

*P*latform for *RO*botic modeling and
*T*ransformations for *E*nd-*U*users
and *S*cientific communities

A Robotic Ontology

S. Dhouib¹, J.L. Farges², B. Patin³, S. Stinckwich⁴ (Speaker)

¹CEA, ²ONERA, ³DASSAULT-AVIATION, ⁴GREYC

IEEE ORA group, Tokyo
November, 3, 2013

- Structure
 - Kernel layer
 - Information layer
 - System layer
 - Environment layer
 - Robot layer
 - Humanoid layer
 - Ontology and RobotML meta-model
- Supplementary data

RobotML ontologies' structure

OpenCyc: Upper ontology used for inferences
Wikipedia: References used for description
DEVS as a guideline

ROBOT

It describes robots

... Robot, ManipulatorRobot, MobileRobot,
UnmannedSystem, ...

SYSTEM

what are the possible systems of a robot

... DeviceSystem, ActuatorSystem,
SensorSystem, PowerSystem, ...

MISSION

It describes a mission that can be used by the components

... MissionObjective, MissionType, ...

EXPERIMENT

It describes experiments process

... Problem, Solution,
Assessment...

SIMULATION

It describes how system can live inside software, hardware, etc

... Simulation, Probe,
providesLifeTo, ...

ENVIRONMENT

It describes where the robots evolve and defines the interactions propagation

... Atmosphere, LandSurface, Building, ...

INFORMATION

It describes data pieces that can be used in different parts / different modules

... algorithms, software modules, ...

KERNEL

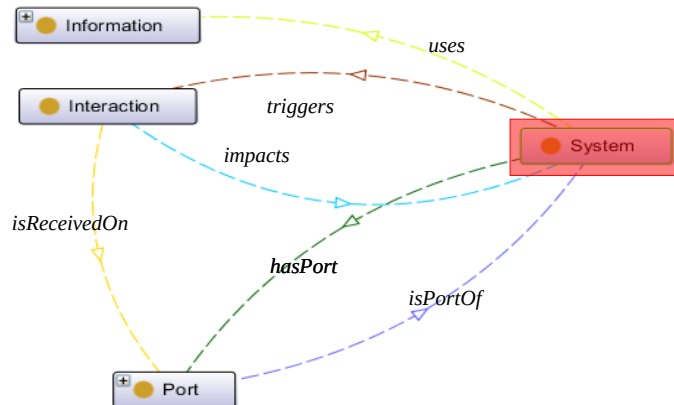
Generic concepts necessary to describe a scenario

... PhysicalObject, Interaction, System...

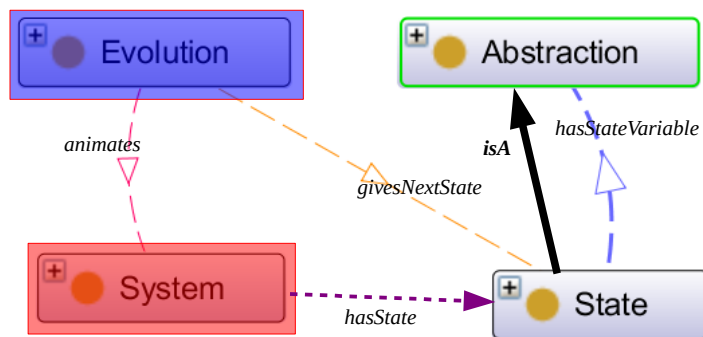
ONTOLOGIES
INTRODUCED

Intellectual Property Rights are governed by PROTEUS Contract and PROTEUS consortium agreements. Before using, reproducing, modifying or disclosing the information contained in that document, it is under the user responsibility to check its rights and if mandatory ask to the Intellectual Property Owner the authorisation to use, reproduce, modify or disclose the information.

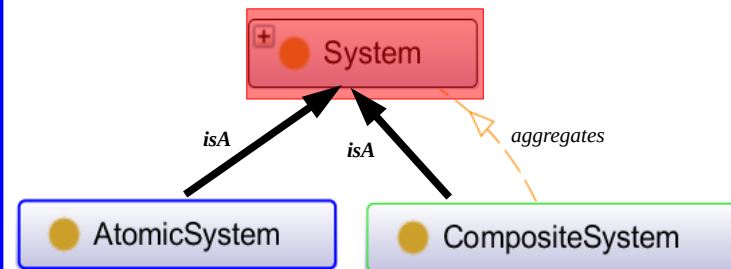
System boundaries *Capability to get / send info*



System evolution *Capability to live*



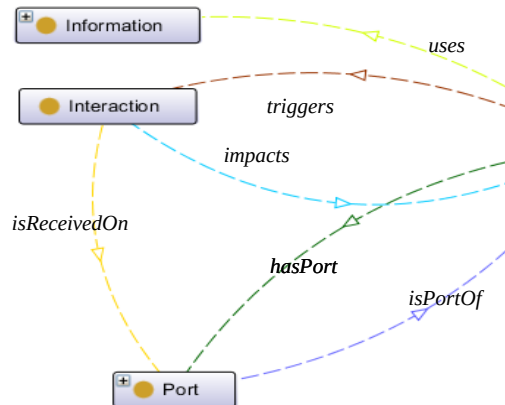
Composition *Capability to build imbricated blocs*



KERNEL – 1 / 2

Many concepts are associated to descriptions and semantic properties

System boundaries *Capability to get / send info*



Annotations

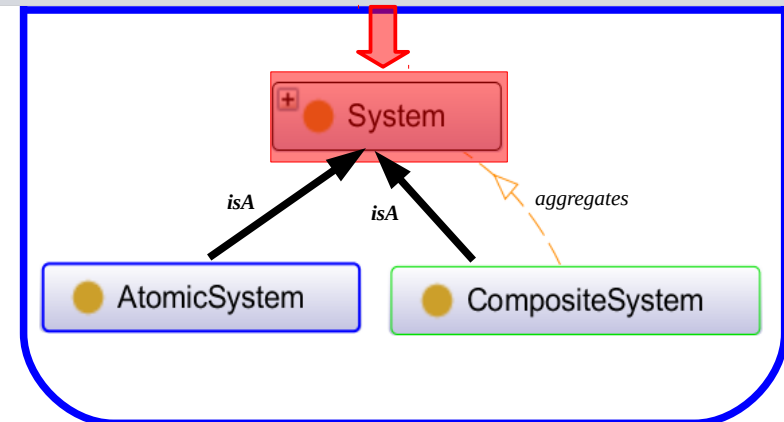
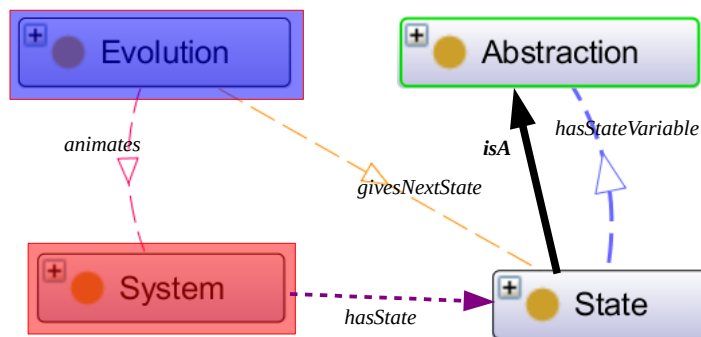
author [language: en]
creation PROTEUS consortium -
modified Bruno Patin - 30/09/2013
- modification of descriptionn to stick to what is a System in this ontology

comment [language: en]
system IS object of the world but that there could be schematic (an actual concrete information) that could represent it (describe it thus). The world IS a system in our ontology. A system that models another means a concrete and actual "in the world" system (such as a simulator that is a software application) providing similar behaviour that another one (such as an aircraft).

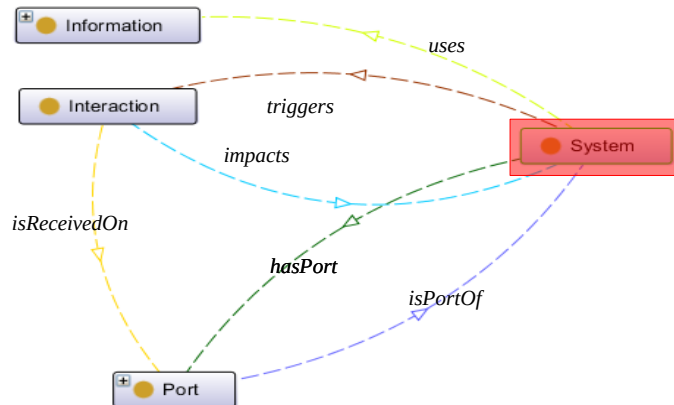
openCycURI [type: anyURI]
<http://sw.opencyc.org/concept/Mx4rv9Xu1ZwpEbGdrcN5Y29ycA>

wikipediaURI [type: anyURI]
<http://en.wikipedia.org/wiki/System>

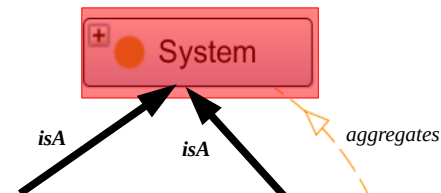
System evolution *Capability to live*



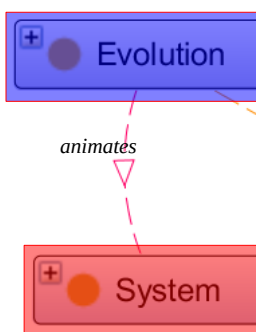
System boundaries *Capability to get / send info*



Composition *Capability to build imbricated blocs*



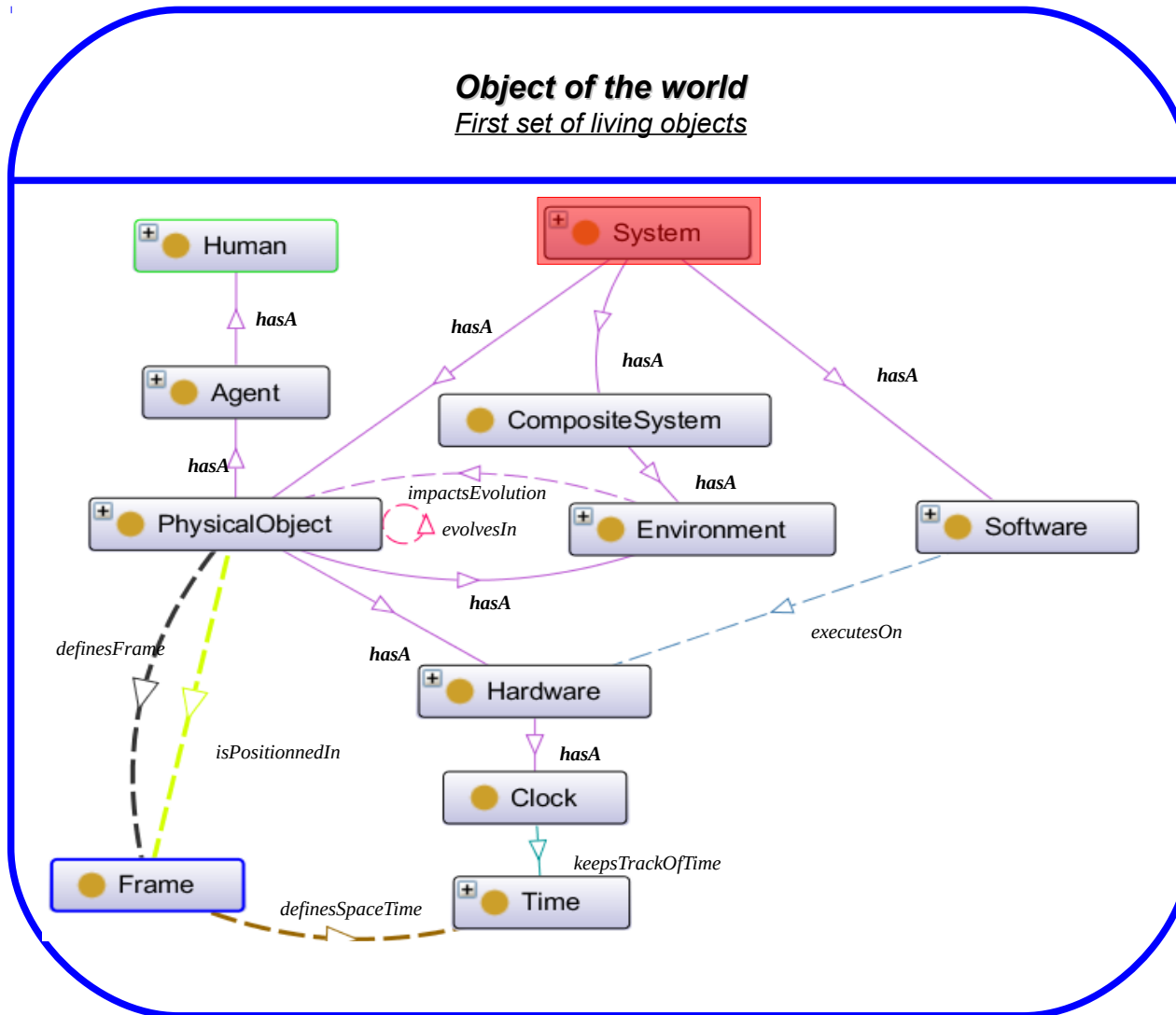
System evolution *Capability to live*



RULES ARE DEFINED THAT PROVIDE MORE SEMANTIC AND INFERENCE CAPABILITIES

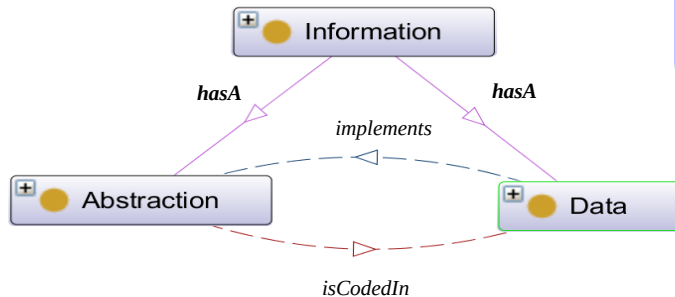
| Rules: | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| Interaction(?z), PhysicalObject(?x1), PhysicalObject(?x2), Mechanical(?x3), Mechanical(?x4), aggregates(?x, ?y), aggregates(?x, ?z) -> Joint(?z) | ? @ x 0 |
| Application(?z), ComputationalHardware(?y), ComputationalSystem(?x), aggregates(?x, ?y), aggregates(?x, ?z) -> executesOn(?z, ?y) | ? @ x 0 |
| Software(?x) -> ComputationalHardware(?y), executesOn(?x, ?y) | ? @ x 0 |
| Build(?y), PhysicalObject(?x), triggers(?x, ?y) -> Human(?x) | ? @ x 0 |
| definesAFrame(?x, ?z), isPositionedIn(?x, ?y) -> DifferentFrom(?y, ?z) | ? @ x 0 |
| Action(?y), PhysicalObject(?x), triggers(?x, ?y) -> Agent(?x) | ? @ x 0 |
| System(?x), System(?y), aggregates(?x, ?y) -> isAtomic(?x, false) | ? @ x 0 |
| Frame(?a), Frame(?b), PhysicalInteraction(?z), PhysicalObject(?x), PhysicalObject(?y), definesAFrame(?x, ?a), definesAFrame(?y, ?b), impacts(?z, ?y), triggers(?x, ?z) -> constrainsFrame(?a, ?b), constrainsFrame(?b, ?a) | ? @ x 0 |
| Interaction(?y), Port(?z), System(?x), emits(?z, ?y), hasPort | ? @ x 0 |

Asserted in: <http://www.anr-proteus.fr/sites/default/files/download/Ontology/kernel.owl>



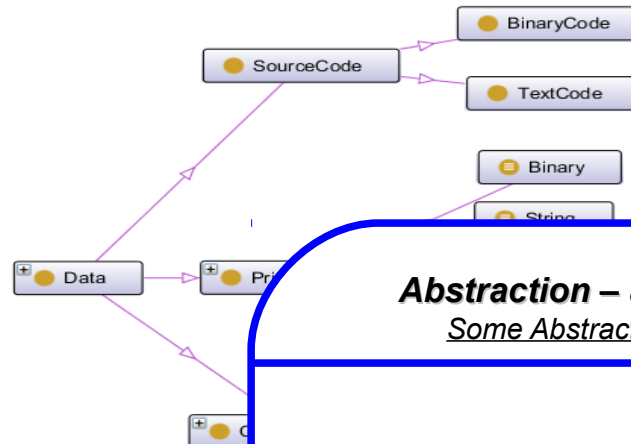
Abstract vs Actual information

Platon and Aristote



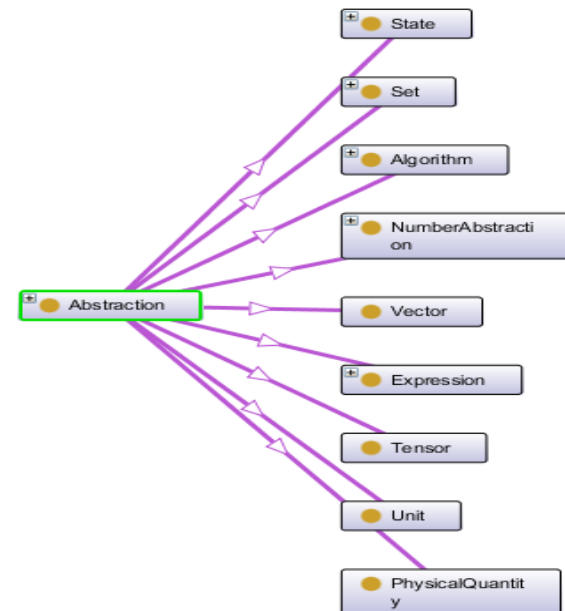
Data – a first zoology

Some Data definitions

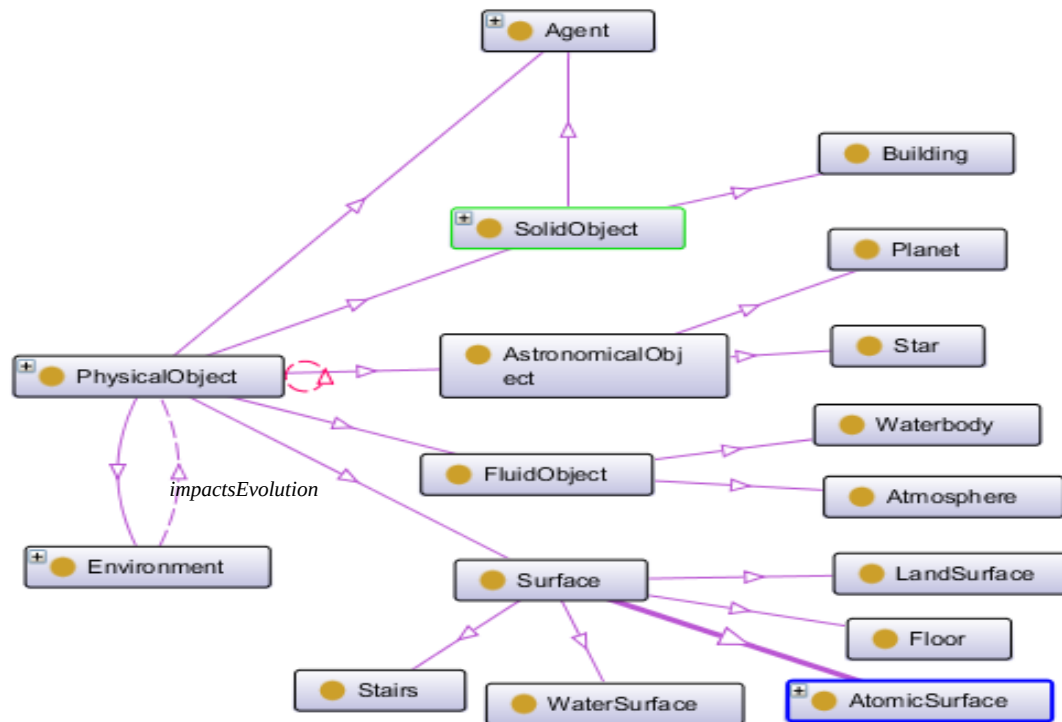


Abstraction – a first zoology

Some Abstraction definitions

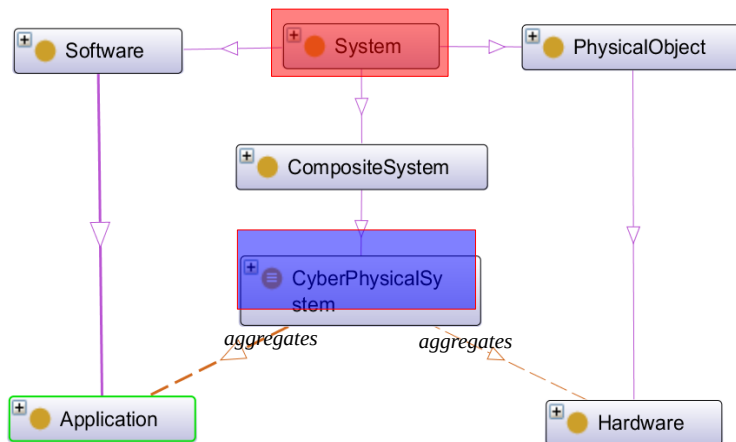


Environment object zoology

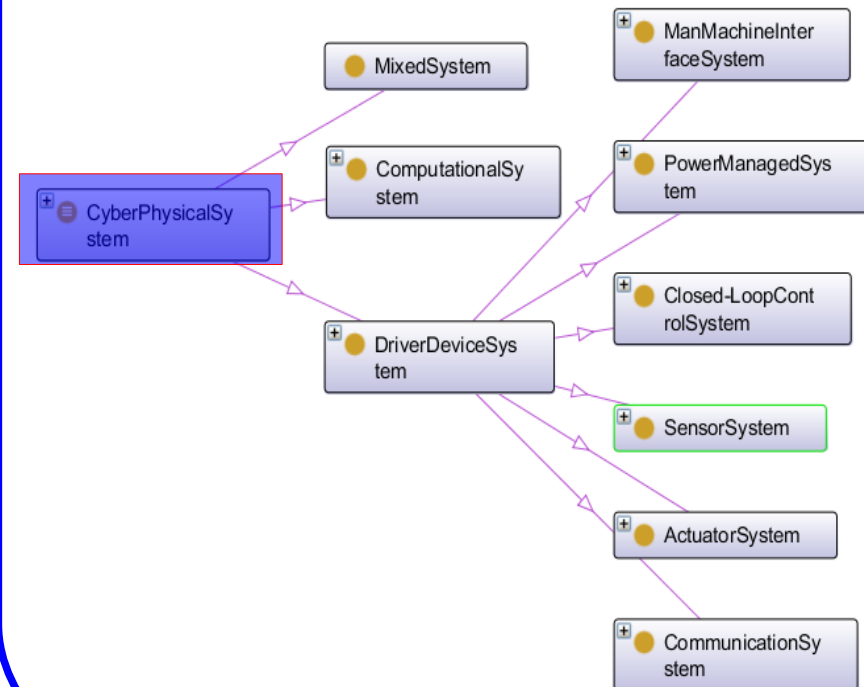


Cyber Physical system - composition

Mechanism to aggregate hardware to software

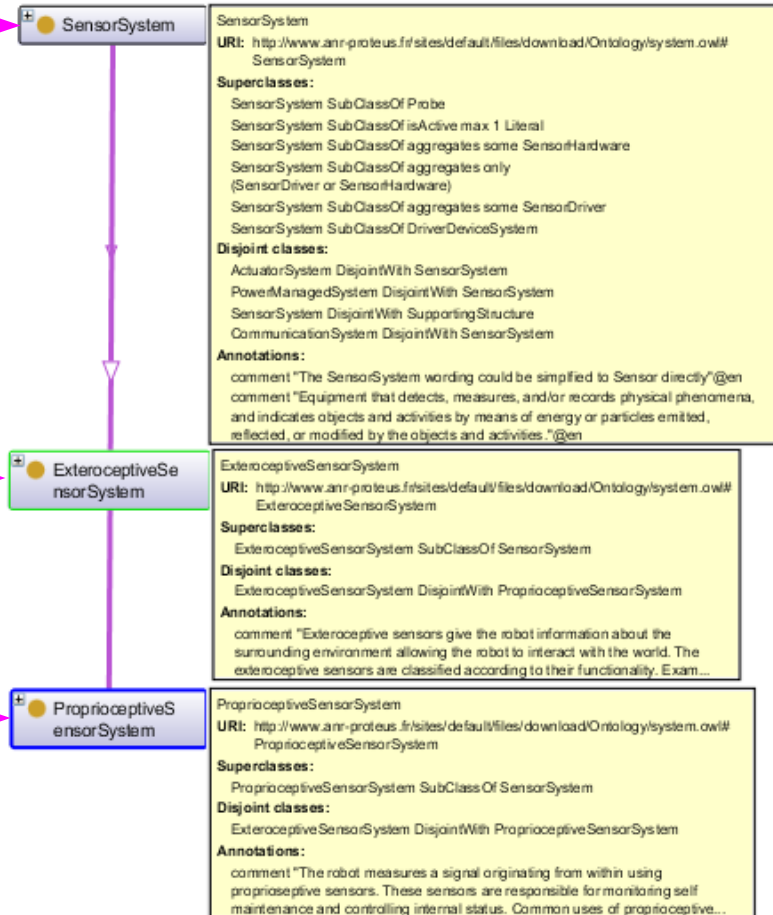
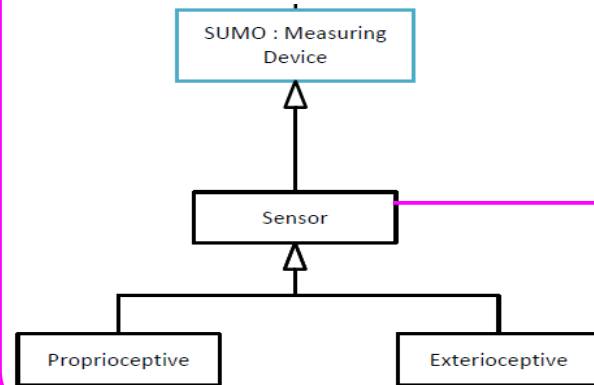


Cyber Physical systems - zoology



ORA alignment

Sensors – How to get data from environment

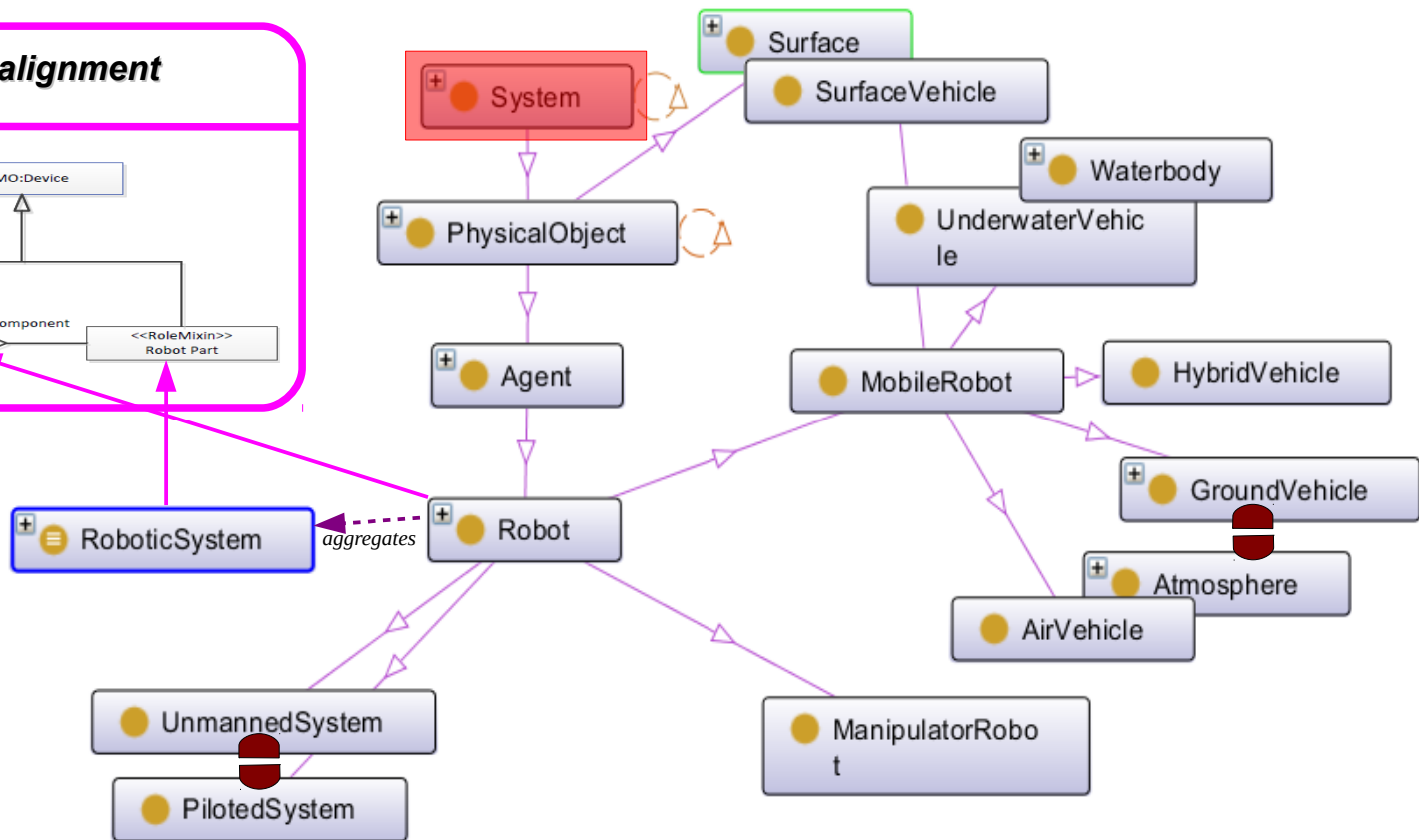
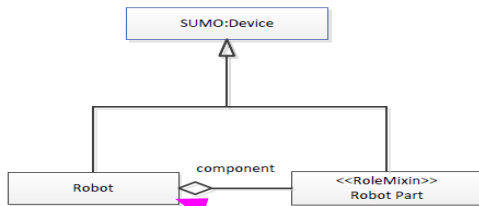


ROBOT – some key concepts

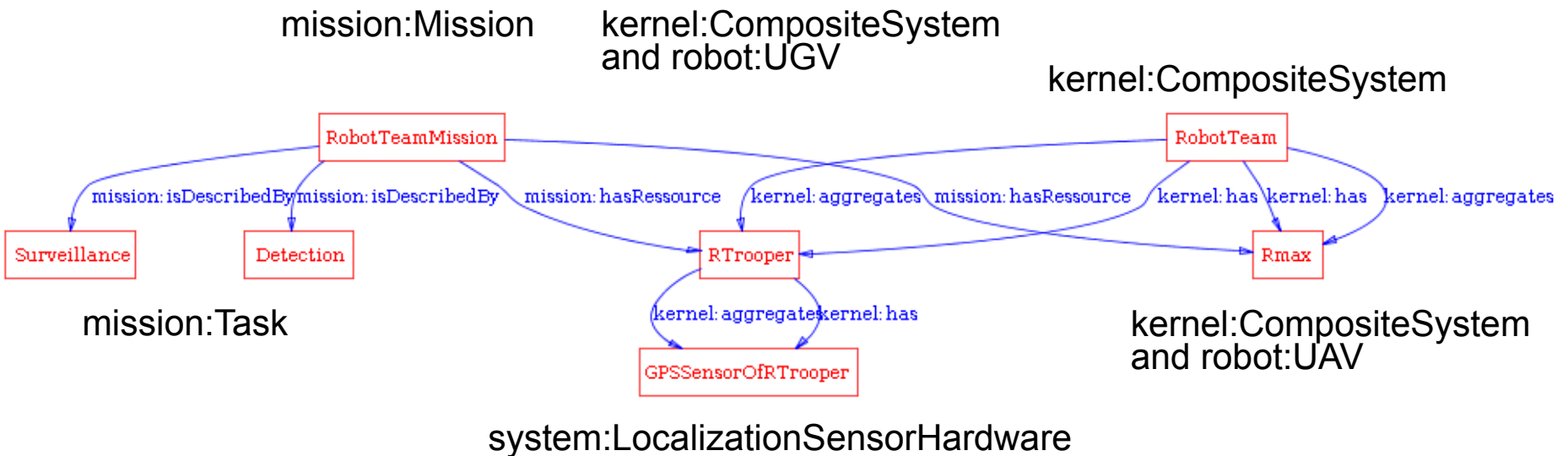
Robot categorisation

Most common classes

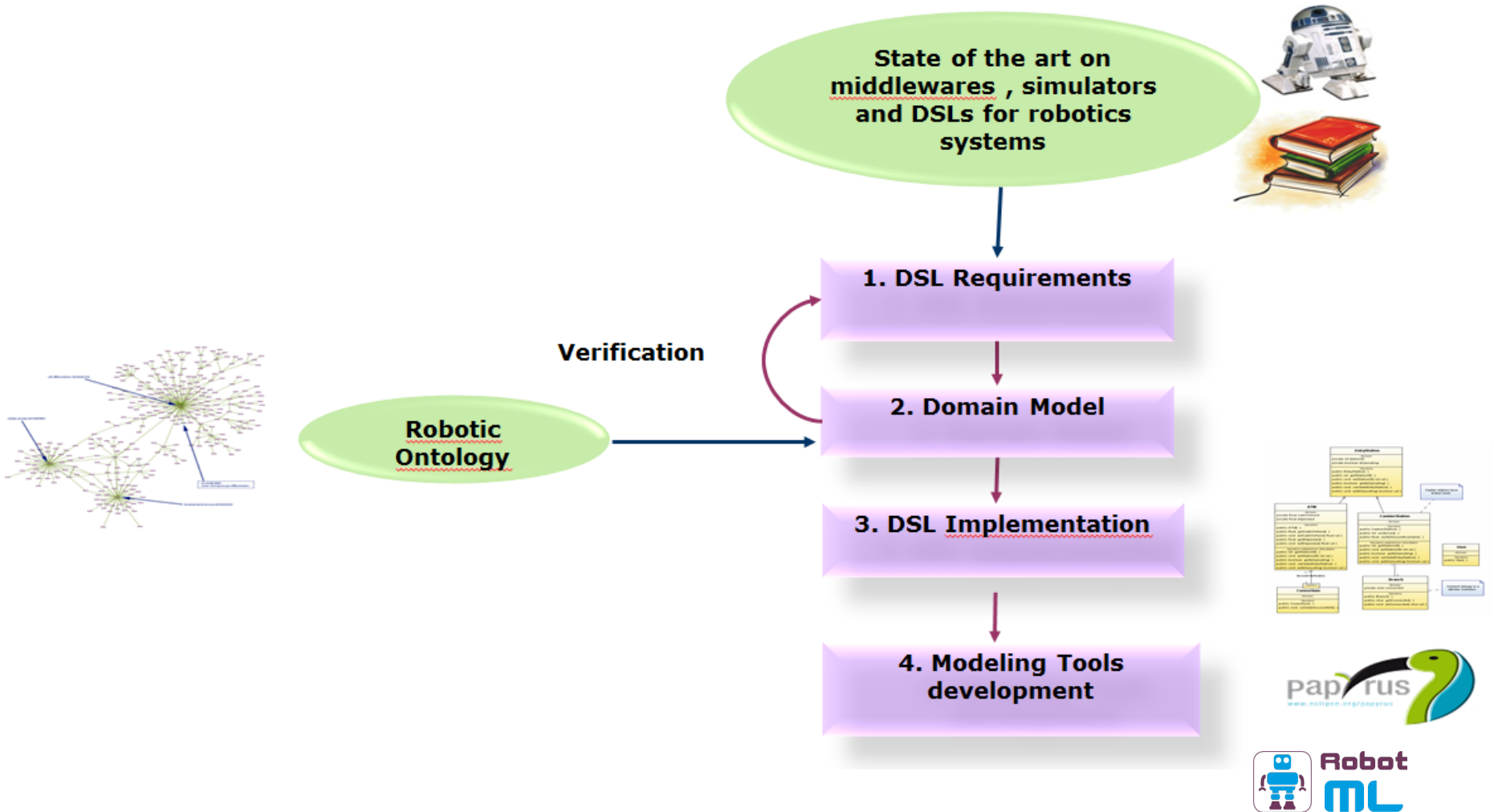
ORA alignment



Validation – An example: part of Air-ground scenario

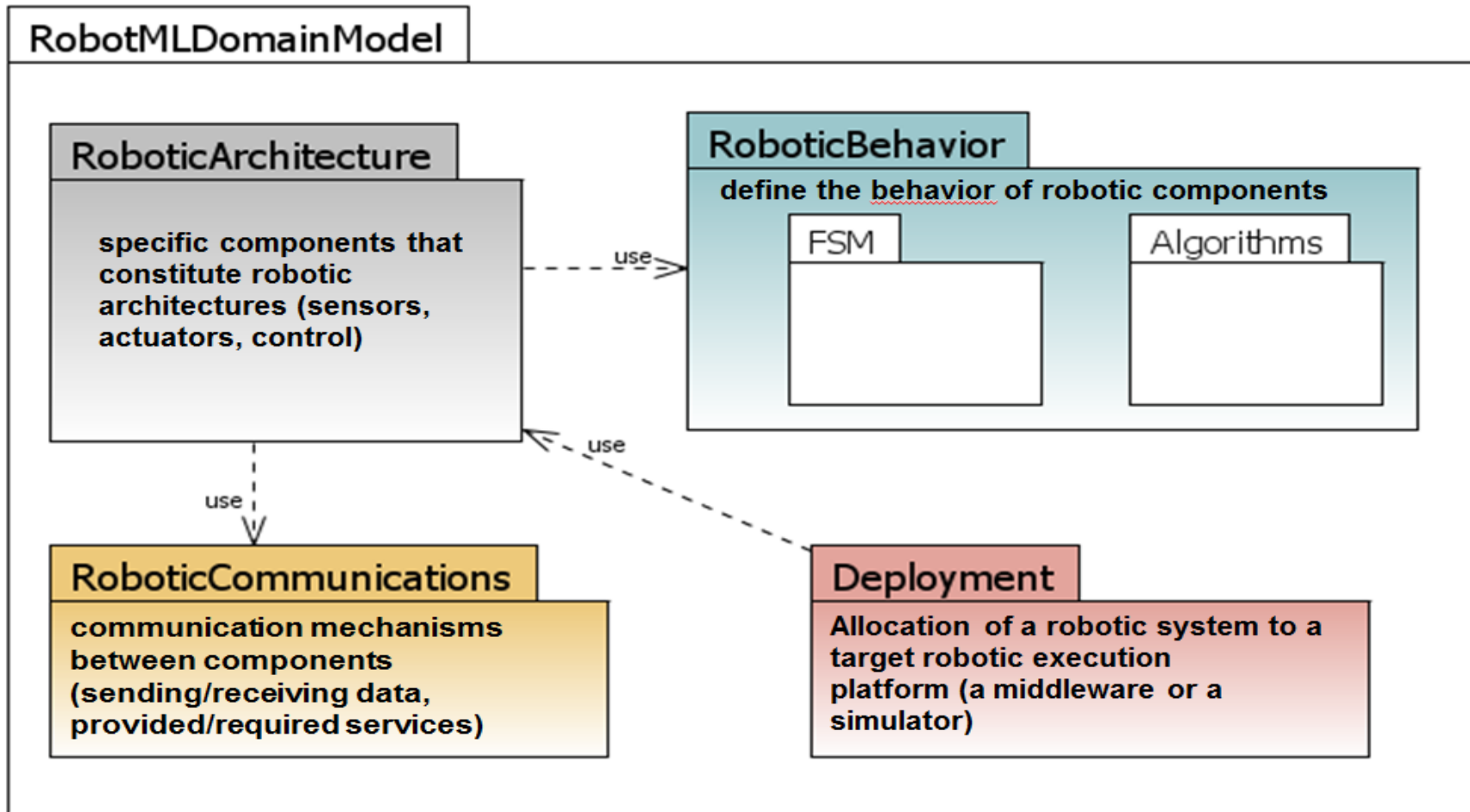


From Ontology to RobotML DSL



Ontology / DSL relationships

| O n t o l o g y (O W L) | D o m a i n m o d e l (U M L c l a s s d i a g r a m) |
|---------------------------|------------------------------------------------------------|
| C o n c e p t | C l a s s |
| s u b C l a s s O f | G e n e r a l i z a t i o n |
| P r o p e r t y | A s s o c i a t i o n |
| P r o p e r t y : I s A | I n h e r i t a n c e |
| P r o p e r t y : H a s A | C o m p o s i t i o n |
| C a r d i n a l i t y | M u l t i p l i c i t y |

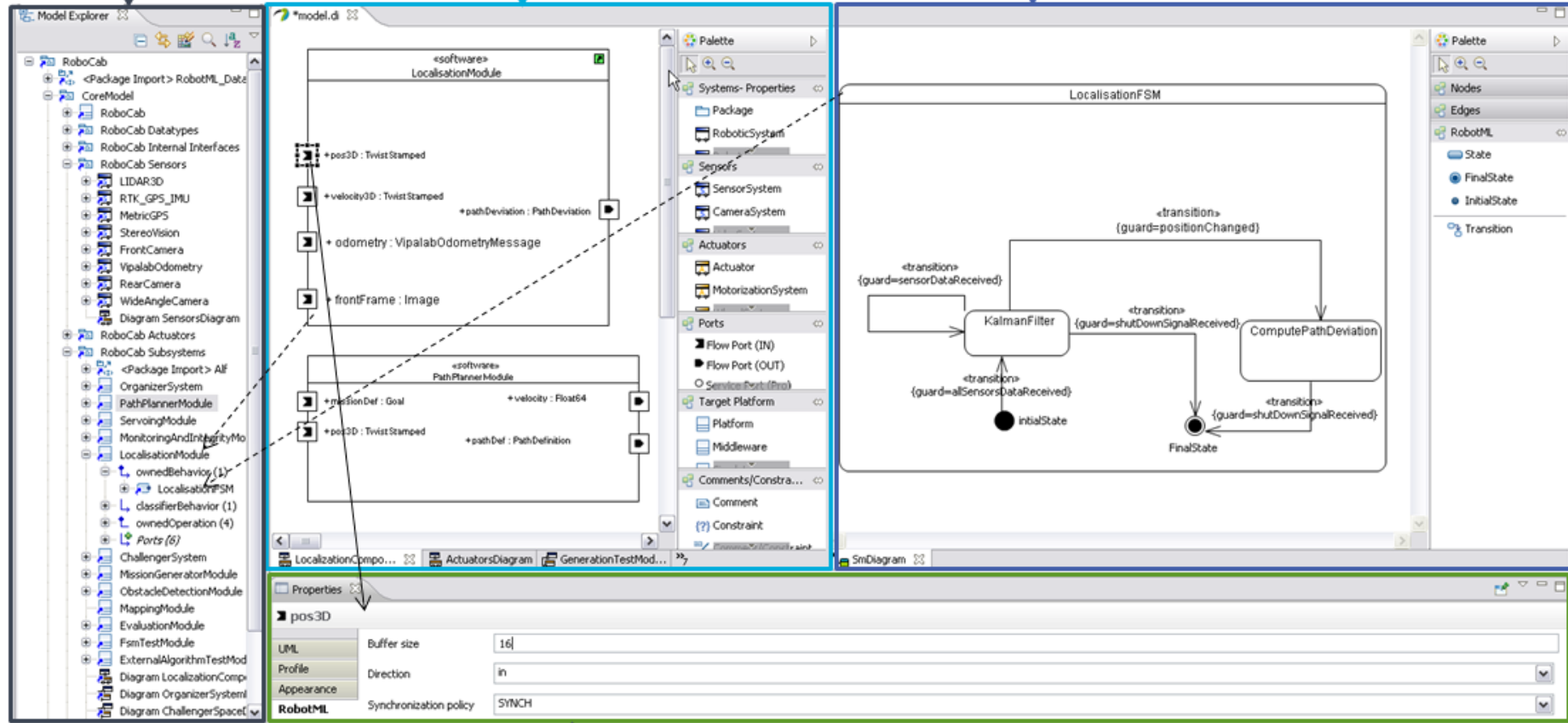


RobotML Modelling Environment

Model Explorer

Components Definition Diagram

Finite State Machine Diagram



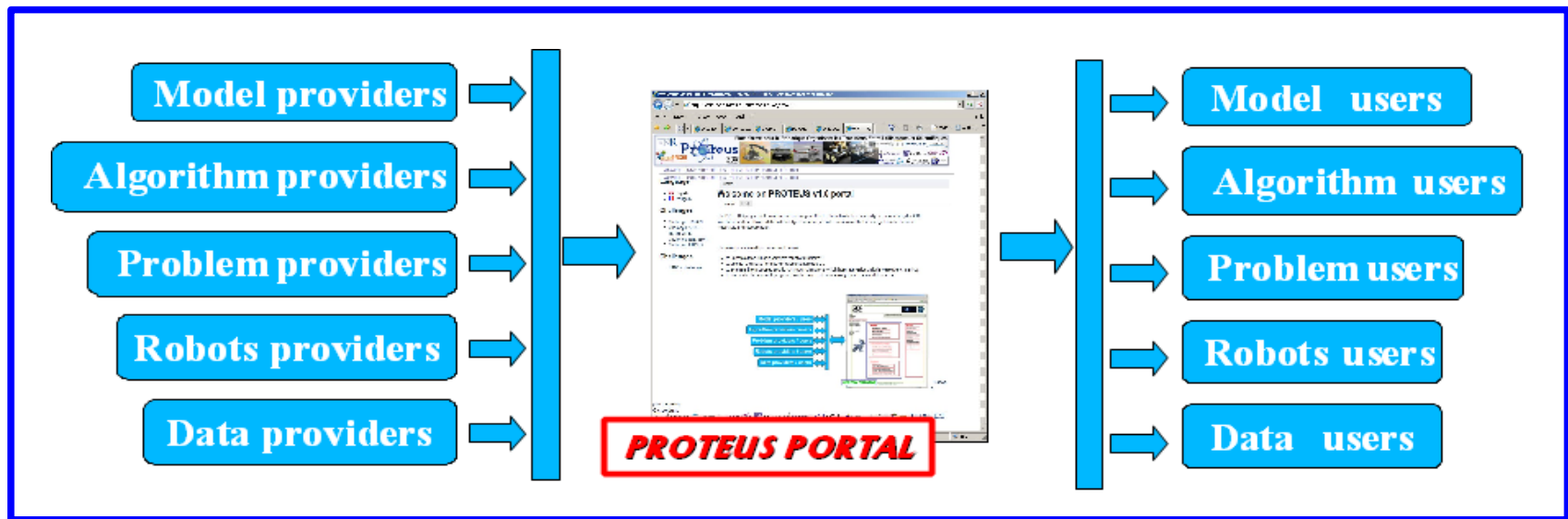
Properties View

Some Conclusions

- This ontology includes more than:
 - ♦ 364 classes
 - ♦ 185 properties
 - ♦ 31 data types
- Nothing actually new, but common frame:
 - ♦ Able to describe some scenarios
 - ♦ Shareable by robotic community
 - ♦ Able to ground a Domain Specific Language (DSL) for robotics
 - Automatic simulation assembly
 - Robotic middleware projection
 - G. Lortal, S. Dhouib and S. Gérard (2011) Integrating Ontological Domain Knowledge into a Robotic DSL, Models in Software Engineering, Lecture Notes in Computer Science, Volume 6627, 401-414
- Work in progress because of expected feedback from:
 - ♦ Language developers
 - ♦ Challenge providers and challengers
 - ♦ **Alignment to existing efforts such as ORA**
 - ♦ **New field such as “Humanoid robotics”**
 - ♦ **New expected project to integrate explicitly “knowledge”**

Questions ?

PROTEUS is a Platform for organizing exchanges between industry and academics in the robotic domain



Use of a common language
Development of a theory grounding the vocabulary



Specification of the theory with an ontology*

*** Ontology : formal representation of knowledge describing a domain**

State of the art* of ontologies for the robotic domain

| <i>Gives Ontology</i> | <i>(Objective</i> | <i>and Scope)</i> |
|-----------------------|------------------------------------------------------------------------|-------------------------------------------------|
| MLCOF | To help object recognition by robot | Robot context |
| OMRKF | To organize robot knowledge | Robot context |
| OCOA | To develop a control architecture | Control architecture of a model with components |
| Deplanques | To assess decisional autonomy | Robotics and environment |
| Schlenoff and Messina | To support development, test and certification of robotic technologies | Urban search and rescue missions |
| RoSta | To standardize | Mobile, handling and service robotics |
| PROTEUS | To support exchanges | Mobile Robotics |

* **not exhaustive, see for example**

http://lists.w3.org/Archives/Public/public-xg-ssn/2009Aug/att-0037/SSN-XG_StateOfArt.pdf&usg=AFQjCNGMxi0CHy4FJuxquXVneXfJF6k2tg



Intellectual Property Rights are governed by PROTEUS Contract and PROTEUS consortium agreements. Before using, reproducing, modifying or disclosing the information contained in that document, it is under the user responsibility to check its rights and if mandatory ask to the Intellectual Property Owner the authorisation to use, reproduce, modify or disclose the information.