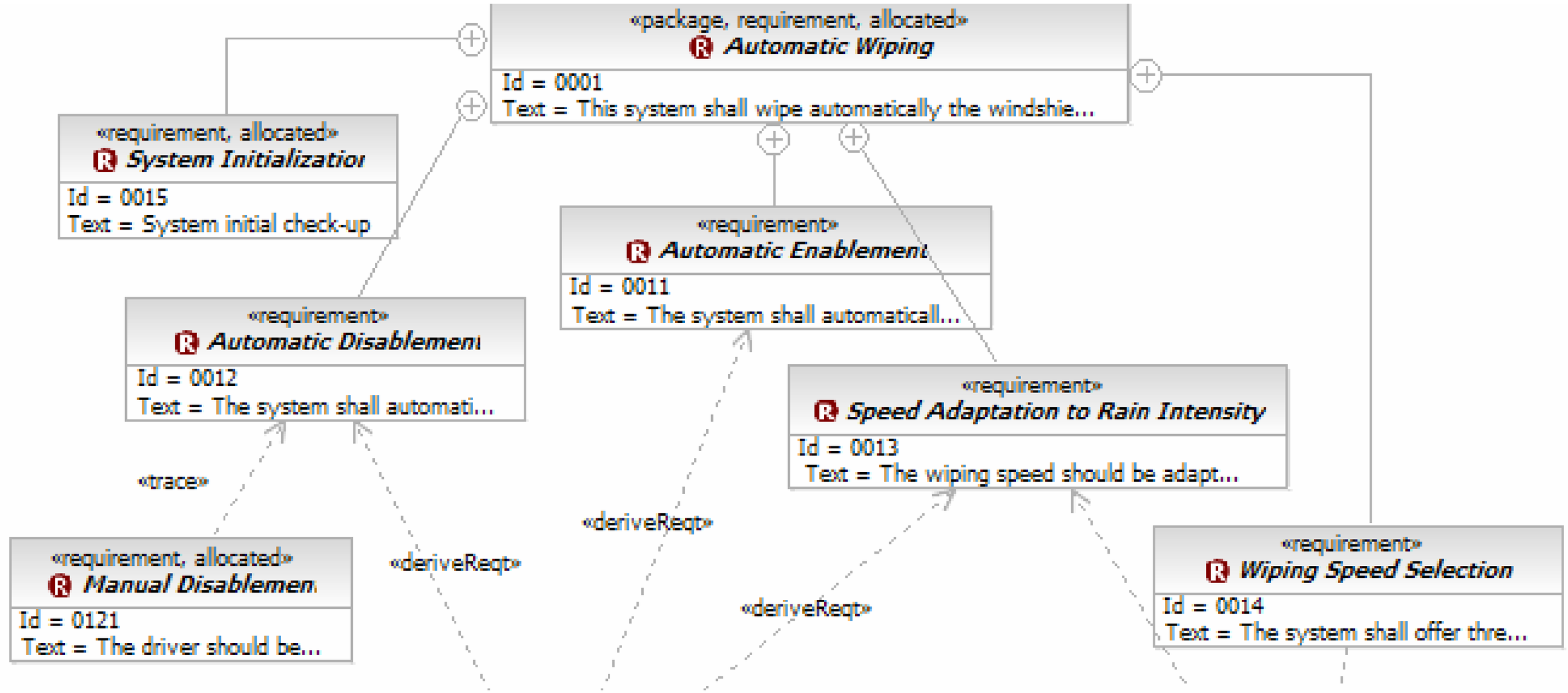
SysML Diagrams



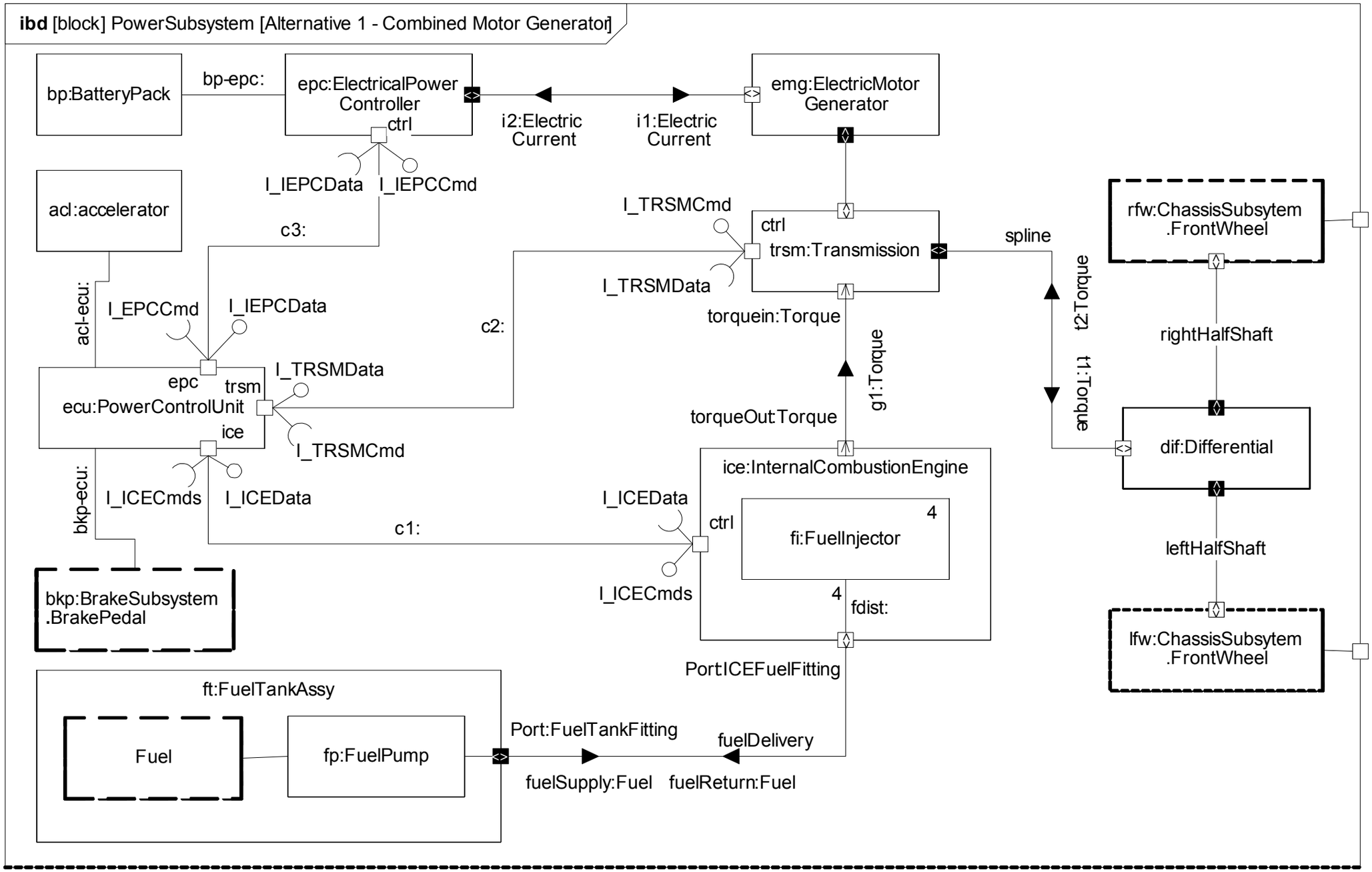
Context diagram for a vehicle



Requirements diagrams



Internal Block Diagram



SysML Drawing Summary

SYSML DIAGRAM	PURPOSE	UML ANALOG
Activity diagram	Show system behavior as control and data flows. Useful for functional analysis. Compare Extended Functional Flow Block diagrams (EFFBDs), already commonly used among systems engineers.	**Activity diagram**
Block Definition diagram	Show system structure as components along with their properties, operations and relationships. Useful for system analysis and design.	**Class diagram**
Internal Block diagram	Show the internal structures of components, including their parts and connectors. Useful for system analysis and design.	**Composite Structure diagram**
Package diagram	Show how a model is organized into packages, views and viewpoints. Useful for model management.	**Package diagram**
Parametric diagram	Show parametric constraints between structural elements. Useful for performance and quantitative analysis.	N/A
Requirement diagram	Show system requirements and their relationships with other elements. Useful for requirements engineering.	N/A

SysML Drawing Summary (II)

Sequence diagram	Show system behavior as interactions between system components. Useful for system analysis and design.	**Sequence diagram**
State Machine diagram	Show system behavior as sequences of states that a component or interaction experience in response to events. Useful for system design and simulation/code generation.	**State Machine diagram**
Use Case diagram	Show system functional requirements as transactions that are meaningful to system users. Useful for specifying functional requirements. (Note potential overlap with Requirement diagrams.)	**Use Case diagram**
Allocation tables* *dynamically derived tables, not really a diagram type	Show various kinds of allocations (e.g., requirement allocation, functional allocation, structural allocation). Useful for facilitating automated verification and validation (V&V) and gap analysis.	N/A

Towards a SysML model of consciousness

Systemic Machine Consciousness?

- The approach based on model-based systems engineering can be described as **explicit model-based, reflective, predictive, adaptive** autonomous systems engineering.

- The key is using the engineering models as self-models

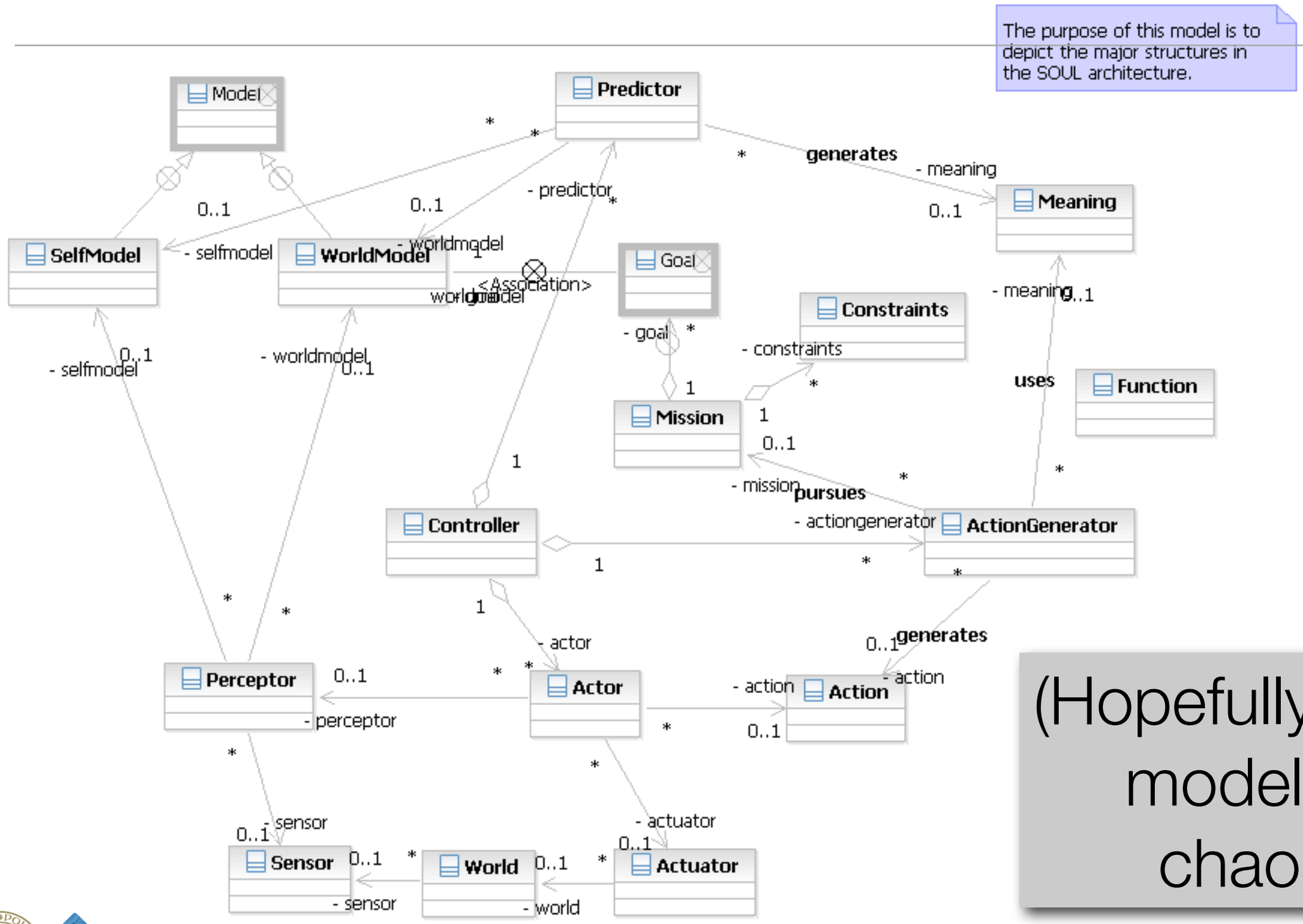
- The major value is that autonomous control based on self-models enables an **increased awareness** that can lead to more **robust performance**

- Provides a road to both 1) expressing an unified theory of consciousness and 2) using it to build machines

An ongoing work

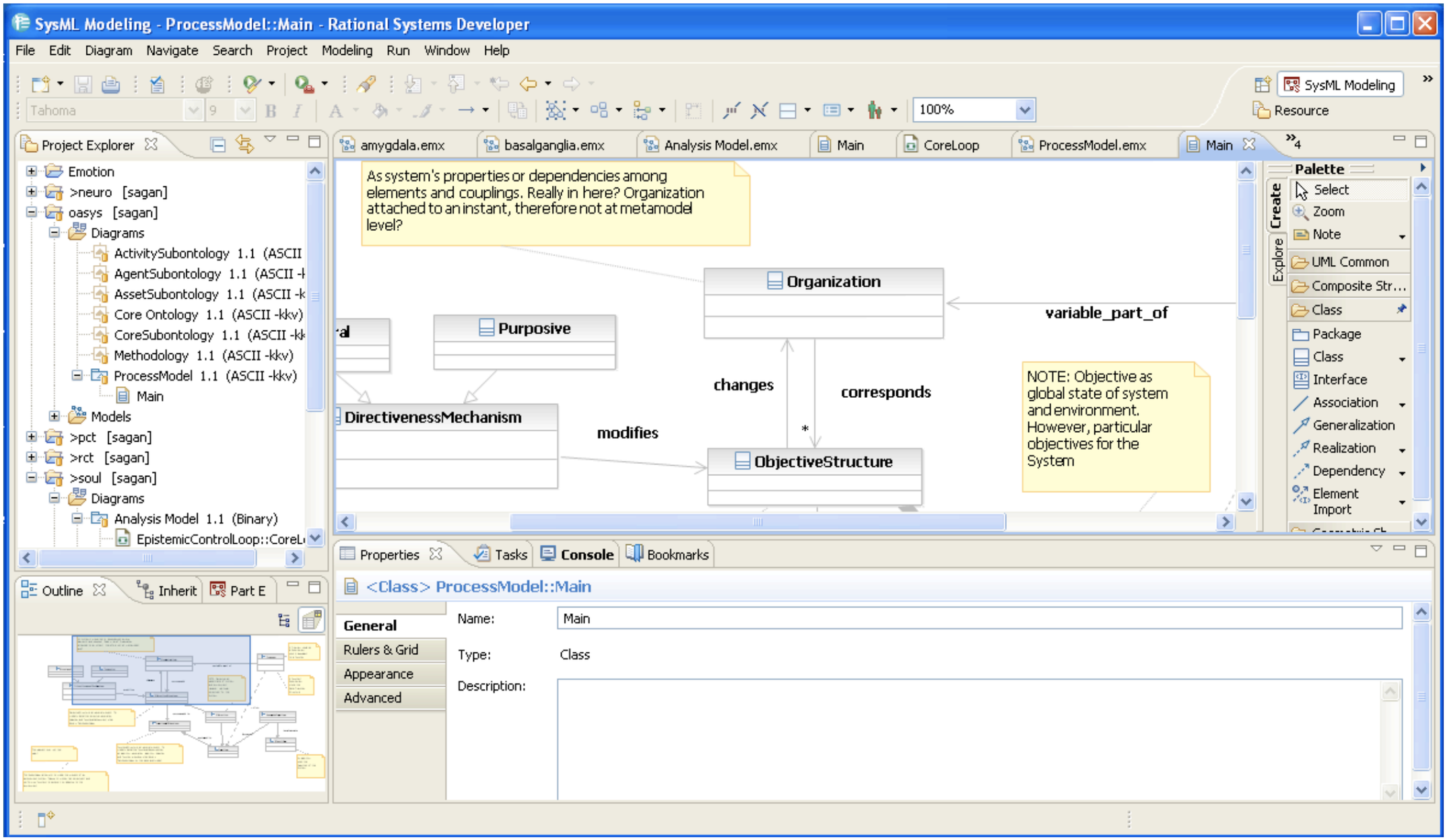
- This is an **ongoing work** being developed in the ICEA/C3 projects
- The focus is the construction of an architecture for self-awareness based on
  - A deep engineering model of self-awareness
  - A model-based construction process based on this engineering model
  - An architecture for model-based autonomous systems exploiting these models

The Present State

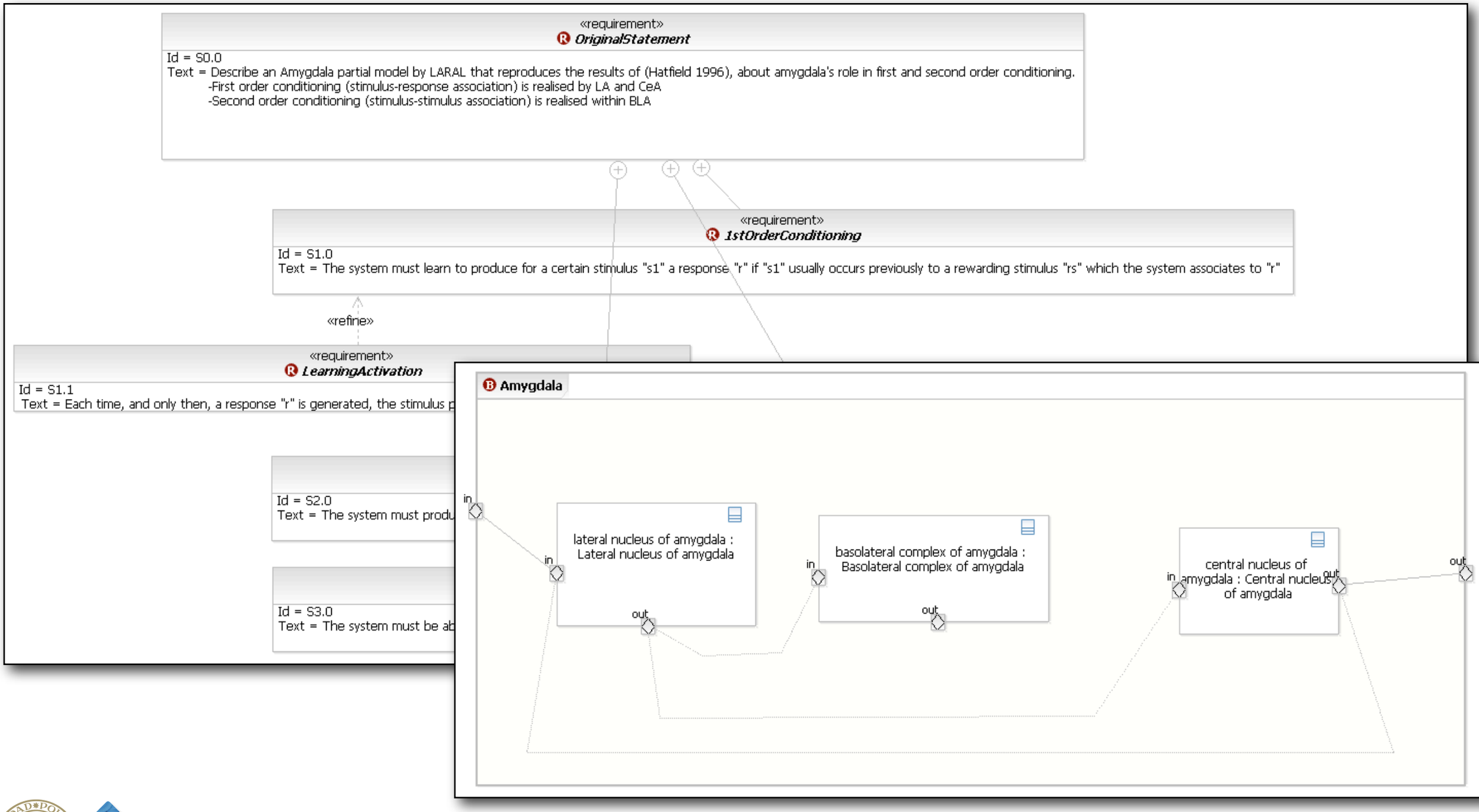


(Hopefully Initial) modelling chaos !

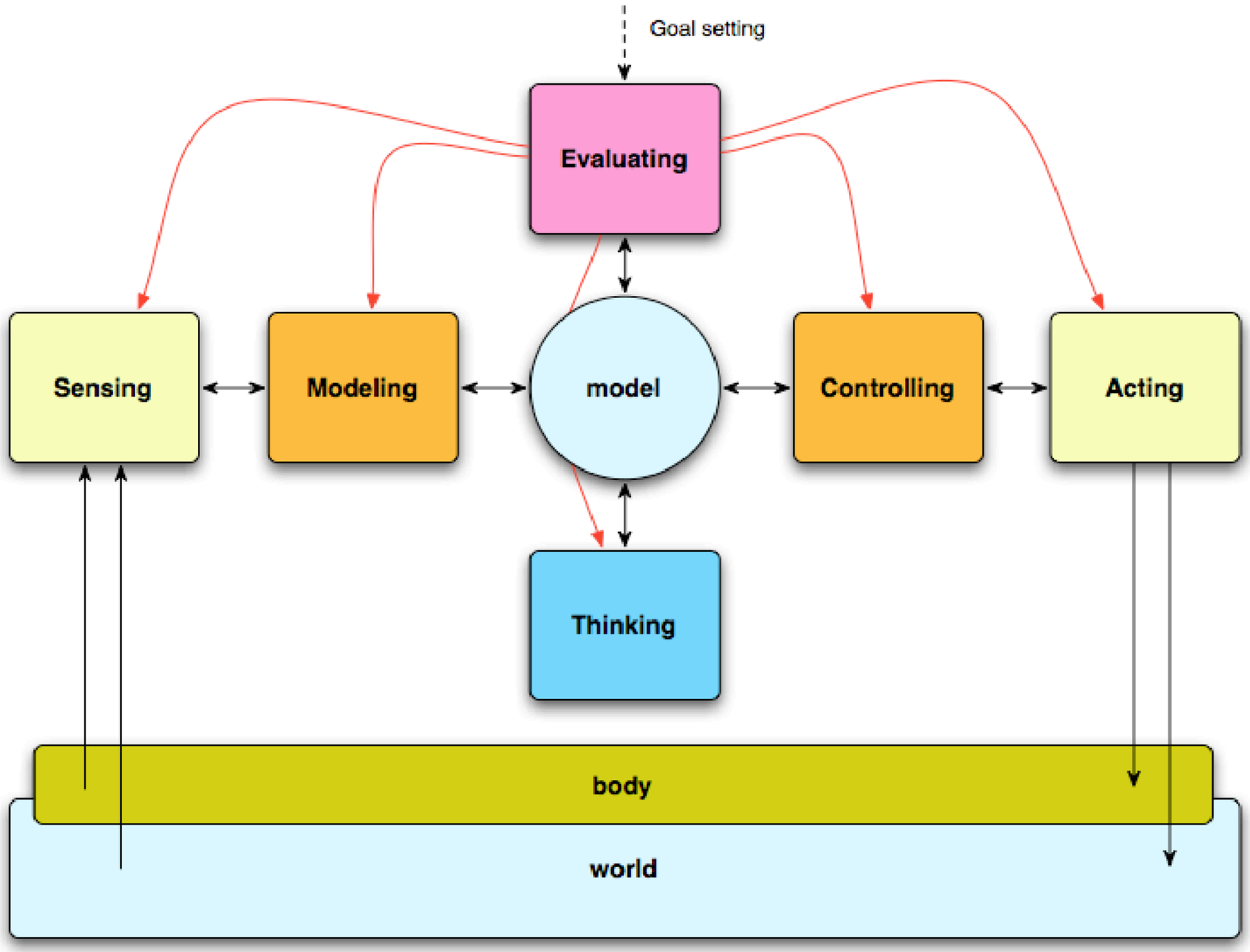
The tooling: Rational Systems Developer



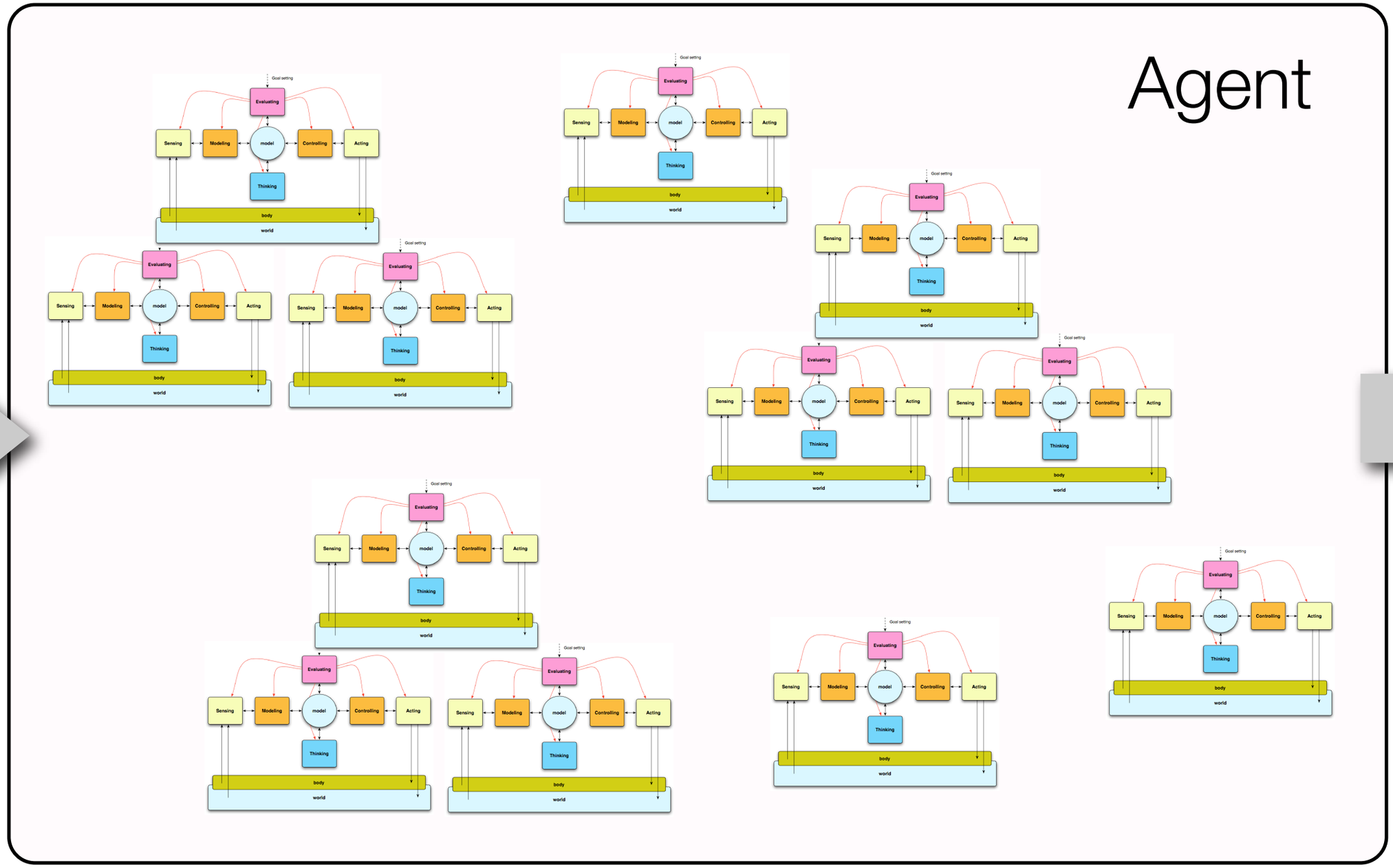
The ongoing modelling effort



The Epistemic Loop

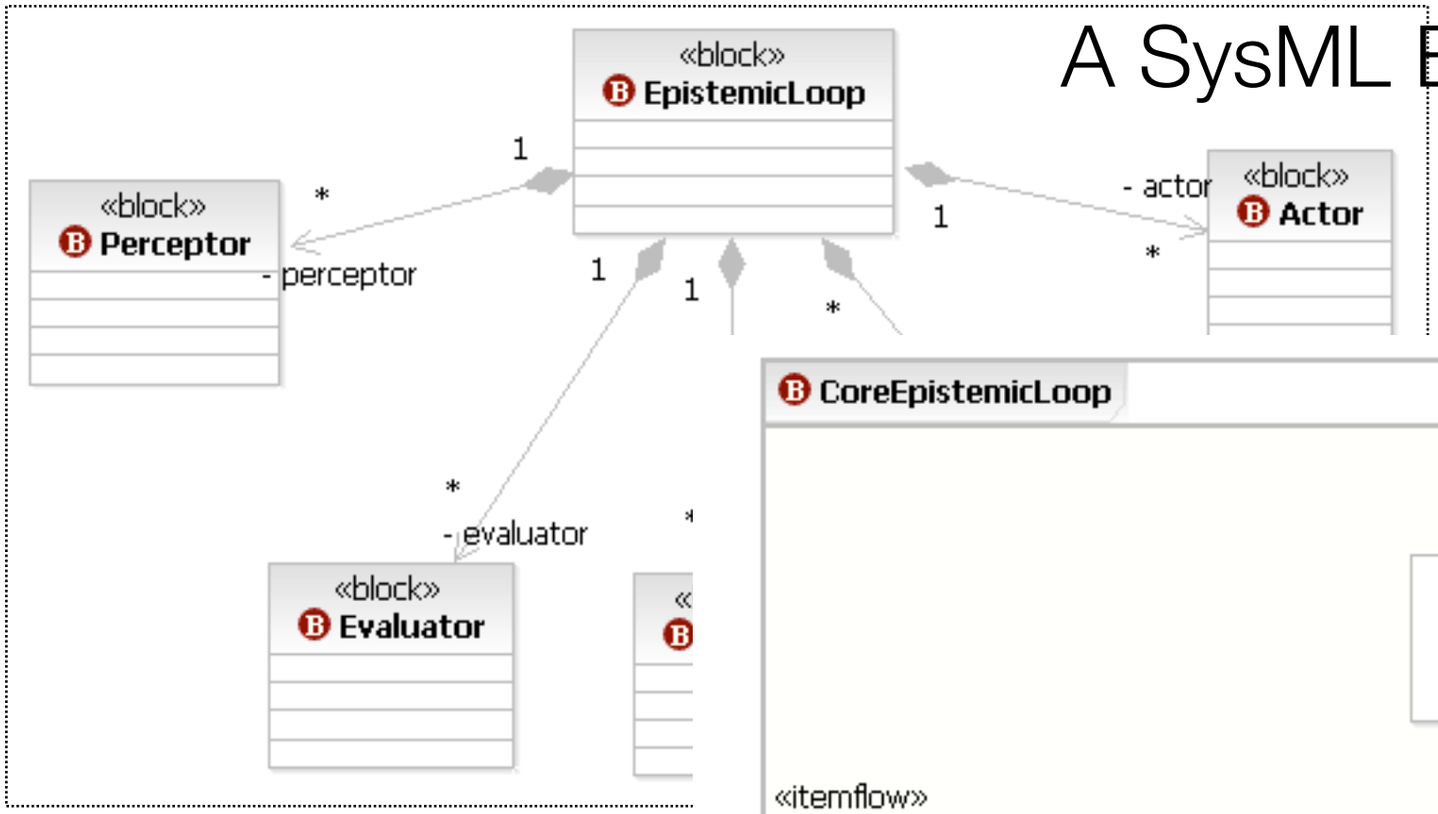


The Distributed Epistemic Loop

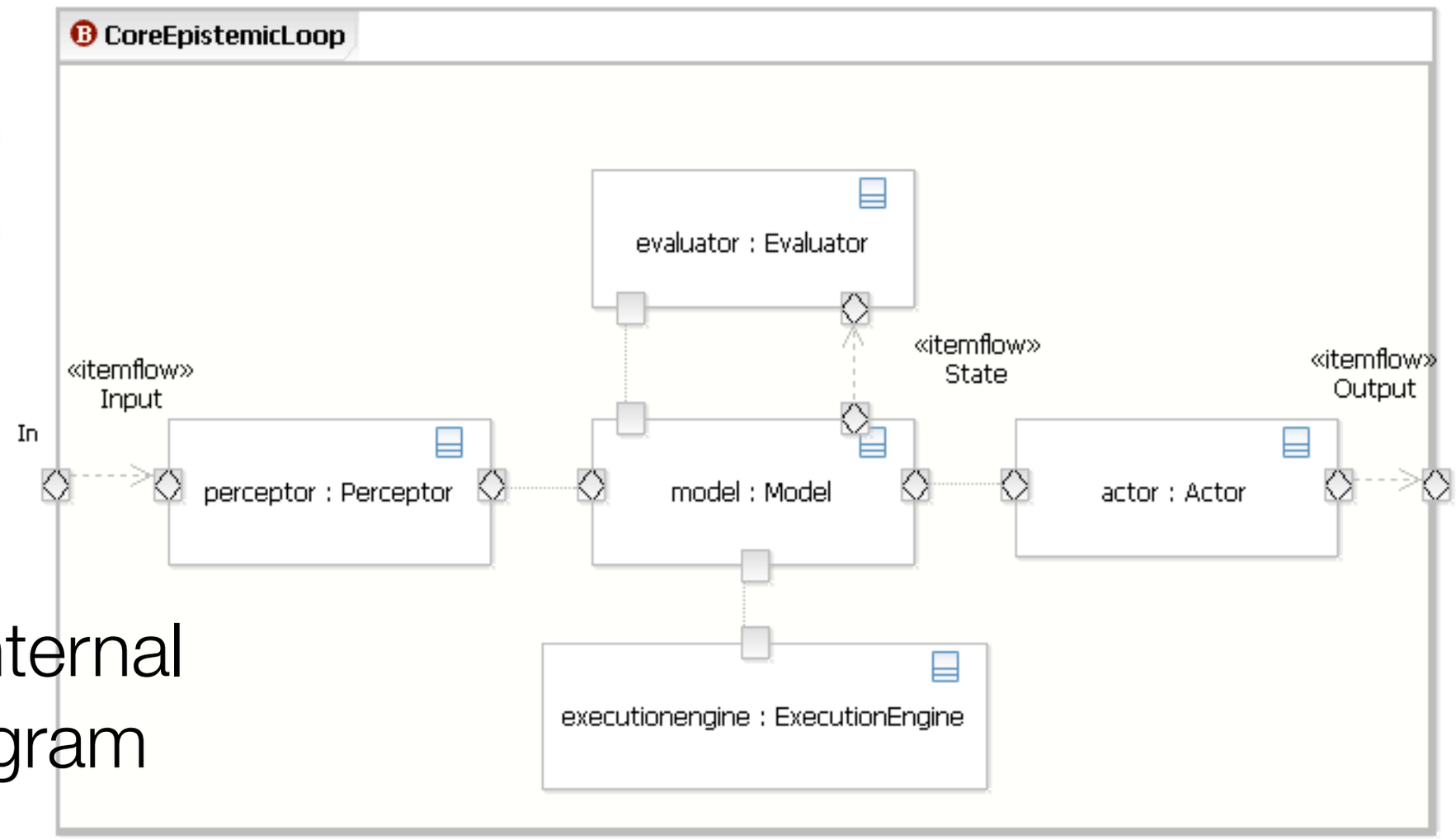


Epistemic Loop models

A SysML Block Definition Diagram



A SysML Internal Block Diagram



Model Structure

- The model is structured in several submodels:
  - The **ASys Ontology** model: a collection of core concepts on autonomous systems from the perspective of General Systems Theory
  - The **SOUL Architecture** model: a model of a self-aware autonomous agent architecture
  - The **Domain** models: models of domain-specific character (now working on **robotics** and **process control**)
  - The application models: concrete, final systems (**PCT**,**RCT**)

Summary

- The core objective of this approach is the development of **Reference Models of Consciousness**

- The SysML model will enable the **expression of the model-based control theory of consciousness in a formal enough language** as to minimise the problem of multiple interpretations

- The existence of the models in this form will also straightforward the development of systems based on this architecture

- The first public release vill be available for **end of 2008**

Some References

- OMG **Systems Modeling Language Specification** - Final Adopted Specification ptc/06-05-04 – 6 July 2006 – Final public version planned in April 2007.
- Laurent Balmelli. **An Overview of the Systems Modeling Language for Products and Systems Development**. Journal of Object Technology. Vol. 6, No. 6, July-August 2007.
- UML SuperStructure specification: OMG
- MDA Specification: OMG
- Systems Engineering: INCOSE

Acknowledgements

- NOKIA / Pentti Haikonen
- European Commission through grant:

  ICEA: Integrating Cognition, Emotion and Autonomy

- Spanish Government through grant:

  C3: Conscious Cognitive Control

- Visit: **www.iceaproject.eu**

ICEA

Sommerhoff, 1990

"the various obstacles that confront those who seek to deal with consciousness in a physical language can be overcome if a strictly methodical approach is followed in which from the start all *analytical concepts are accurately defined* in physical terms."

The End

Systems, models and self-awareness
Toward a SysML model of consciousness

Ricardo Sanz
Carlos Hernández

Autonomous Systems Laboratory
Universidad Politécnica de Madrid

www.aslab.org