Biologically Inspired Cognitive Systems Conference
BICA 2012, Palermo

Towards Architectural Foundations for Cognitive Self-aware Systems

Ricardo Sanz

Universidad Politécnica de Madrid
Autonomous Systems Laboratory

Abstract

•The BICA 2012 conference main purpose is to take a significant step forward towards the BICA Challenge -creating a real-life computational equivalent of the human mind. This challenge apparently calls for a global, multidisciplinary joint effort to develop biologically-inspired dependable agents that perform well enough as to to be fully accepted as autonomous agents by the human society. We say "apparently" because we think that "biologically-inspired" needs to be re-thought due to the mismatch between natural and artificial agent organization and their construction methods: the natural and artificial construction processes. Due to this constructive mismatch and the complexity of the operational requirements of world-deployable machines, the question of dependability becomes a guiding light in the search of the proper architectures of cognitive agents. Models of perception, cognition and action that render self-aware machines will become a critical asset that marks a concrete roadmap to the BICA challenge.

•In this talk we will address a proposal concerning a methodology for extracting universal, domain neutral, architectural design patterns from the analysis of biological cognition. This will render a set of design principles and design patterns oriented towards the construction of better machines. Bio-inspiration cannot be a one step process if we we are going to to build robust, dependable autonomous agents; we must build solid theories first, departing from natural systems, and supporting our designs of artificial ones.

Content

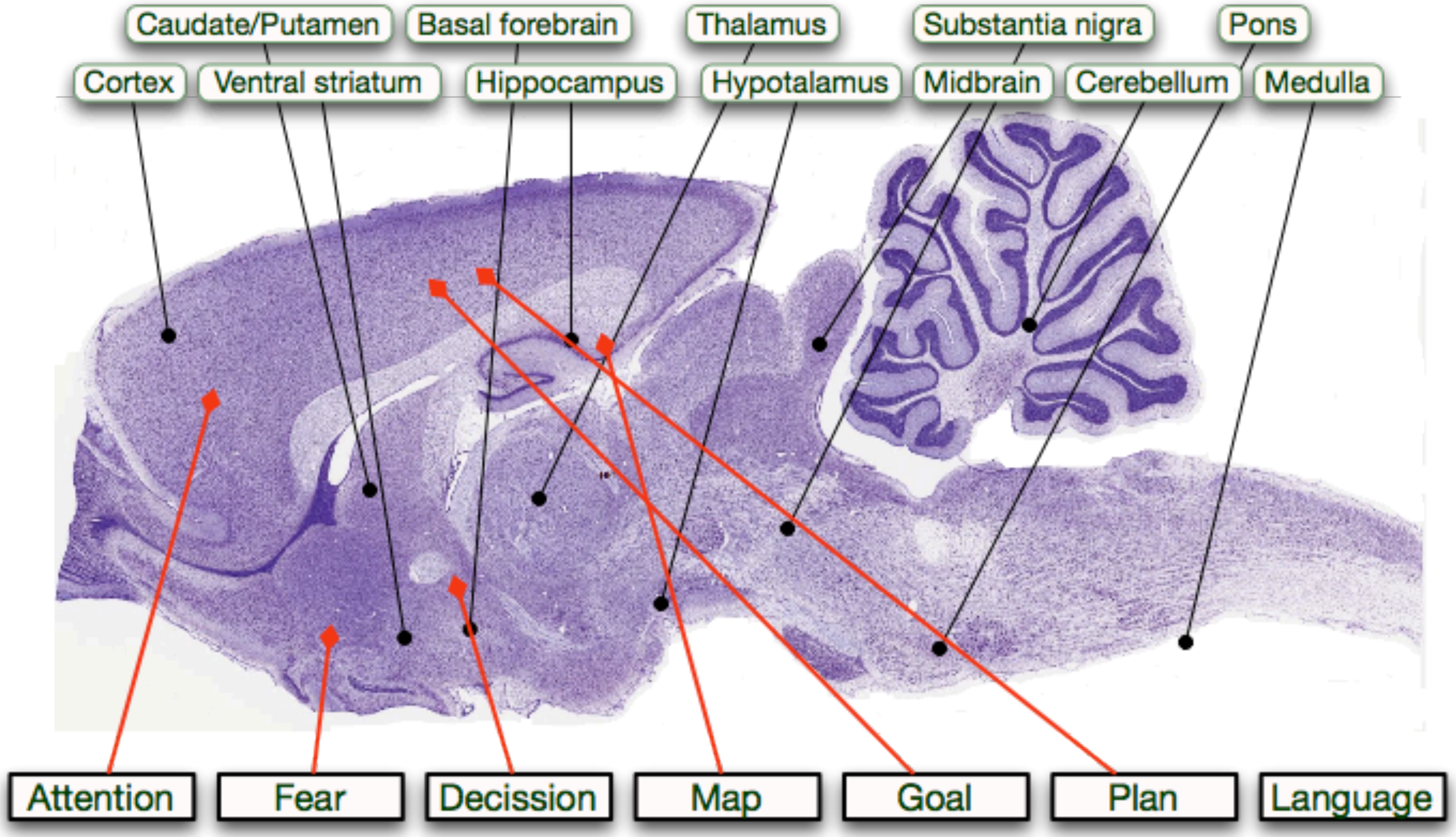
Rethinking
"The Challenge"

The BICA Challenge

- "Creating a real-life computational equivalent of the human mind."
- This challenge apparently calls for a global, multidisciplinary joint effort to develop biologically-inspired dependable agents.
- Agents that perform well enough as to to be fully accepted as autonomous agents by the human society.

... apparently

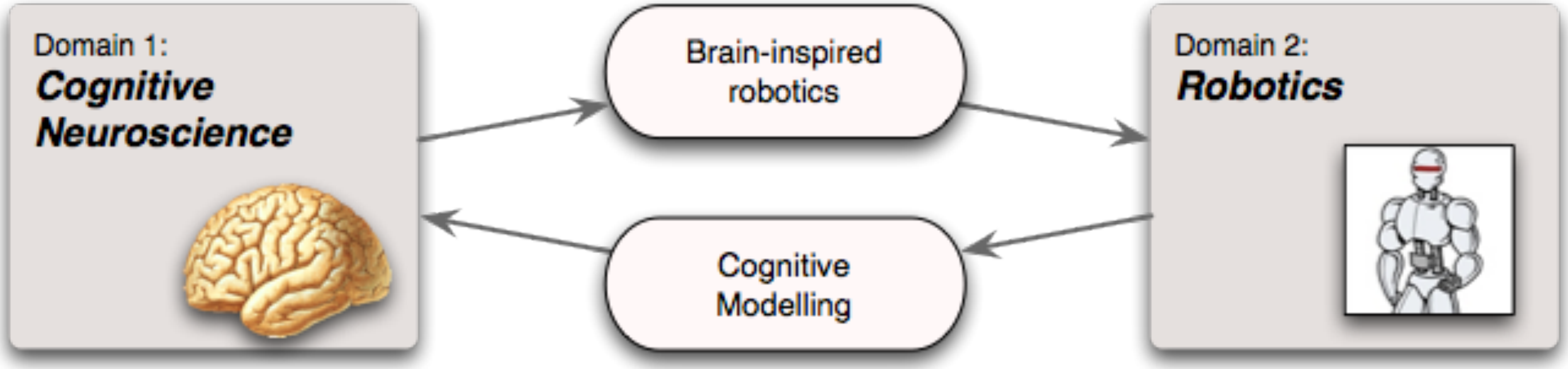
The brain modules trap



Rethinking “biologically inspired”

- "Biologically-inspired" needs to be **re-thought** due to the mismatch between natural and artificial agent organization and their construction methods.
- Due to the complexity of the operational requirements of world-deployable machines, the question of **dependability** becomes a guiding light in the search of the proper architectures of cognitive agents.
- Models of perception, cognition and action that render **self-aware machines** will become a critical asset that marks a concrete roadmap to the BICA challenge.

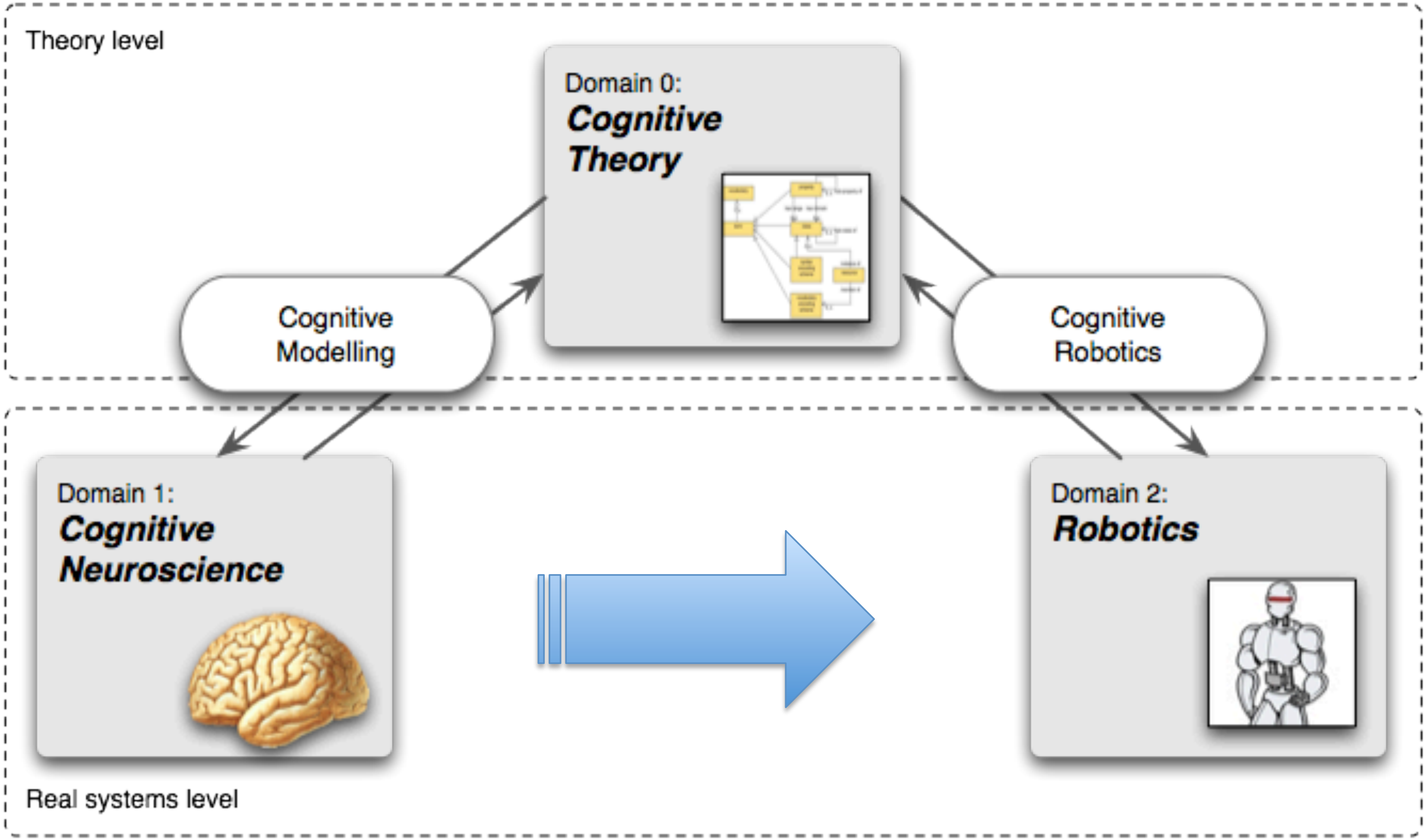
Beware the constructive mismatch



The ASys Strategy

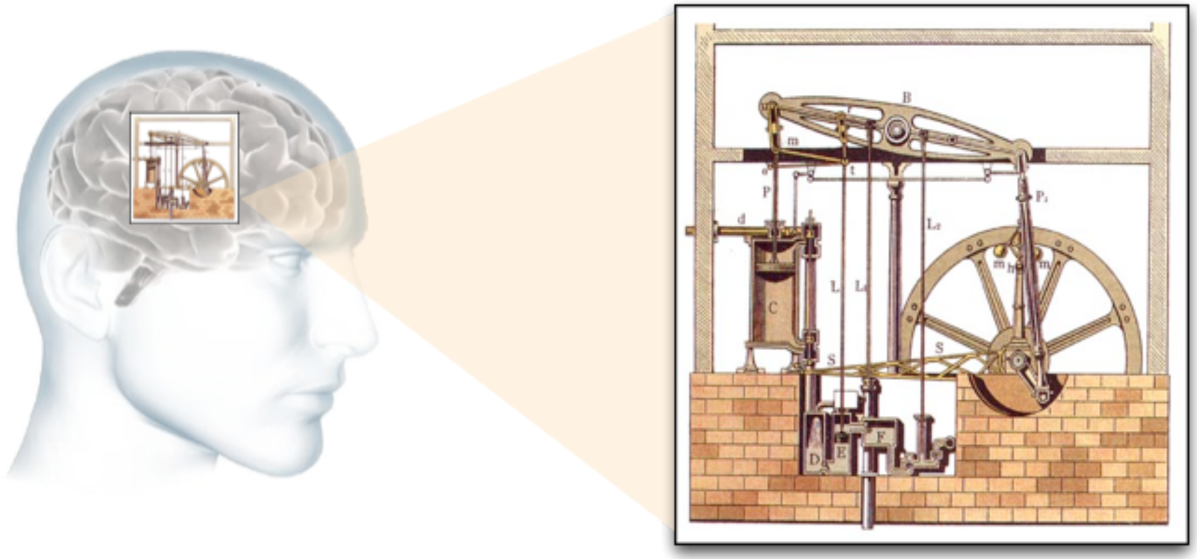
- We need a methodology for extracting universal, domain neutral, **architectural organizations** from the analysis of biological cognition.
- This will render a set of design **principles** and design **patterns** oriented towards the construction of better, more robust machines.
- Bio-inspiration cannot be a one step process; we must build **solid theories** first, departing from natural systems, being general and supporting our designs of artificial agents.

Build theories first



Cognitive Systems
Biological Insights

The agent models the world



Models are not just photographs.

They are **functionally equivalent** dynamical systems.

Cognitive Systems Insights

- Cognition is a control mechanism that enables body efficacy in a complex environment
- This efficacy is achieved internalizing a model of the environment, the agent and their relation

  M(agent) + M(environment) + M(R(Agent,Environment))

- The agent integrates and uses the models to compute actions, anticipating their effects and hence their value
- **Minds** are **model-predictive, integrated controllers**

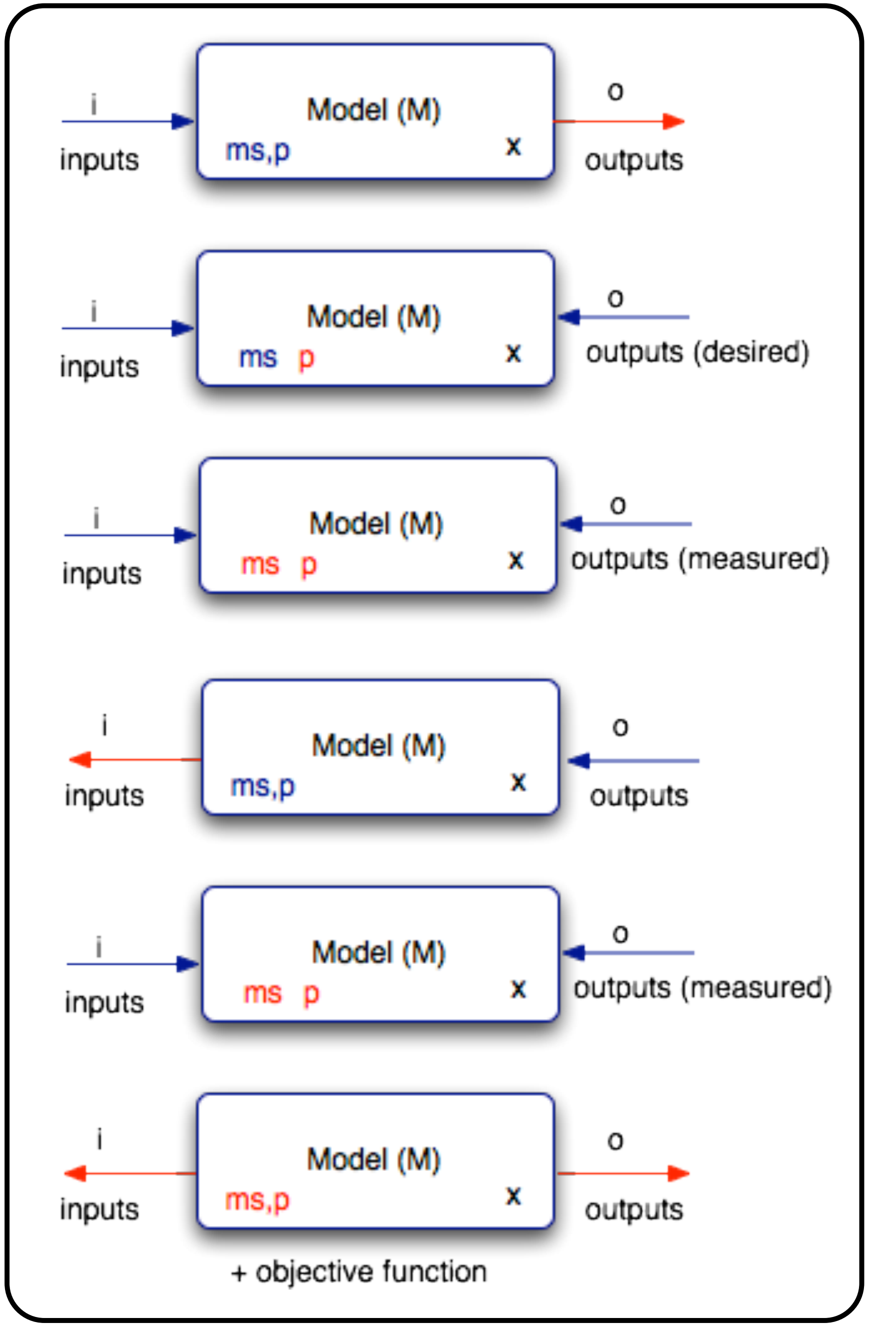
Models and reality

- Using George Box words:

  "**all models are false; some are useful**".

- They are used by means of **execution engines** to:
  - Predict
  - Retrodict
  - Invert (computing control actions)
  - etc.
- Models and engines may be collapsed into single components

Multiple model **uses**



Prediction

Design

Identification

Control

Diagnosis

Optimisation

Universal Cognition Design Principles

- The application of the generalization doctrine to the analysis of the problem of construction of a general cognitive controller has lead us to the formulation of several design principles.
- These principles can be used in two ways:
  - For the construction of reusable assets for a product line approach to robust autonomous systems.
  - As theoretical analyses of natural cognition

Principles

- **Model-based cognition**: A cognitive system exploits models of other systems, of self, in their interaction with them.
- **Model isomorphism**: An embodied, situated, cognitive system is as good performer as it is able to keep its models isomorphic to realities.
- **Anticipatory behavior**: Agents maximise timely performance by means of anticipatory (predictive) models.

Principles

- **Unified cognitive action generation**: Agents 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 realized as the continuous update of the integrated models used by the agent 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**: A self-aware system is continuously generating meanings from continuously updated self-models in a model-based cognitive control architecture.

Self-aware Systems

Self-Aware Autonomous Control Systems

Self-control and model

- [Conant and Ashby 1969]

  Every good controller shall **internalise** a model of the controlled plant.

- Cognitive science reading:
  - Self-representation is necessary for proper behavior.

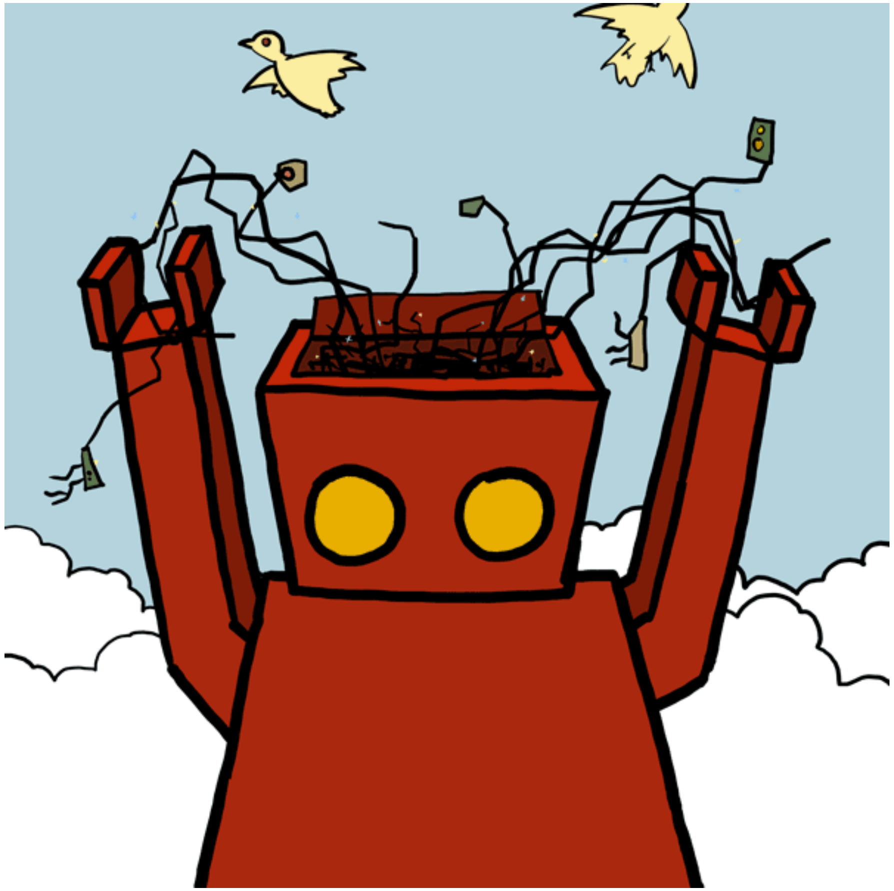
- A system with a self-model can better control its own behavior

Agent access to Self-Body

Leverage **model-based engineering** of the system

Use the
engineering models
as run-time
self-models

The agent becomes
self-aware



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

Architectural Foundations

Architecture and Self

- The system **self-models** shall essentially capture the **system architecture**
- Architecture encompasses the set of significant **design decisions** about the organization of a system
  - **Structure** as elements and their interfaces by which a system is composed
  - **Behavior** as specified in collaborations among those elements
  - **Composition** of these structural and behavioral elements into larger subsystems
  - Architectural **style** that guides this organization

[Booch]

Architecture defined

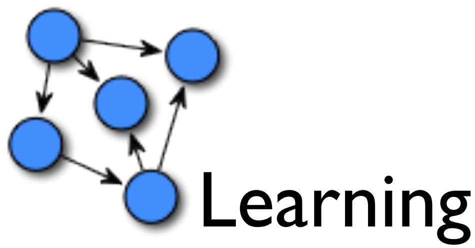
- There is a "Formal Definition" of **architecture**
- IEEE 1471-2000
  - Software architecture is the **fundamental organization** of a system, embodied in its **components**, their **relationships** to each other and the **environment**, and the **principles** governing its design and evolution.
- *ISO/IEC 42010:2007, Systems and Software Engineering -- Recommended practice for architectural description of software-intensive systems.*

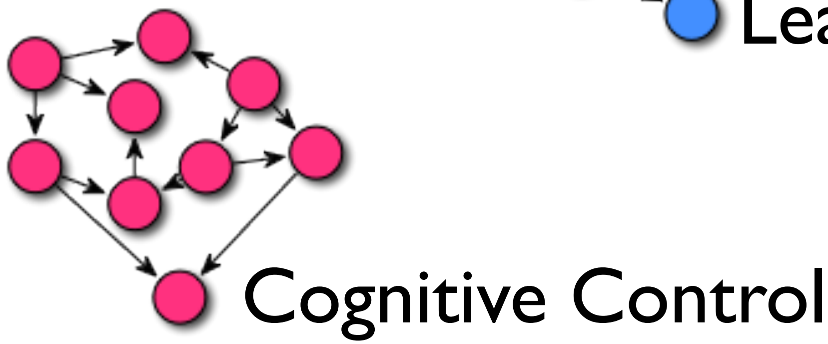
Choose the right description level

- What is the **right description level** ?
  - statement/rule/neuron
  - procedure/kb/nn
  - module ...
  - pattern ...
  - architecture ...
- We choose the **design pattern level** as the proper level for system description, analysis and construction
- The rationale for this election is their **function-centric nature** and **intrinsic composability**

What is a design pattern?

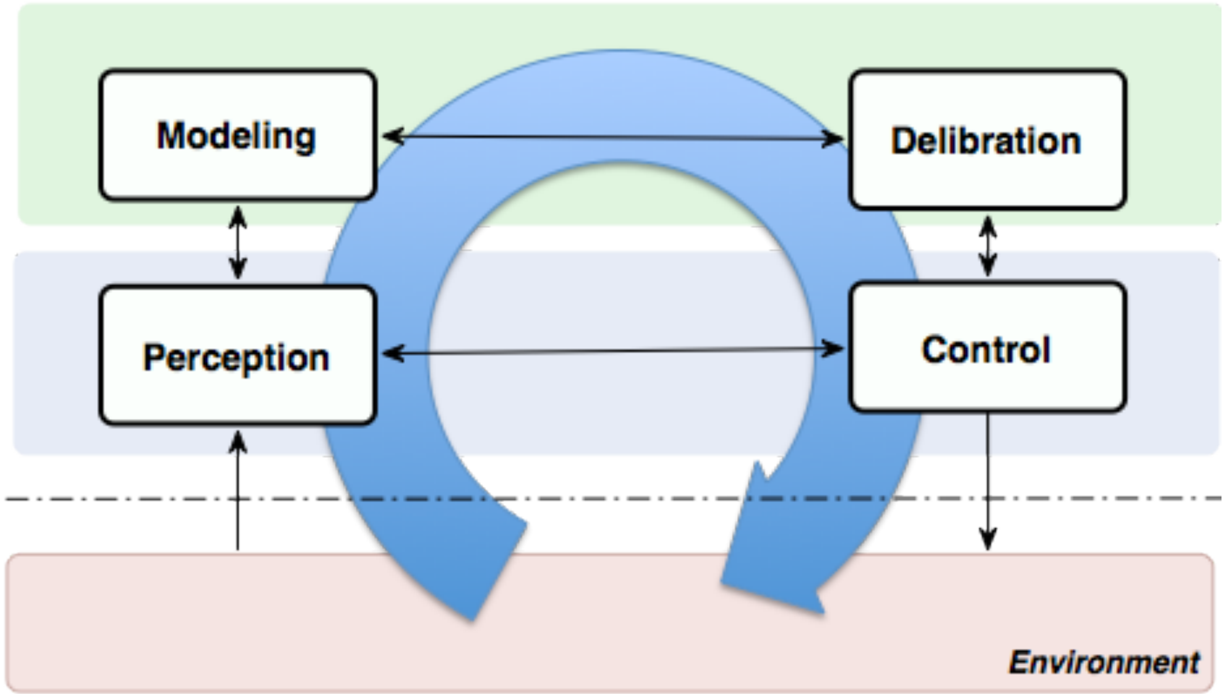
- A mini-architecture, incomplete but useful
- A partial organization of systems components that focus on a specific aspect
- Typically associated to a concrete system function or system property
- Also related to processes





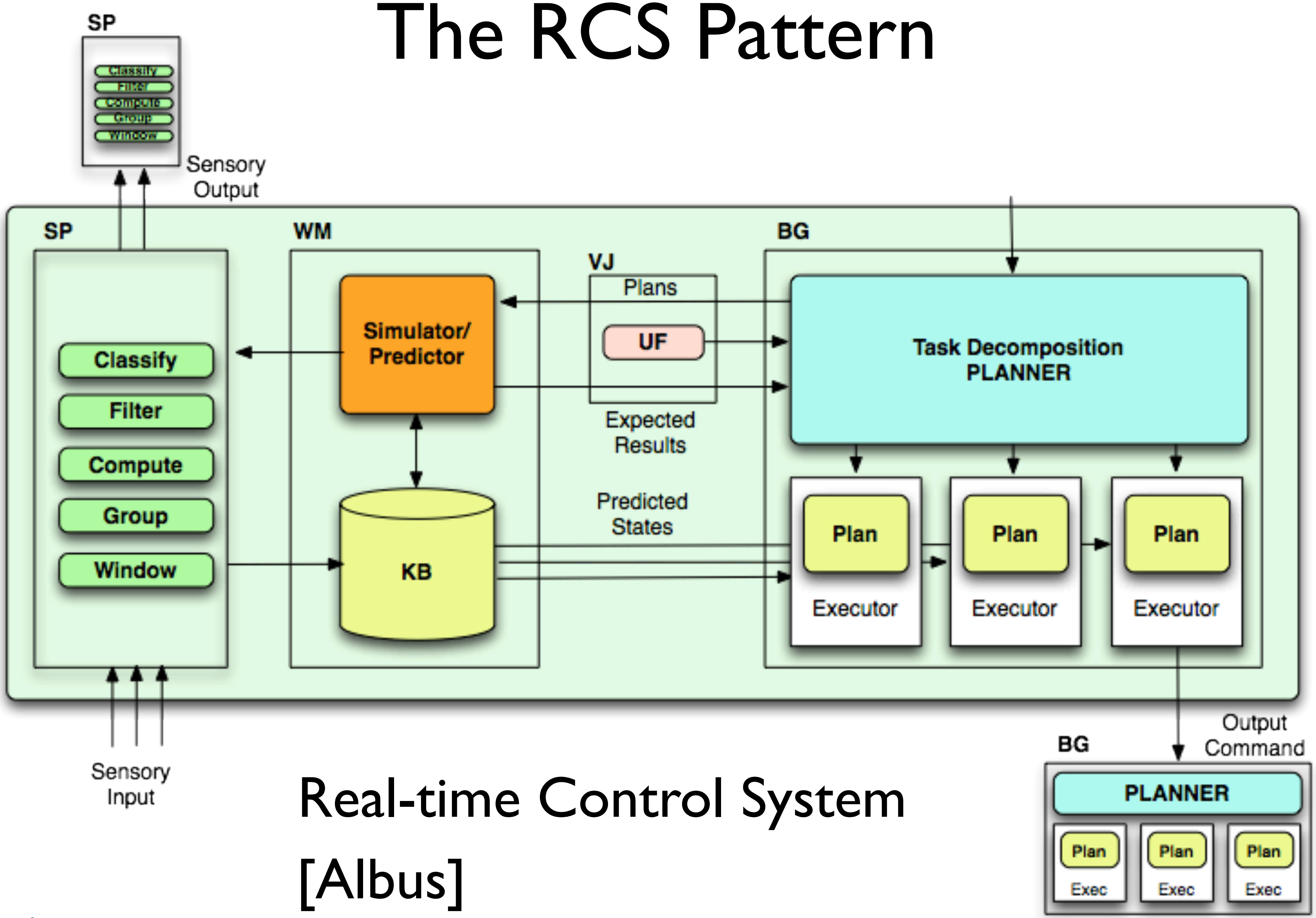
The PEIS Pattern

Physically Embedded Intelligent System



[Saffiotti]

The RCS Pattern



Real-time Control System

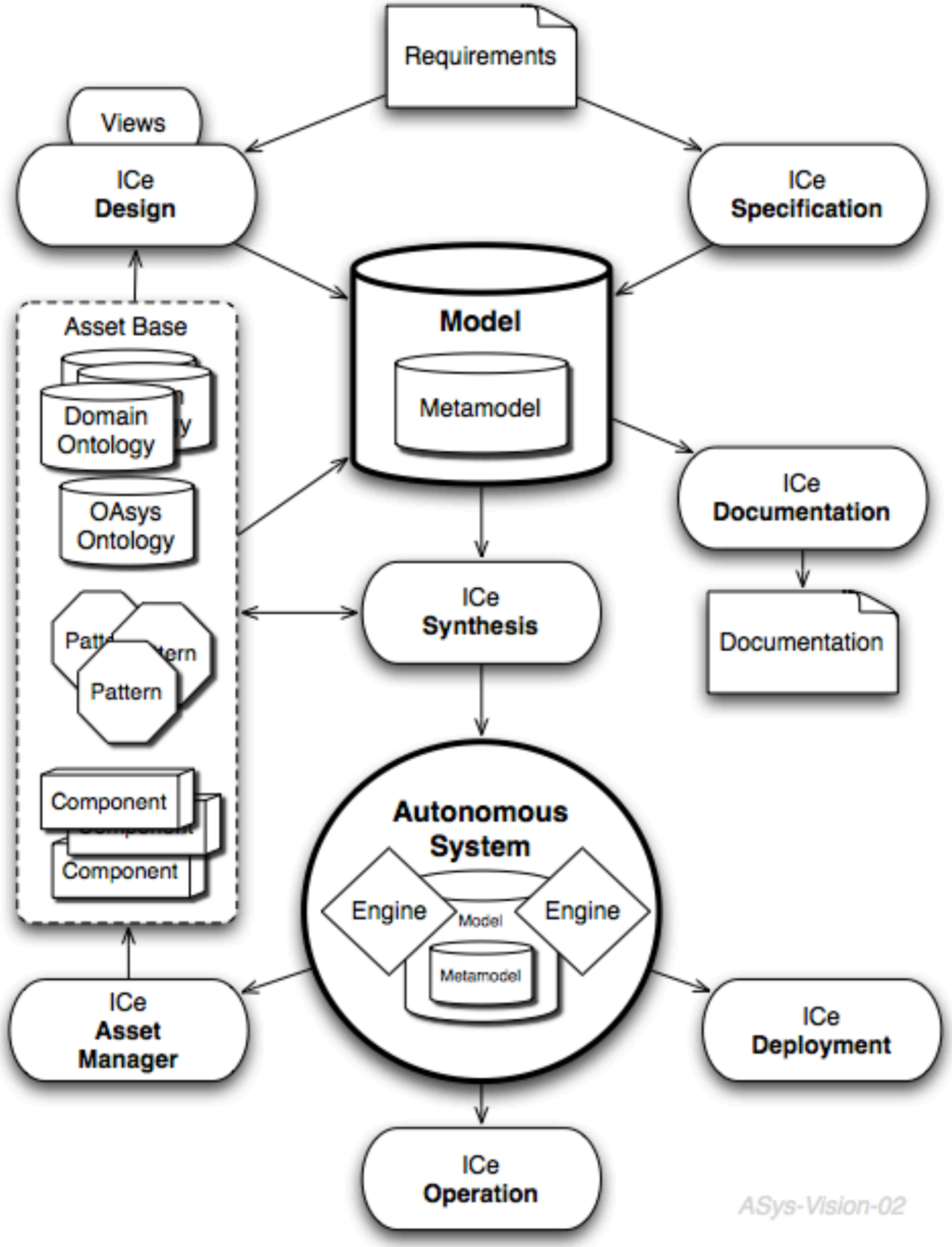
[Albus]

The ASys Patterns

- We try to identify and re-think biological patterns -esp. at the cognitive level- to define general patterns ofr autonomous systems engineering
- Process patterns:
  - Captured in the ASys engineering process
- System patterns
  - To guide the design and organisation of the autonomous system
  - Examples: the ECL, the MC, the DMR ...

The ASys Process Patterns

- Patterns can describe processes
- The **ASys Process**:
- Autonomous self-aware systems can be built by model transformation
- The agent **Synthesis** pattern is pattern-driven and exploits an asset base

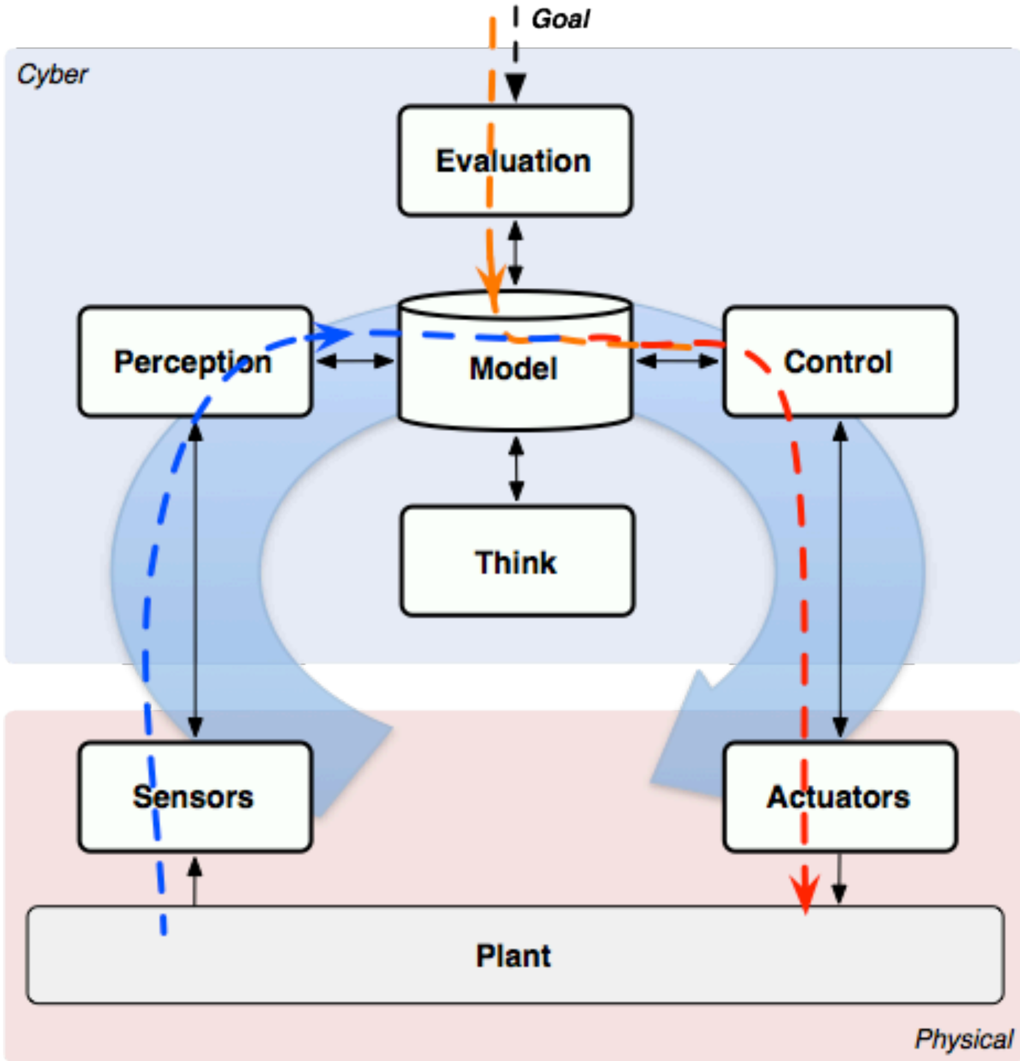


The ECL Pattern

- Knowledge = Model

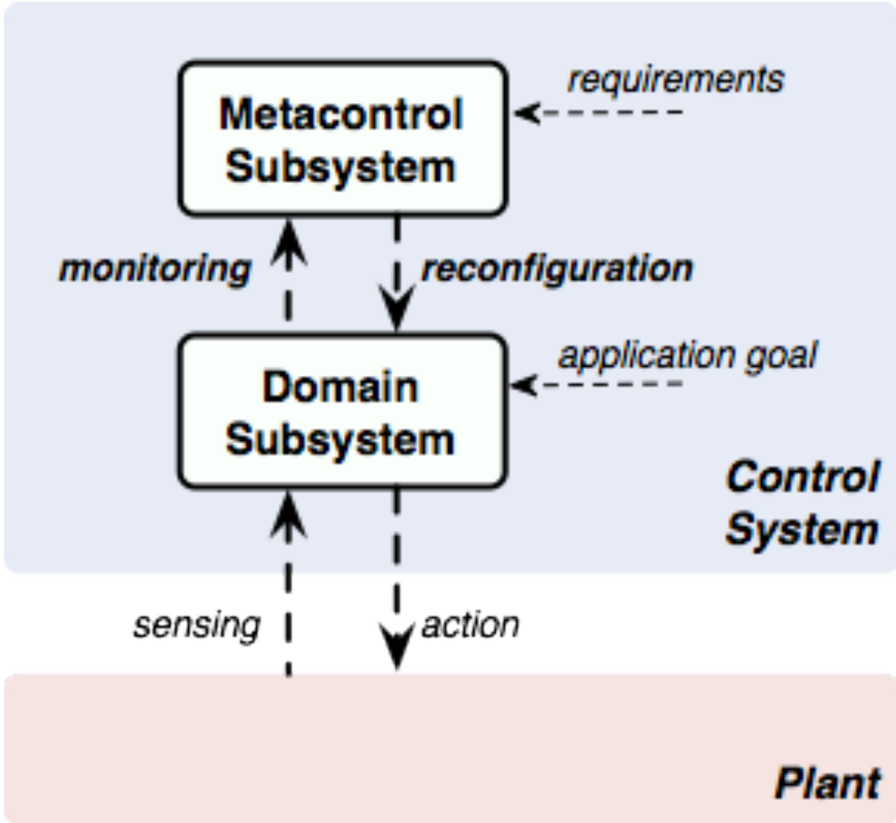
- The Epistemic Control Loop (ECL) pattern generalizes several atom-of-thought patterns (e.g. PEIS and RCS)

- It captures the functional view of model-based cognitive control



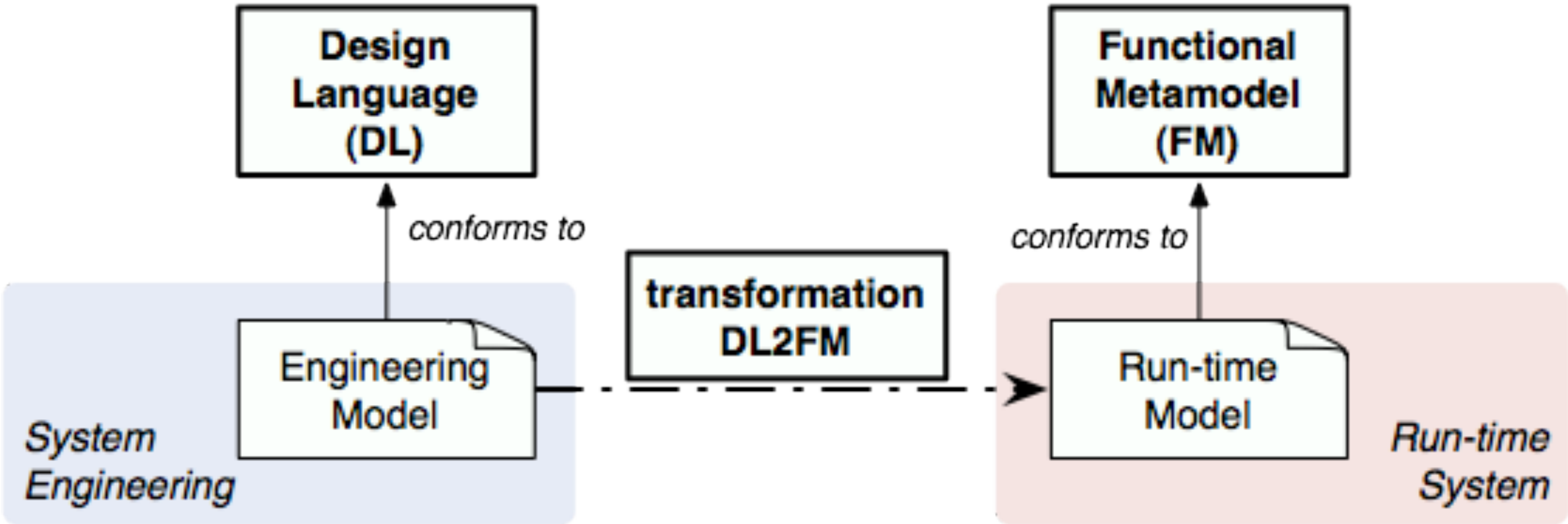
The MC Pattern

- The MetaControl pattern captures the capability of self-observation and control
- The agent sees itself as a behaving -active- entity
- Notice that metacontrol is different from nested control (Layers pattern)
- They differ in the domains of control



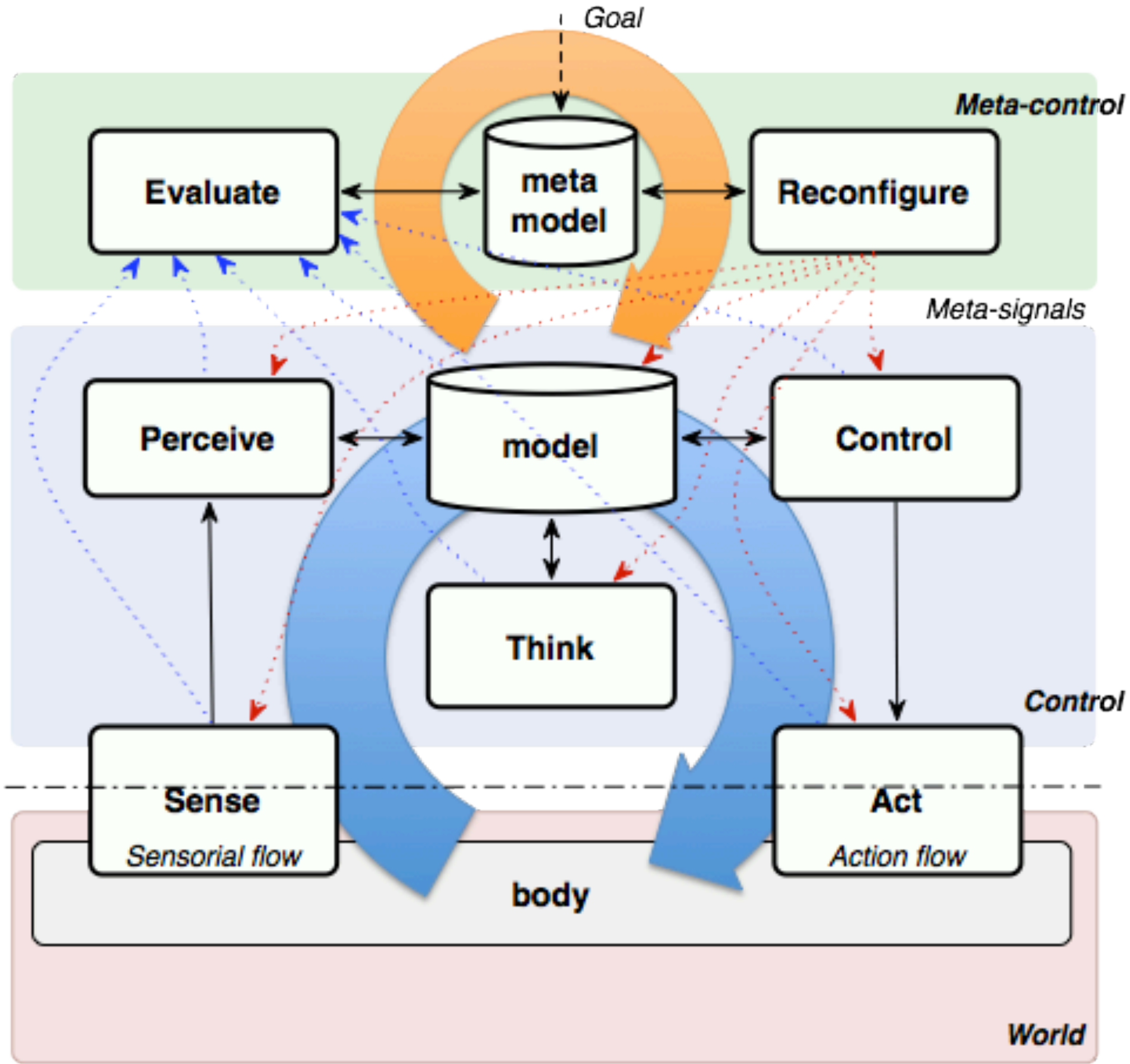
The DMR Pattern

- The Deep Model Reflection (DMR) pattern exploits the deep models used in model-driven systems engineering
- It enables agent deep architectural reflection during runtime



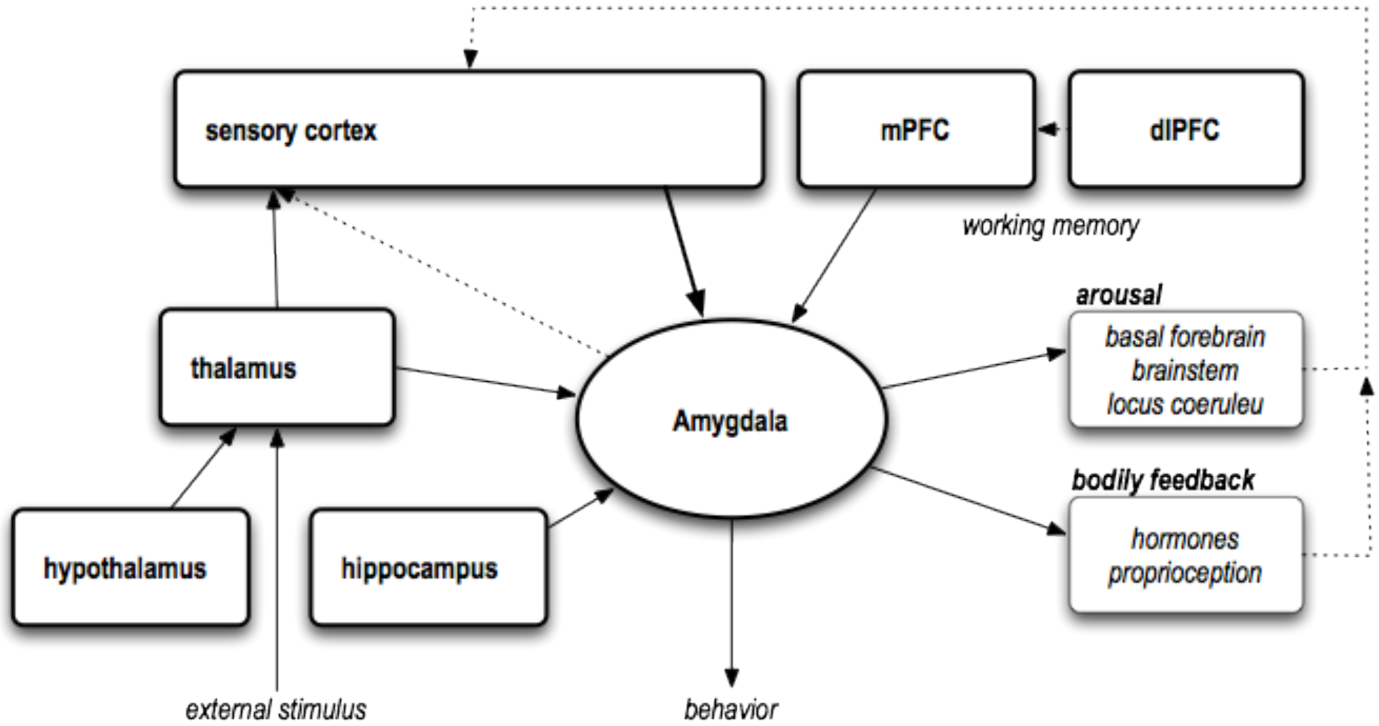
Composing Patterns

- Patterns can be composed to generate complex architectures
- E.g. composing the ECL and the MC
- Implements the MC layer as another ECL



Patterns may be explanatory

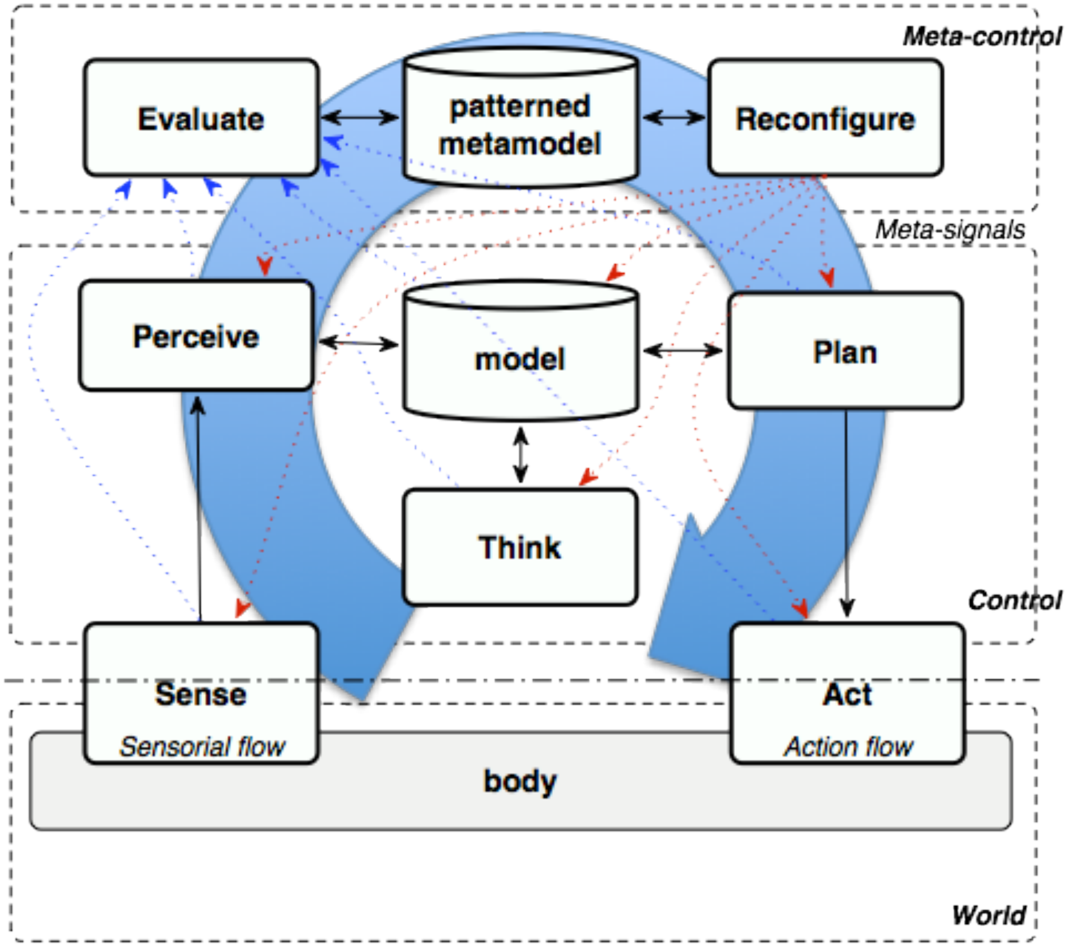
- They can be used to capture, describe and analize biological system organization



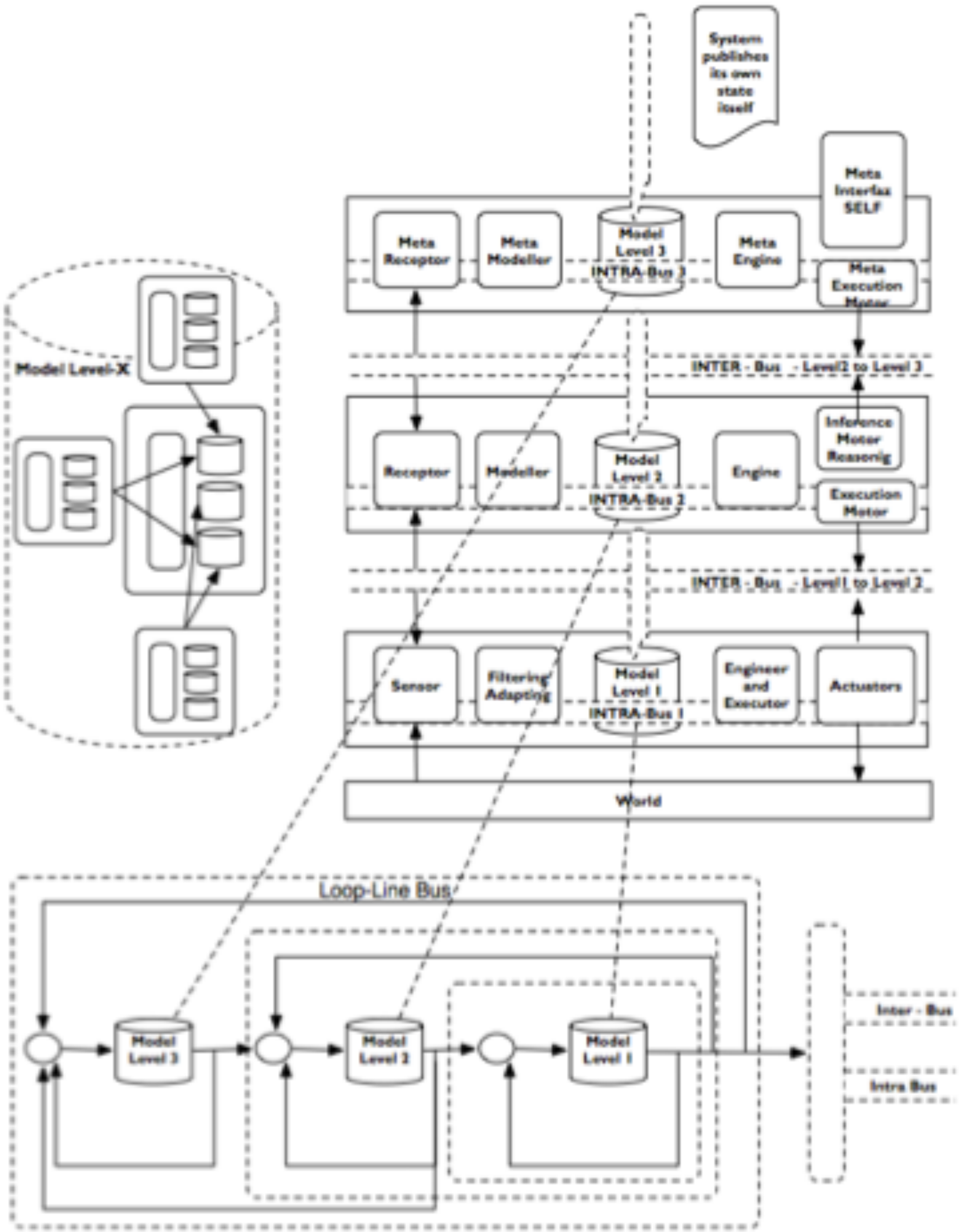
[Arbib-Fellous]

The Emotion Pattern

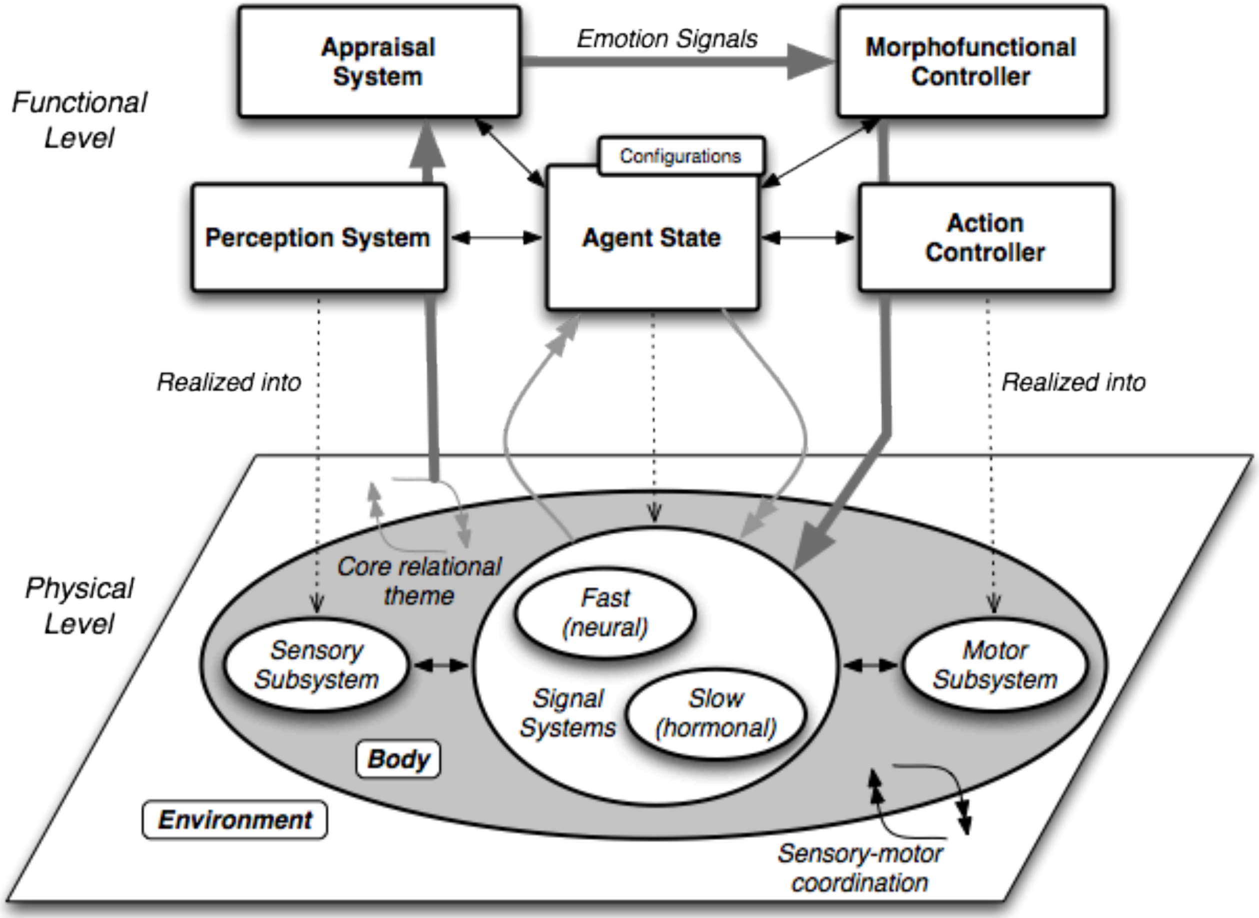
- A double ECL
- An architectural model of emotion
- Emotion as **patterned metacontrol**
- Presented at BICA last year



- A full agent architecture under development
- Exploits the double **ECL** emotional pattern, the **Layers** pattern and multiple **Broker** patterns



Emotion and Function



ECL and MC as explanation of Morphofunctionality

Example Implementation

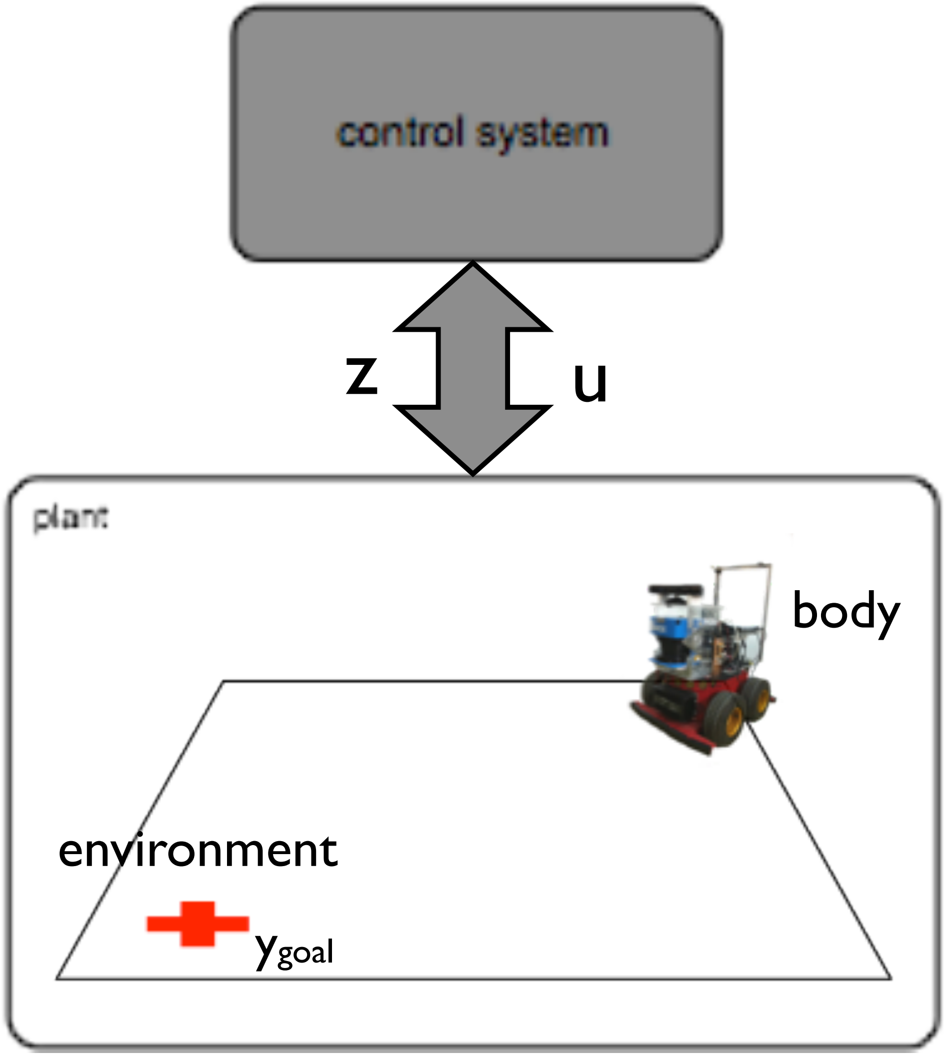
The Operative Mind

- An implemented architecture for robots
- Follow the Self-Awareness Principle
  - System Self-awareness: A system is conscious if it is continuously generating meaning from continuously updated self-models.
- Consciousness as a model-based operating system:
  - Set of services that manage the structure of the control system to adapt it to the objectives. For that a model of the control architecture is used.

Exemplary System: mobile robot

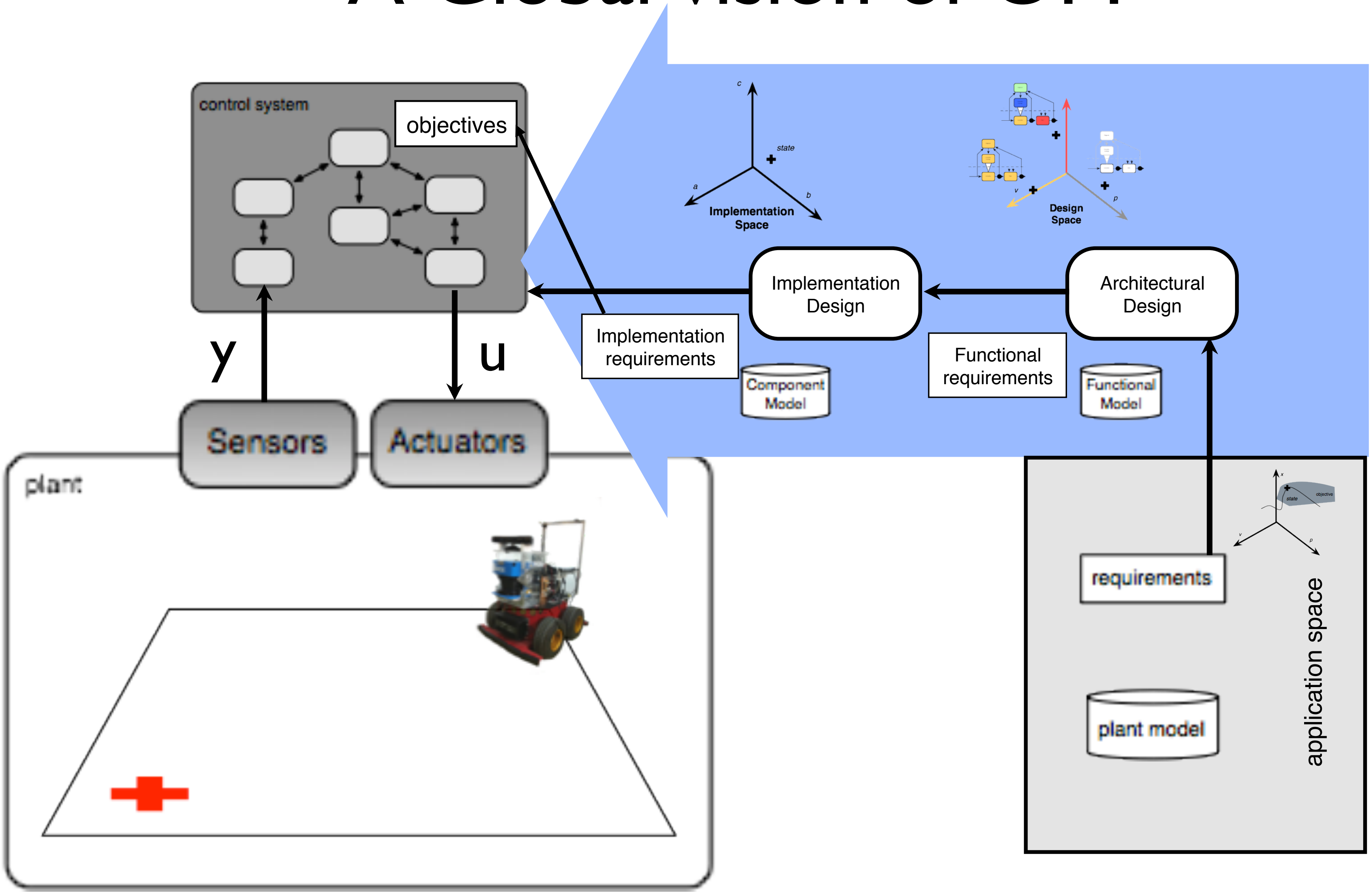
- **System**
  - variable of interest: position
- **Mission**
  - objective: navigate to a destination
- **Environment**
  - laboratory: perform SLAM
- **Realization**
  - ROS stacks

Domain Control System

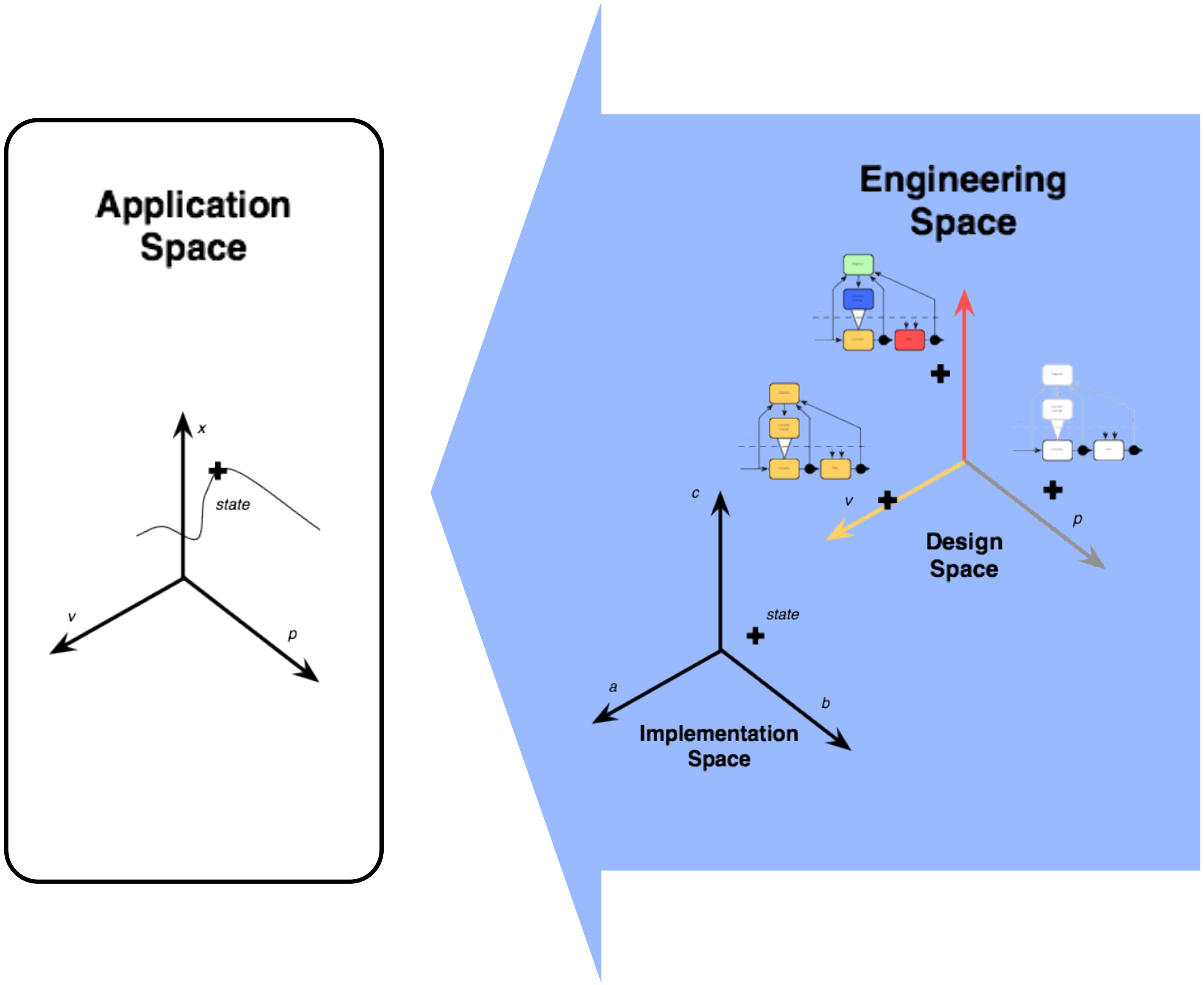


- Controlled system (plant)
  - robot (y)
- action / sensing
  - u = (v,w); z = (odom, laser)
- Obtain a desired behaviour (system's requirements)
  - $y = y_{goal}$
  - constraints on y(t)
  - constraints on u(t)

A Global vision of OM

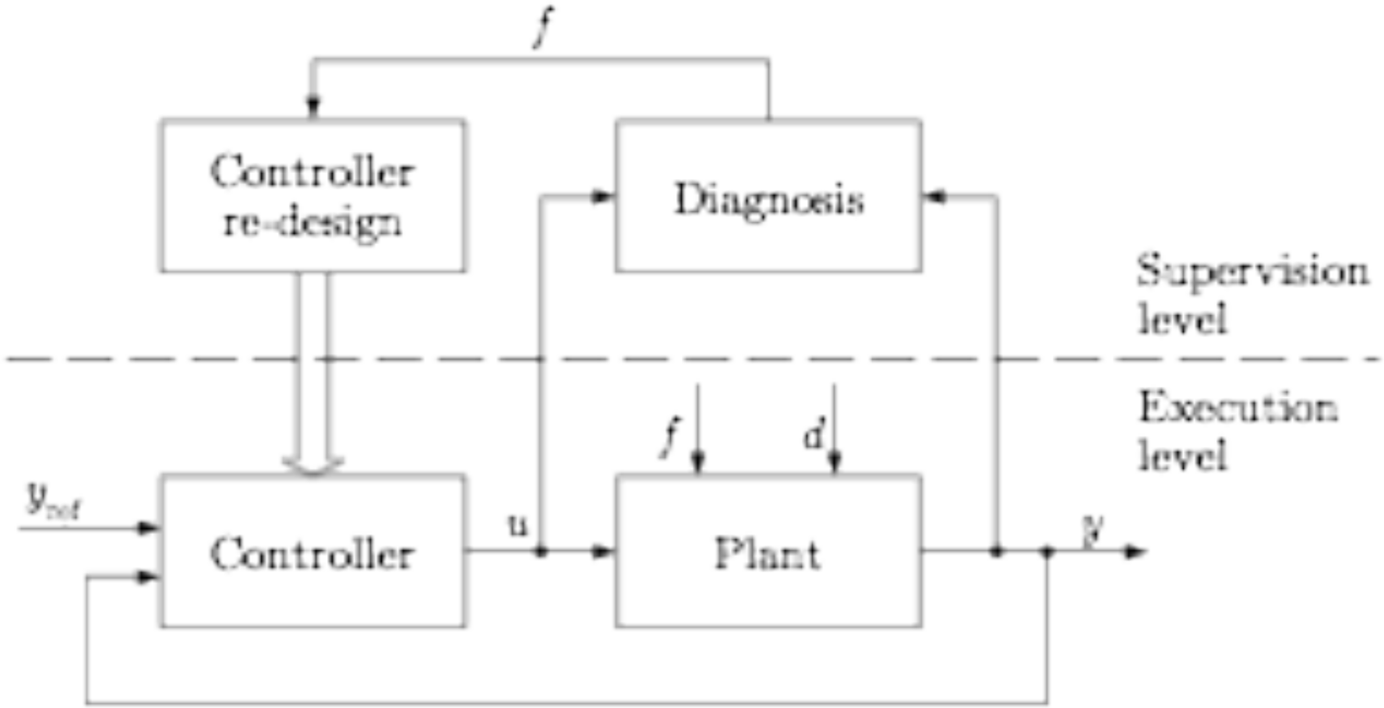


Domains/Spaces involved

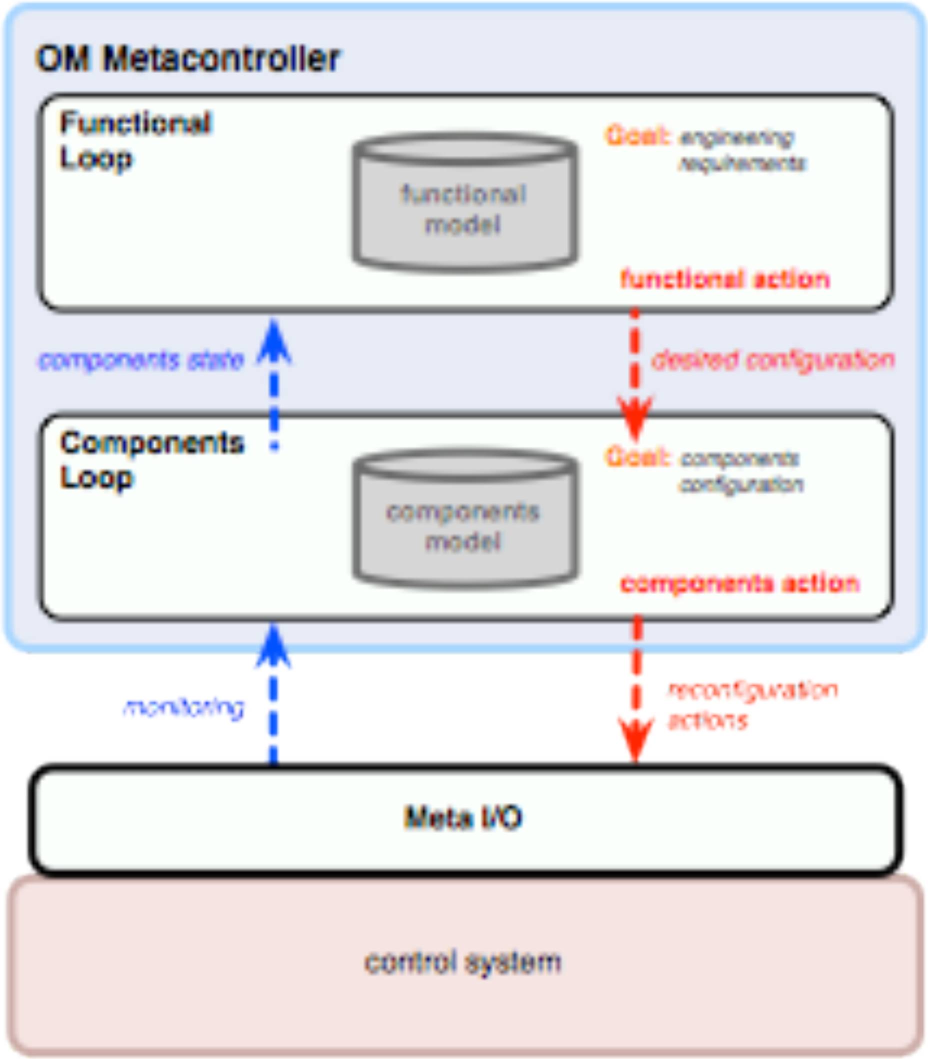


Basic Objective

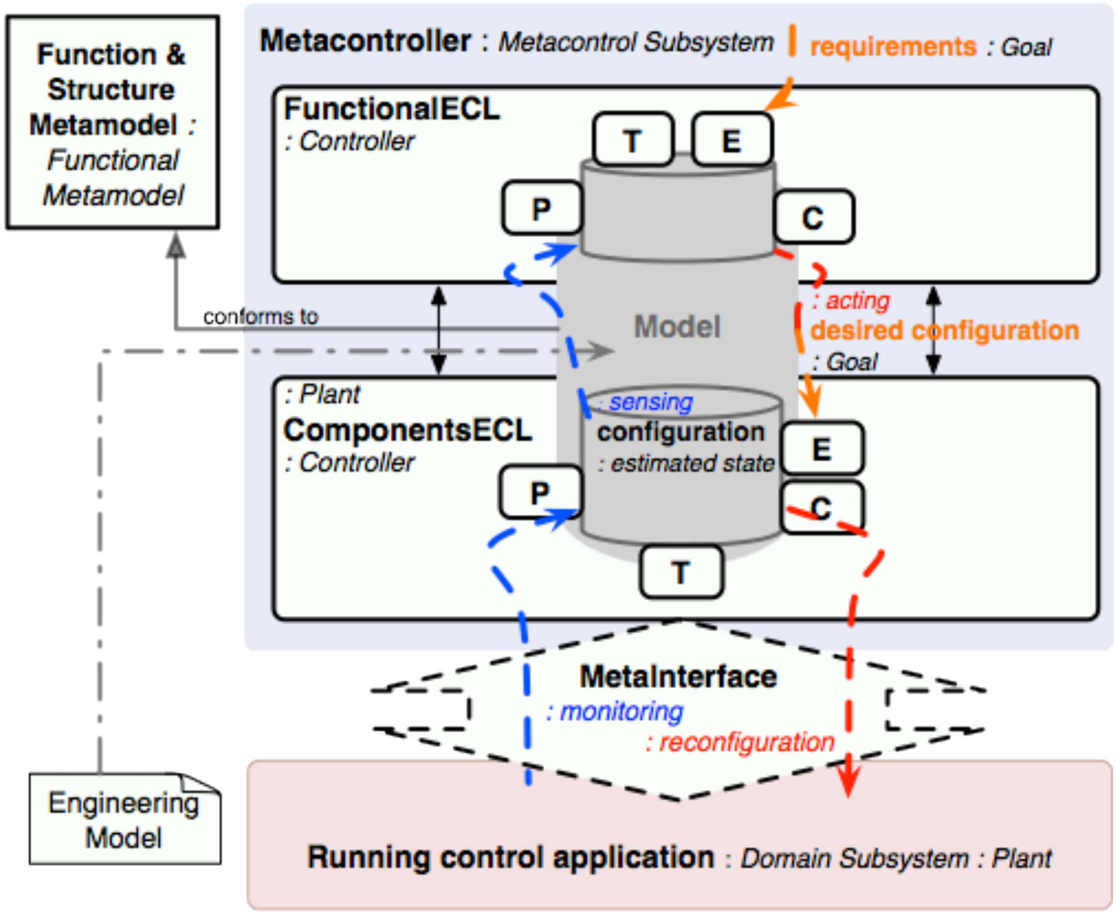
Realize a fault-tolerant control



Overall View of OM

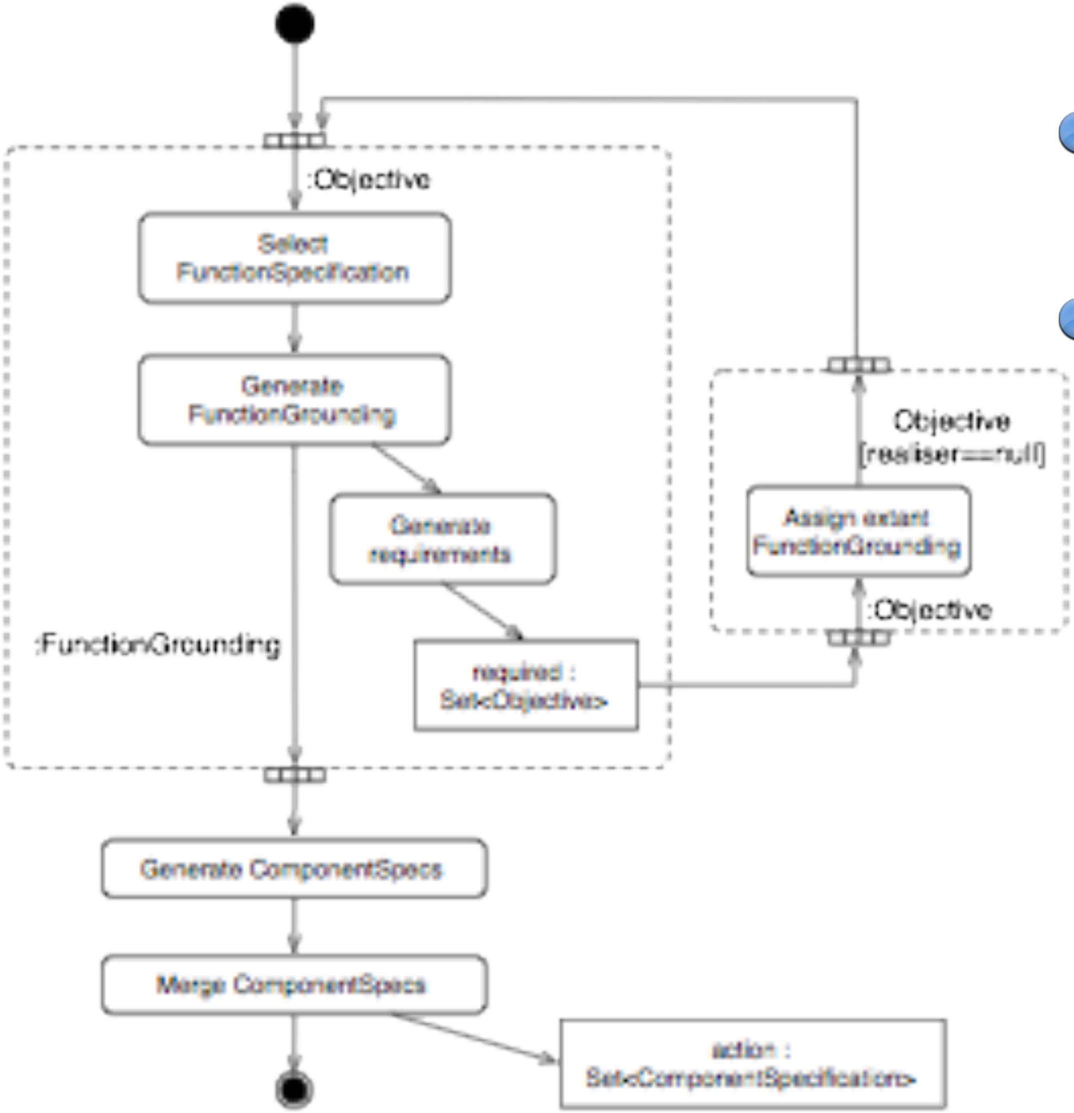


The OM Core Design



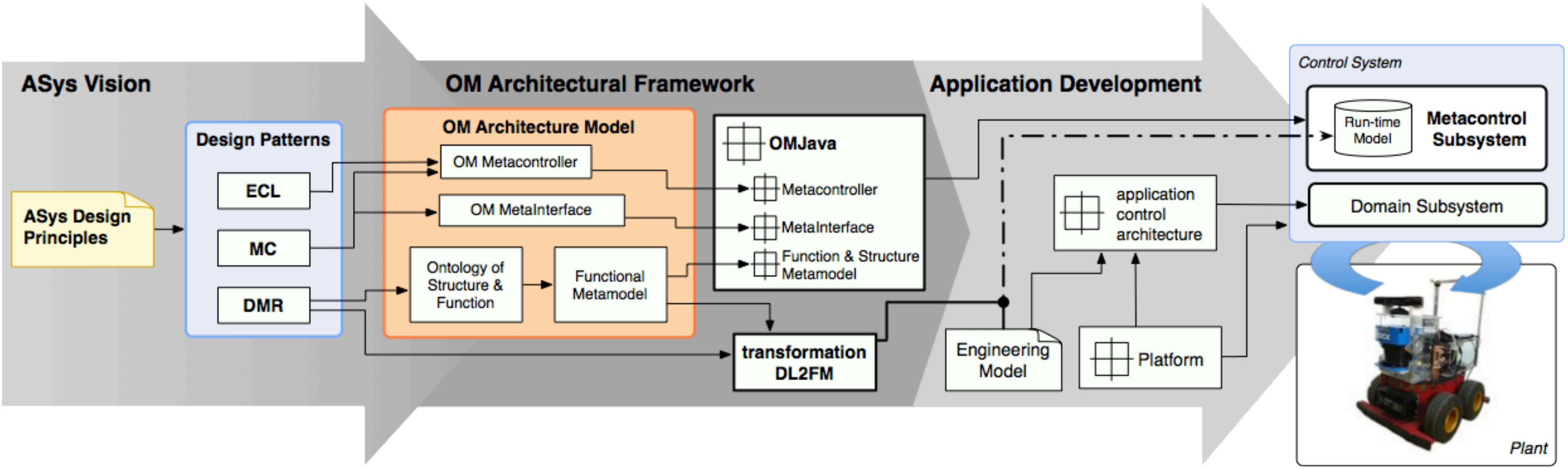
- A double ECL
- ECL at the functional level (how functions attain system goals)
- ECL at the component level (how component organizations provide functions)

OM Functional Control



- Core control process at the functional level
- It exploits at run-time the OM **Function-Structure Metamodel** to reason about component realizations of functions

OM Development Process



Conclusions

Conclusions

- **Models** enable cognition and, in particular, enable self-cognition.
- **Patterns** enable the capture of **architectural aspects of cognitive systems** in a domain independent way.
- Pattern **composition** can be used to generate complex and even complete architectures
- We have presented three patterns that capture basic aspects of cognitive self-awareness: **ECL**, **MC**, **DMR**
- They can be used to **model biological aspects** -e.g. emotion or morphofunctionality- or to **build controllers** -e.g. **OM** for mobile robots.

Thnks

- Students
  - Carlos Hernández
  - Guadalupe Sánchez
  - Carlos Herrera
- Funding
  - European Commission through grants ICEA and HUMANOBS

References

- A. Saffiotti, M. Broxvall, M. Gritti, K. LeBlanc, R. Lundh, J. Rashid, B. Seo, and Y. Cho, "The PEIS-ecology project: vision and results," in Proc of the IEEE/RSJ Int Conf on Intelligent Robots and Systems (IROS), Nice, France, 2008, pp. 2329–2335.
- Albus, J. S. (1995). RCS: A Reference Model Architecture for Intelligent Systems. In AAAI 1995 Spring Symposium on Lessons Learned from Implemented Software Architectures for Physical Agents.

Biologically Inspired Cognitive Systems Conference
BICA 2012 Palermo

Towards Architectural Foundations for Cognitive Self-aware Systems

Ricardo Sanz

Universidad Politécnica de Madrid
Autonomous Systems Laboratory