Title

ASys Models

Model-driven Engineering in ASys

Author

Ricardo Sanz, Carlos Hernández, Manuel Rodríguez

Reference R-2007-016
Release 0.1 Draft
Date August 5, 2008

Address
Autonomous Systems Laboratory
UPM - ETS Ingenieros Industriales
José Gutierrez Abascal 2
28006 Madrid
SPAIN

ASys Models

ASLab R-2007-016 v 0.1 Draft of August 5, 2008

Abstract

This report describes the modeling strategy in the **ASys Project**.

As this is a long-term research project focused in the development of technology for the construction of autonomous systems, the selected strategy for design knolwedge capture is the strategy of Model-driven Engineering.

One of the central topics in the ASys Project is the pervasive model-based approach. An ASys will be using models to coordinate, drive and perform its activities. An ASys will be though-of, designed and built using models of it.

The report presents the rationale for an MDE approach in ASys, the tooling employed and the structure of the models used.

The report will also contain summarial description of the ASys models that will be addressed in detail in specific reports.

Keywords

Autonomy, product-line engineering, models, complex control systems, software-intensive real-time systems.

Acknowledgements

We acknowledge the support get from the Spanish Ministry of Education and Science, the European Comission and the Goverment of Madrid that have provided support for this long-term project through several, topic-focused grants.

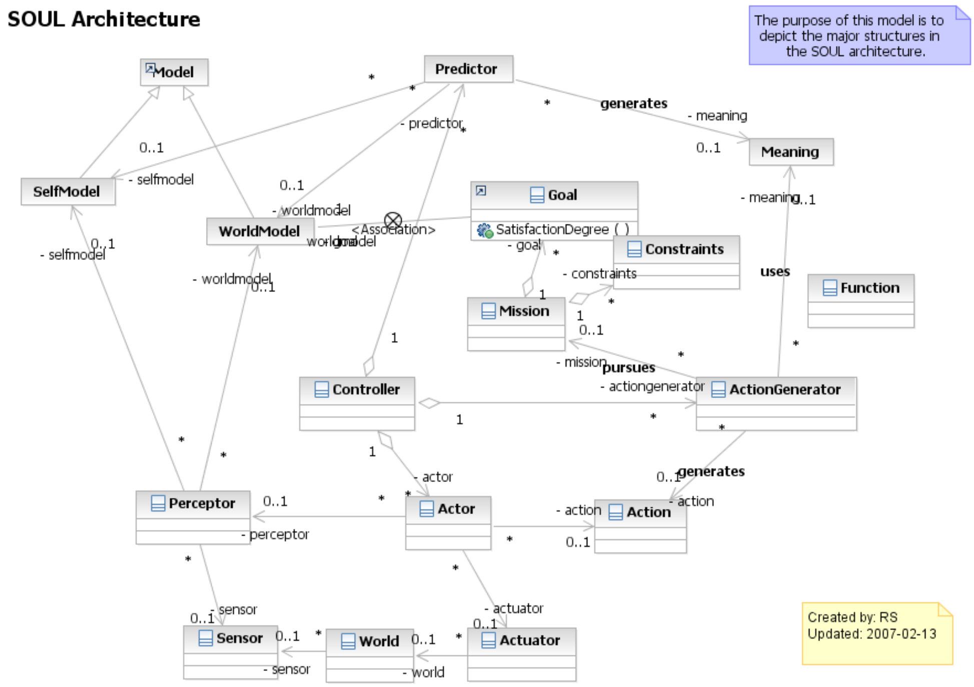
Table of Contents

Chapter 1

Introduction

Models are pervasive entities in engineering activities and not only as a vehicle for design capture but as essential pieces in the perfromance of certain engineering tasks. Examples abound: from physical models used in wind tunnels for the performance-tuning of vehicles to the formal models used in railway engineering to mathamatically proof the operational safety of a signaling system.

1.1 Excerpts from the ASys Vision



The purpose of this model is to depict the major structures in the SOUL architecture.

Created by: RS
Updated: 2007-02-13

Figure 1.1: The ASys models capture the design of the different aspects of the autonomous systems construction; from low level, implementation related issues to high-level mental aspects.

Chapter 2

The Model-driven Approach

2.1 The System and its Environment

In most modelling approaches used in engineering, the *system* and *its environment* are treated very differently from the modelling perspective.

We can put some examples of UML models, control models and simulation-oriented models.

However he he ASys approach system and environment do have the same ontological/epistemological status from the perspective of the ASys' epistemic system: both are **knowables** . Hence if both are knowables, the mental stuctures used by te ASys to know about itself and to know about the environment must share their class (or just superclass?).

2.2 State, behavior and components

According to Mesarovic [Mesarovic and Takahara, 1989, p.45], the concept of state can be explained as follows: *"The state enables the determination of a future output solely on the basis of the future input and the state the system is in. In other words, the state enables a 'decoupling' of the past from the present and future. The state embodies all past history of a system.* Knowing the state 'supplants' knowledge of the past. *Apparently, for this role to be meaningful, the notion o past and future must be relevant for the system considered; this leads to the notion of an abstract time system."*

2.3 The problem of scale

Any practical methodology for model-based systems engineering needs to solve the problems derived from system scale and try to overcome the many limitations of current computer-supported design, synthesis and verification techniques [Henzinger and Sifakis, 2007]. These techniques come from the domain of safety-critical and performance-critical systems engineering and are often limited in their applicability by the exponential problem-solving complexity derived from the size of the system under analysis.

One route to circumvent this limitation and achieving scalability is to rely on composability properties such as compositionality, incrementality, and non-interference, which allow inferring global properties of the system from those of its components and the way these are integrated.

From the perspective of autonmous systems, the embedding of modular self-awareness and autonomy will render modules much more stable and hence simpler for the analysis and design task. A self-managing component is more trustable and hence some considerations about its state and behavior may be obviated.

Chapter 3

The Model-based Toolset

Chapter 4

ASys Model Structure

The ASys models follow the principle of progressive domain focalisation [Sanz et al., 1999].

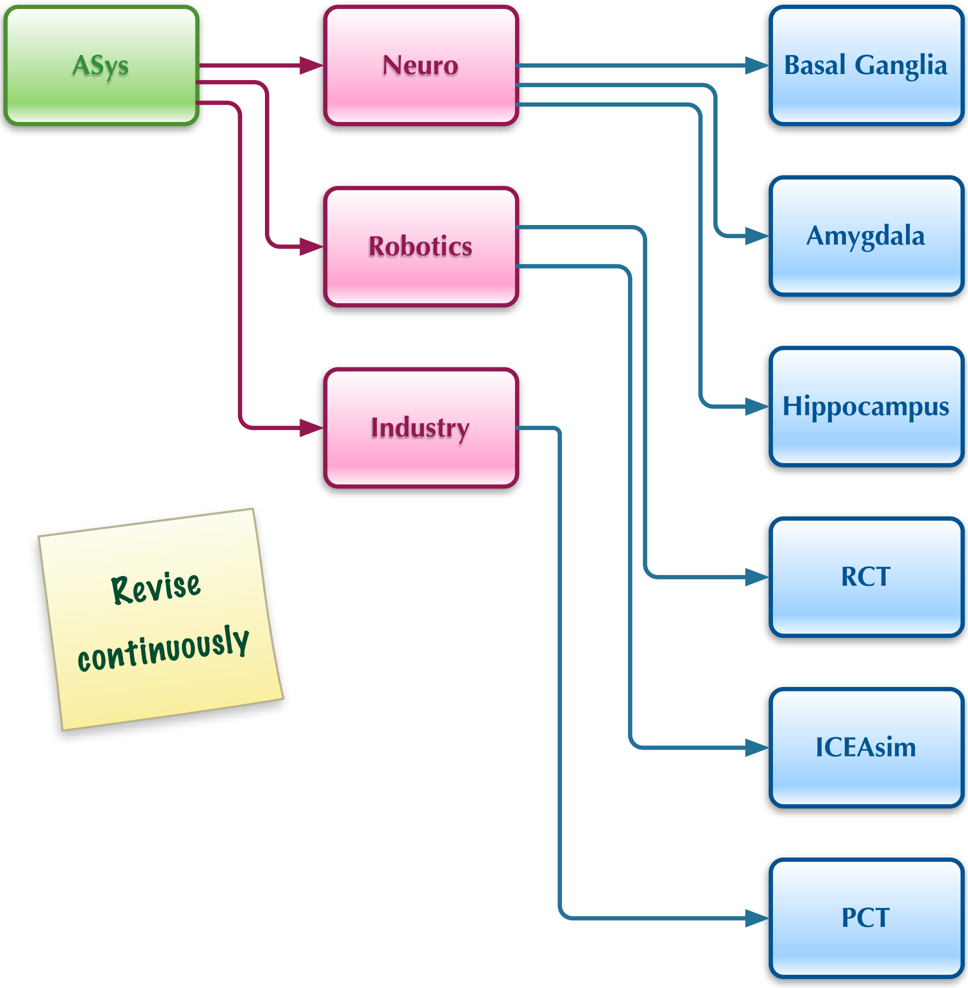


Figure 4.1: The ASys models are organised by domains.

Chapter 5

ASys Core Model Entities

The ASys models contain may elements of varying degrees of abstraction. A full MDE process requires a maximal degree of precision and hence terms must be used with care. This chapter contains the rationale behind some of the core ASys Framework concepts:

- System
- Agent
- Context
- Environment
- ...

5.1 System

Used in the sense of systems theory we do not constrain the use of **System** to the system under study or the object of engineering. While the objective of the ASys project is the development of autonomous cognitive systems, the term may be used to describe other foci of attention of a systemic analysis and not necessarily the ASys.

5.2 Agent

There is a lot of literature about agents in the fields of AI and artificial life, where the term is basically defined as "any entity that is capable of perceiving its environment and carrying out goal-directed action"

For us, within the framework of ASys, an agent will be any system that can act upon its environment basing that action in a model of the environment.

5.3 Context

Let's see a definition of **Context** from the domain of systems/software engineering:

> *In systems engineering, context includes the set of things (people, other systems, and so forth) with which the system interacts and how those interactions proceed so that the system can fulfill its role in the enterprise.*
>
> [Balmelli et al., 2006]

5.4 Environment

Chapter 6

ASys Neural Models

The ASys neural models are base on the study of the biological control systems of animals and the attempt to reverse engineer the architectural patterns of some of the core neural structures.

6.1 The basal ganglia

6.2 The amygdala

6.2.1 Overall description of amygdala

6.2.2 Amygdala functionality

6.2.3 Amygdala model

6.3 The hippocampus

from Petrovich et al. Brain Res. Rev. 38: 247-89, ’01

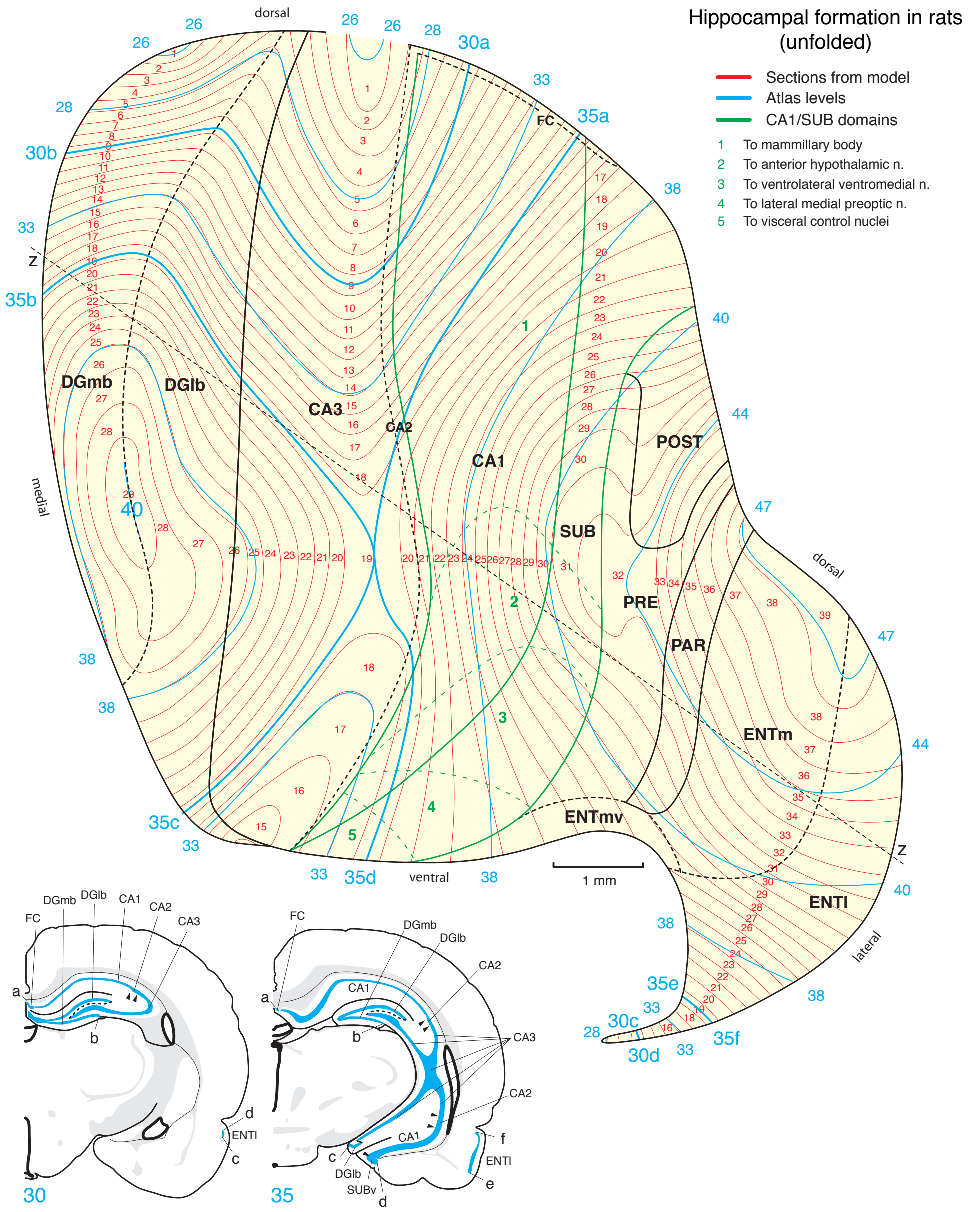


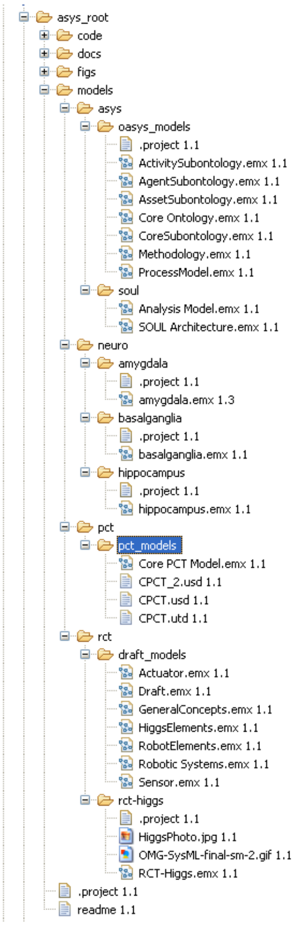
Figure 6.1: The hippocampal formation. The colored curves and labeling corresponds to section in Swanson 2003.

Chapter 7

Reference Materials

7.1 Model files

Models are built using modelling languages and are stored in the CVS repository.



```
asys_root
  code
  docs
  figs
  models
    asys
      oasys_models
        .project 1.1
        ActivitySubontology.emx 1.1
        AgentSubontology.emx 1.1
        AssetSubontology.emx 1.1
        Core Ontology.emx 1.1
        CoreSubontology.emx 1.1
        Methodology.emx 1.1
        ProcessModel.emx 1.1
      soul
        Analysis Model.emx 1.1
        SOUL Architecture.emx 1.1
    neuro
      amygdala
        .project 1.1
        amygdala.emx 1.3
      basalganglia
        .project 1.1
        basalganglia.emx 1.1
      hippocampus
        .project 1.1
        hippocampus.emx 1.1
    pct
      pct_models
        Core PCT Model.emx 1.1
        CPCT_2.usd 1.1
        CPCT.usd 1.1
        CPCT.utd 1.1
    rct
      draft_models
        Actuator.emx 1.1
        Draft.emx 1.1
        GeneralConcepts.emx 1.1
        HiggsElements.emx 1.1
        RobotElements.emx 1.1
        Robotic Systems.emx 1.1
        Sensor.emx 1.1
      rct-higgs
        .project 1.1
        HiggsPhoto.jpg 1.1
        OMG-SysML-final-sm-2.gif 1.1
        RCT-Higgs.emx 1.1
  .project 1.1
  readme 1.1
```

Figure 7.1: The ASys models are organised by domains in the CVS repository.

```
/cvsroot/asys_root/models/neuro/amygdala
/cvsroot/asys_root/models/neuro/basalganglia
/cvsroot/asys_root/models/neuro/hippocampus
```

7.2 Aditional Readings

Model Lifecycle Management
Dr. Michael Tiller
Frontiers 2007
http://www.pslm.gatech.edu/events/frontiers2007/

7.3 Acronym Nightmare

AD PTF OMG Analysis & Design Task Force

BMI DTF OMG Business Modeling & Integration Domain Task Force

BPEL Business Process Execution Language (OASIS)[1]

BPEL4WS Business Process Execution Language for Web Services

CL ISO 24707 Common Logic: a family of first order logic languages, including Conceptual Graphs & Common Logic Interchange Format —a successor to the Knowledge Interchange Format (KIF)[2]

DAML DARPA Agent Mark-up Language, one of the primary languages leading to the development of OWL[3]

DAML-S Services ontology for DAML[4]

DARPA Defense Advanced Research Projects Agency[5]

DL Description Logics: a subset of first order logic, for which tractable & complete reasoning systems are available

ER Entity Relationship modeling

IMM Information Management Metamodel (a.k.a. CWM2)

MDA Model-Driven Architecture[6]

MMF Metamodel Management Framework (ISO 19763)

ODM Ontology Definition Metamodel

---

[1] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel

[2] http://cl.tamu.edu/

[3] http://www.daml.org/

[4] http://www.daml.org/services/

[5] http://www.darpa.mil/

[6] http://www.omg.org/mda/

OWL W3C Web Ontology Language, a formal W3C Recommendation as of 10 February 2004[7]

OWL DL The normative description logics dialect of OWL

OWL Full The normative OWL dialect that has increased expressivity over OWL DL, but does not conform to DL reasoning requirements

OWL-S A set of OWL ontology components that extend the W3C OWL specifications to support Semantic Web Services[8]

PRR Production Rules Representation

QVT MOF Query / View / Transformations Specification[9]

RIF Rule Interchange Format[10]

RDF Resource Description Framework[11]

SBVR Semantics for Business Vocabularies and Rules

SOA Service Oriented Architecture

SOAP Simple Object Access Protocol[12]

SWSF Semantic Web Services Framework[13]

TM ISO 13520 Topic Maps[14]

WSDL Web Services Description Language

---

[7] http://www.w3.org/TR/owl-semantics/
[8] http://www.daml.org/services/
[9] http:// www.omg.org/docs/ptc/05-11-01.pdf
[10] http://www.w3.org/2005/rules/wg
[11] http://www.w3.org/TR/rdf-concepts/
[12] http://www.w3.org/TR/soap/
[13] http://www.w3.org/Submission/SWSF/
[14] http://www.isotopicmaps.org/sam/sam-model/

Bibliography

[Balmelli et al., 2006] Balmelli, L., Brown, D., Cantor, M., and Mott, M. (2006). Model-driven systems development. *IBM Systems journal*, 45(3):569–585.

[Henzinger and Sifakis, 2007] Henzinger, T. and Sifakis, J. (Oct. 2007). The discipline of embedded systems design. *Computer*, 40(10):32–40.

[Mesarovic and Takahara, 1989] Mesarovic, M. and Takahara, Y. (1989). *Abstract Systems Theory*. Springer-Verlag, Berlin.

[Sanz et al., 1999] Sanz, R., Alarcón, I., Segarra, M. J., de Antonio, A., and Clavijo, J. A. (1999). Progressive domain focalization in intelligent control systems. *Control Engineering Practice*, 7(5):665–671.

Title: ASys Models
Subtitle: Model-driven Engineering in ASys
Author: Ricardo Sanz, Carlos Hernández, Manuel Rodríguez

Date: August 5, 2008
Reference: R-2007-016 v 0.1 Draft

URL: http://www.aslab.org/documents/R-2007-016.pdf

Autonomous Systems Laboratory

Universidad Politécnica de Madrid
c/José Gutiérrez Abascal, 2
Madrid 28006 (Spain)