CHESS

# CHESS

Composition with Guarantees for High-integrity
Embedded Software Components Assembly

## ARTEMIS JU Project

**Silvia Mazzini**
**Intecs**

Credits to University of Padua

# CHESS Project

**ARTEMIS JU project**

Call 1 2008

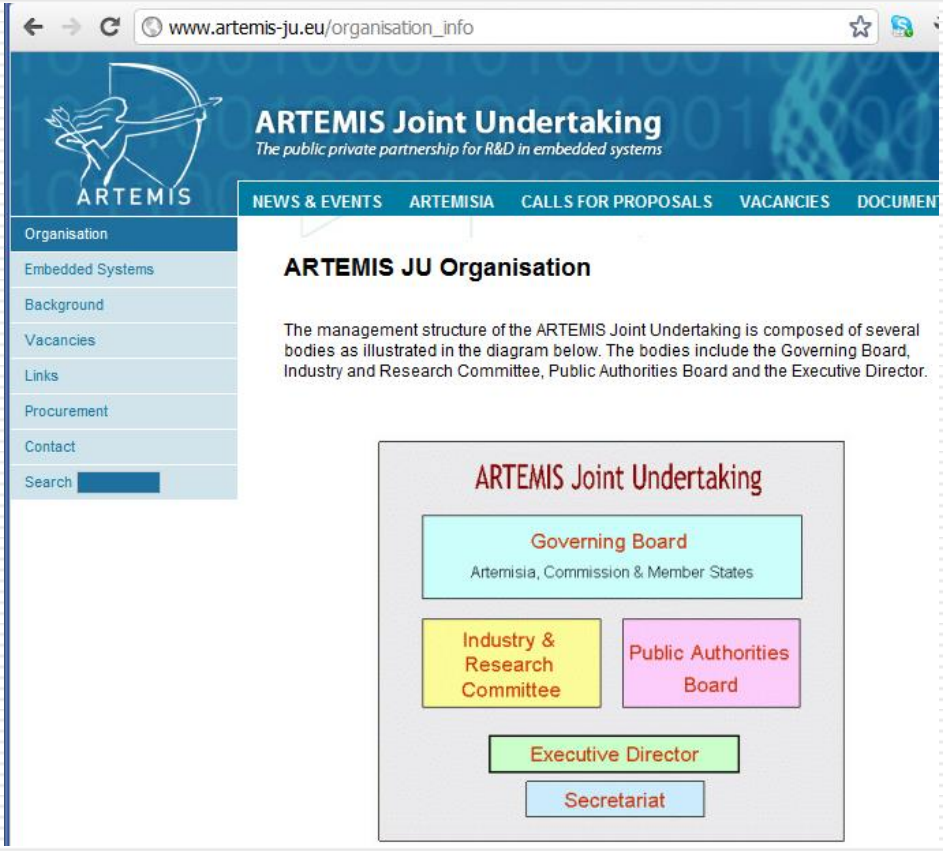
**Technical Coordinator** Intecs

**Partners** 18

**Countries** 6

**Start** February 1st, 2009

**Duration** 3 Years



# CHESS Partners

- Industrial Partners
  - Intecs (I)
  - Italcertifer (I)
  - Thales Alenia Space (F)
  - Thales Communications (F)
  - Aonix (F)
  - GMV (E)
  - Atos Origin (E)
  - Aicas (D)
  - X/Open Company Limited-The Open Group (UK)
  - Ericsson (SW)
  - Enea (SW)
- Research Centres
  - CNR/ISTI (I)
  - INRIA (F)
  - Fraunhofer ESK (D)
  - Forschungszentrum Informatik FZI (D)
- Universities
  - University of Padua (I)
  - Universitad Politecnica de Madrid (E)
  - Maelardalen University (SW)
  - University of Florence (I) (as subcontractor of ISTI/CNR)

# CHESS objectives

- Definition of a Multi-Concern Component Methodology and Toolset
  - Provide a Multi-Concern Component Modeling Language and a Graphical Modelling Environment that fits multiple industrial domains
  - Enable the specification of functional and extra-functional* properties of software components
  - Integrate tools for the verification of extra-functional properties
  - Preserve verified properties at code level and run time
- Adaptation of standards and open sources
  - OMG modeling languages
  - Eclipse Environment
- Validation through multi-domain industrial case studies

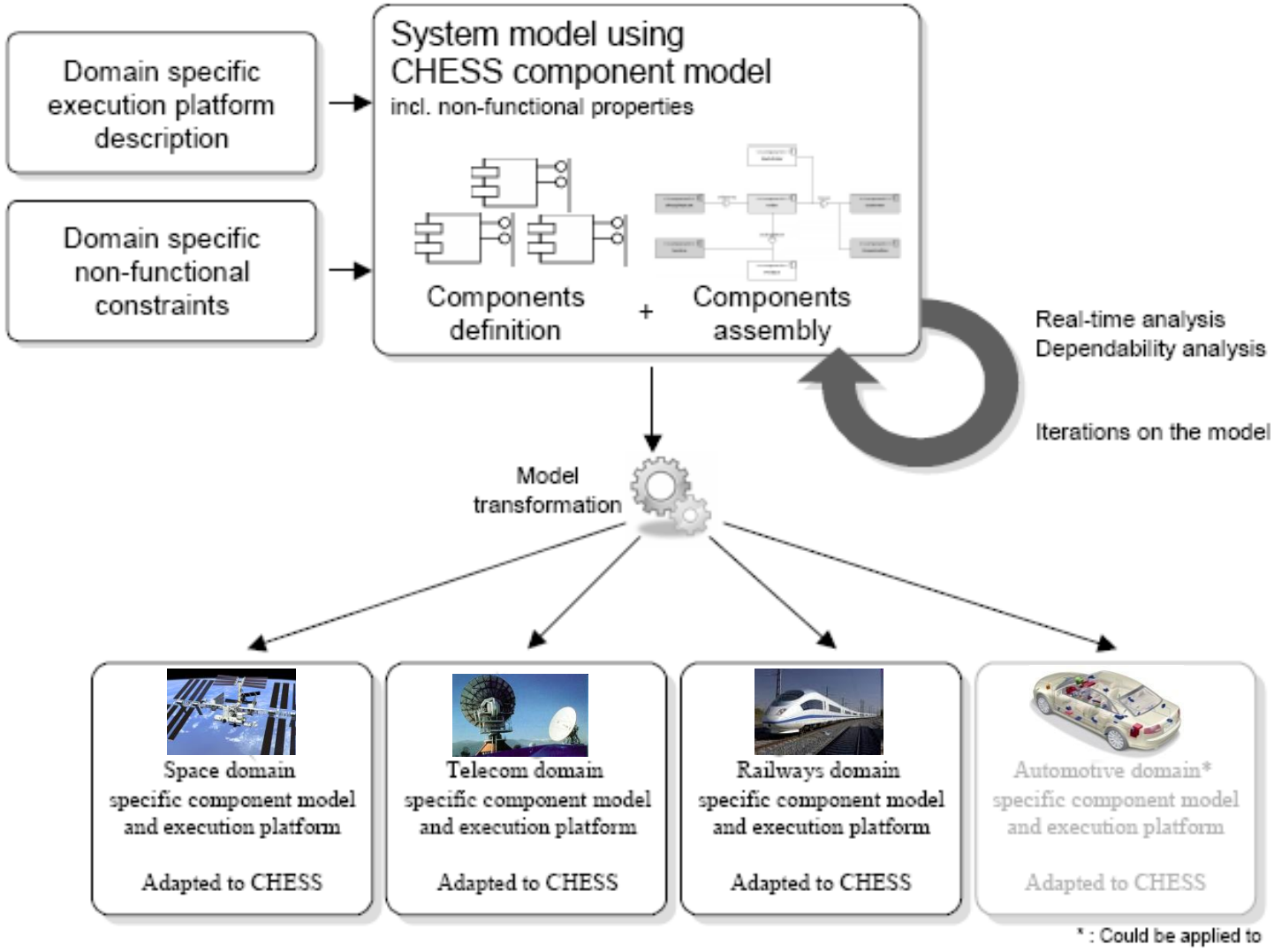
**Extra-functional* is a synonym of ***non-functional***, as non-functional may have connotations of not functioning*

# Extra-functional properties and Analyses

CHESS

- Focus is on
  - clearly and cleanly separating the extra-functional part of a software component from its functional part
  - ensuring that extra-functional properties are asserted and validated at model level and then preserved at code level and run time
- Extra-functional dimensions and analysis methods of interest
  - Real-Time
    - Scheduling Analysis, Bus Configuration Analysis, Simulation Based Timing Analysis, Code and Execution Analysis
  - Dependability/Safety
    - FTA, FMECA, FMEA, State-Based, Wide Data-flow&Call-graph and Failure Propagation Analysis

# The CHESS approach 1/2

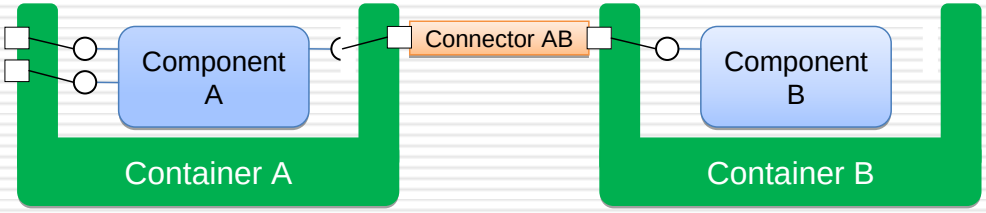


# The CHESS approach 2/2

- Model-driven engineering
  - Models as the central development artifacts
  - Tool assisted automated development
- Component based development
  - Specialized to capture the extra-functional properties of components
    - Real Time
    - Dependability
- Separation of concerns
  - Functional vs extra-functional
  - Among extra-functional dimensions (dependability vs predictability)
  - Among design levels/roles
- Correctness by construction
  - Extra-functional properties are:
    - asserted and verified at design time
    - Preserved/guaranteed at code level and run time

# The Component Model

- Component
  - Reusable functional unit
- Container and Connector
  - Encapsulation of the extra-functional properties of components
  - Factorized implementation

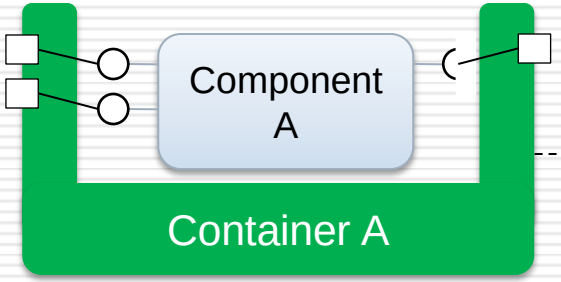



- Composability
  - properties of individual components are preserved on component composition
- Compositionality
  - properties of the system as a whole can be derived as a function of the properties of components

# Separation of concerns with the CHESS component model

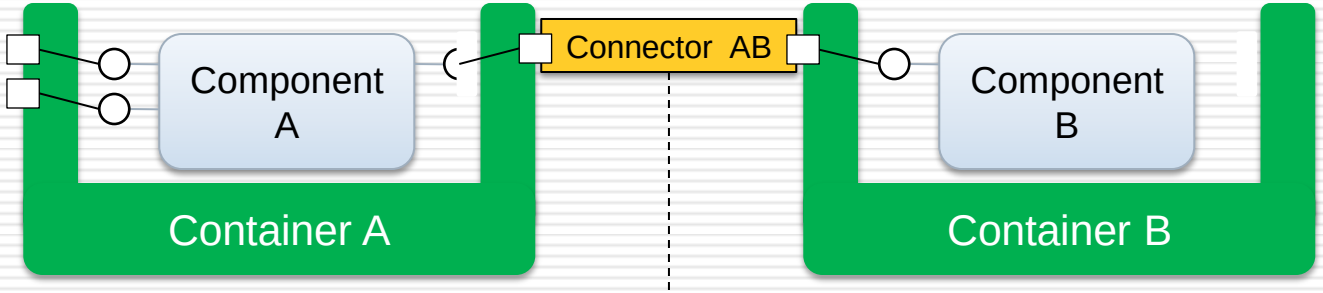
CHESS

Container




- Wrapper responsible for the declared extra-functional attributes
- Provides the component with a mediated connection with the execution platform and the system in general

Connector

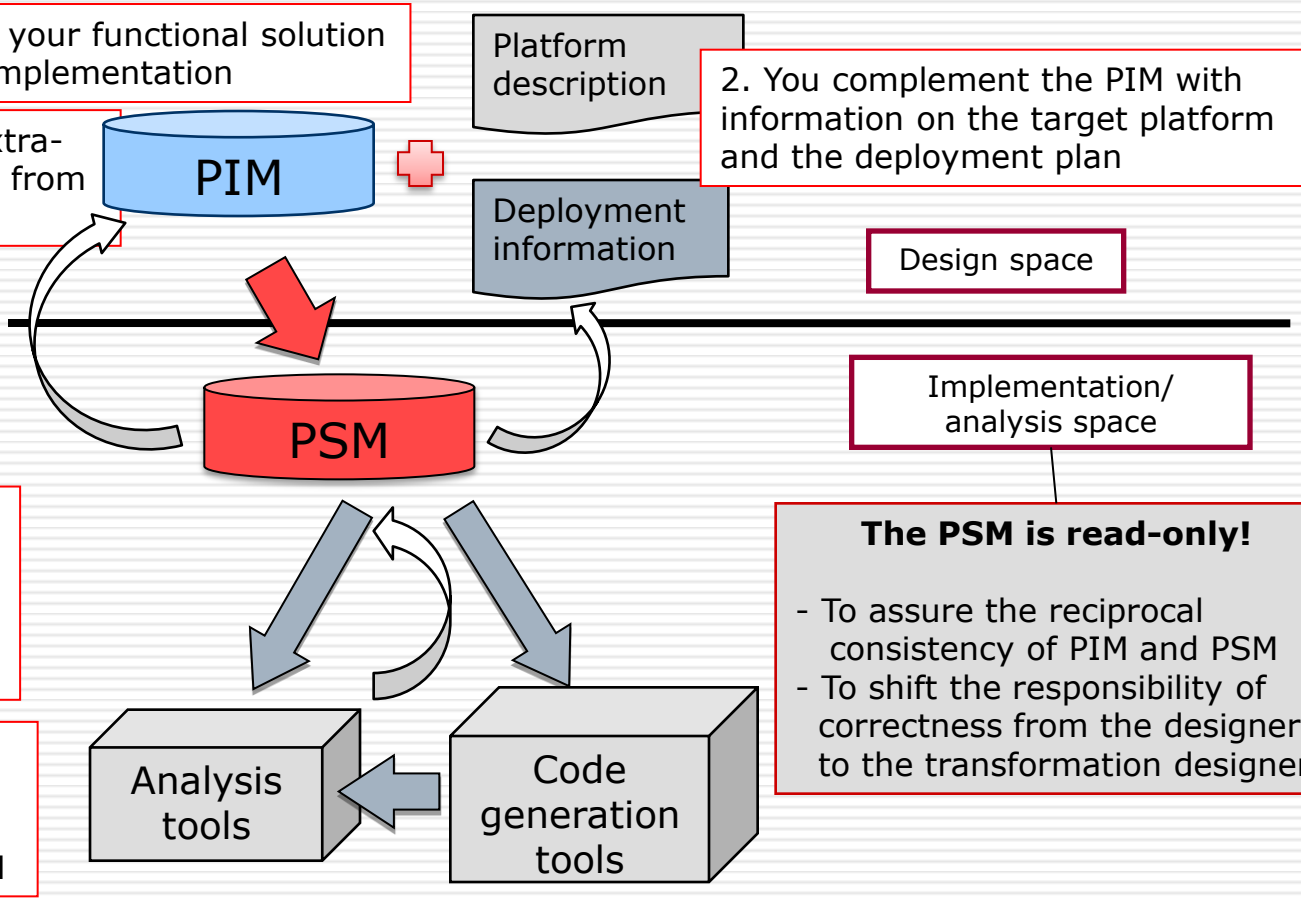



- Addresses interaction concerns
- Decouples the component from the other end-point(s) of a communication
- Realizes connection properties (best-effort, at most once, exactly once)
- E.g. procedure/function call, remote message passing, I/O file operation, …

# The CHESS high level design process

1a. You construct a PIM to specify your functional solution to your problem, independent of implementation

1b. You decorate your PIM with extra-functional attributes (independent from any computational model)




3. The design environment generates a PSM automatically via model transformation. The PSM is bounded to a given computational model.

4. A back-end tool processes the PIM, the PSM or the code to feed specialized analysis tools (dependability, schedulability, etc)

5. The back-end tool reports the analysis results back on to the PSM and attaches them to the corresponding entities in the PIM

**The PSM is read-only!**

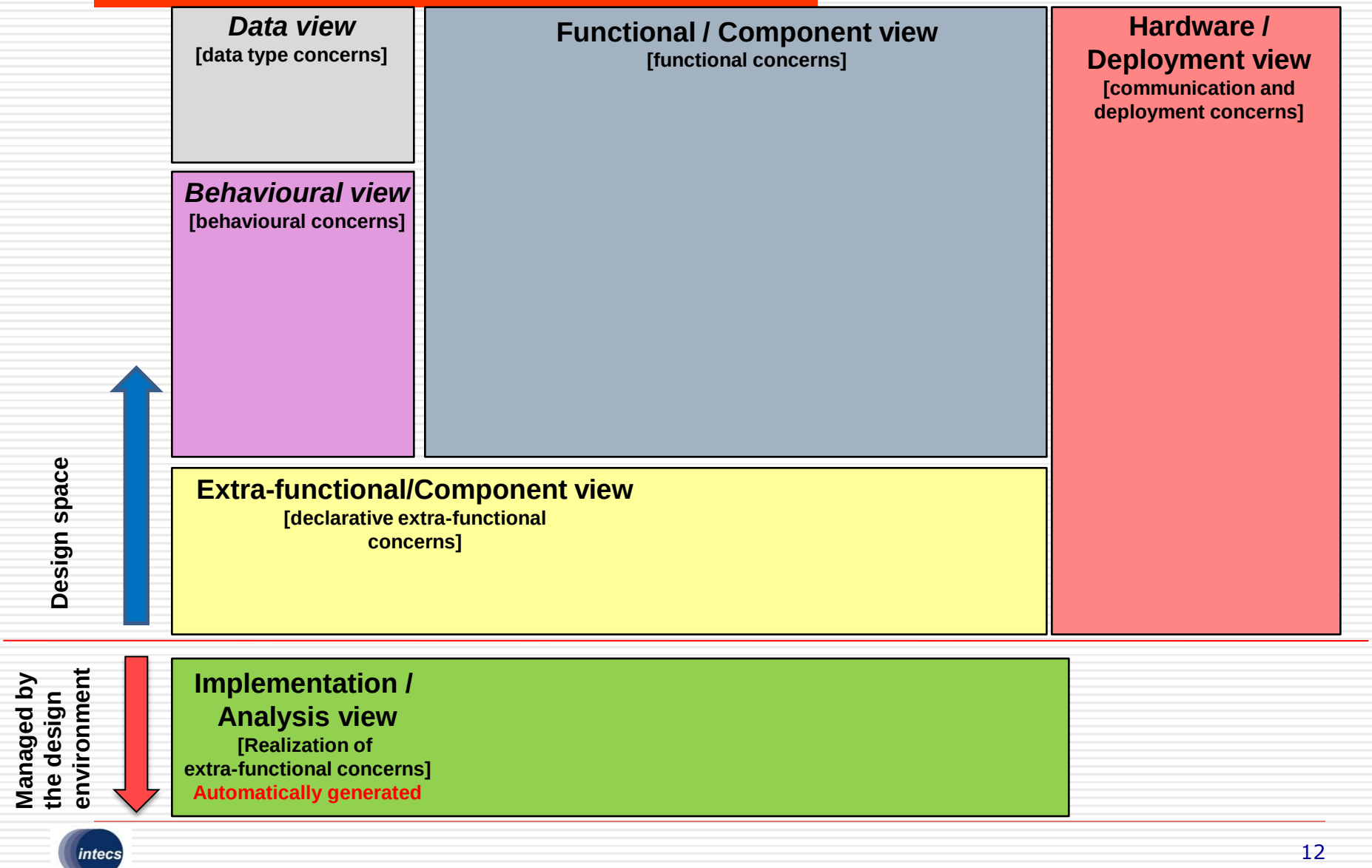
- To assure the reciprocal consistency of PIM and PSM
- To shift the responsibility of correctness from the designer to the transformation designer

6. You change entities' attributes in the PIM as needed and iterate the analysis until the system is satisfactory in all the functional and extra-functional dimensions of interest

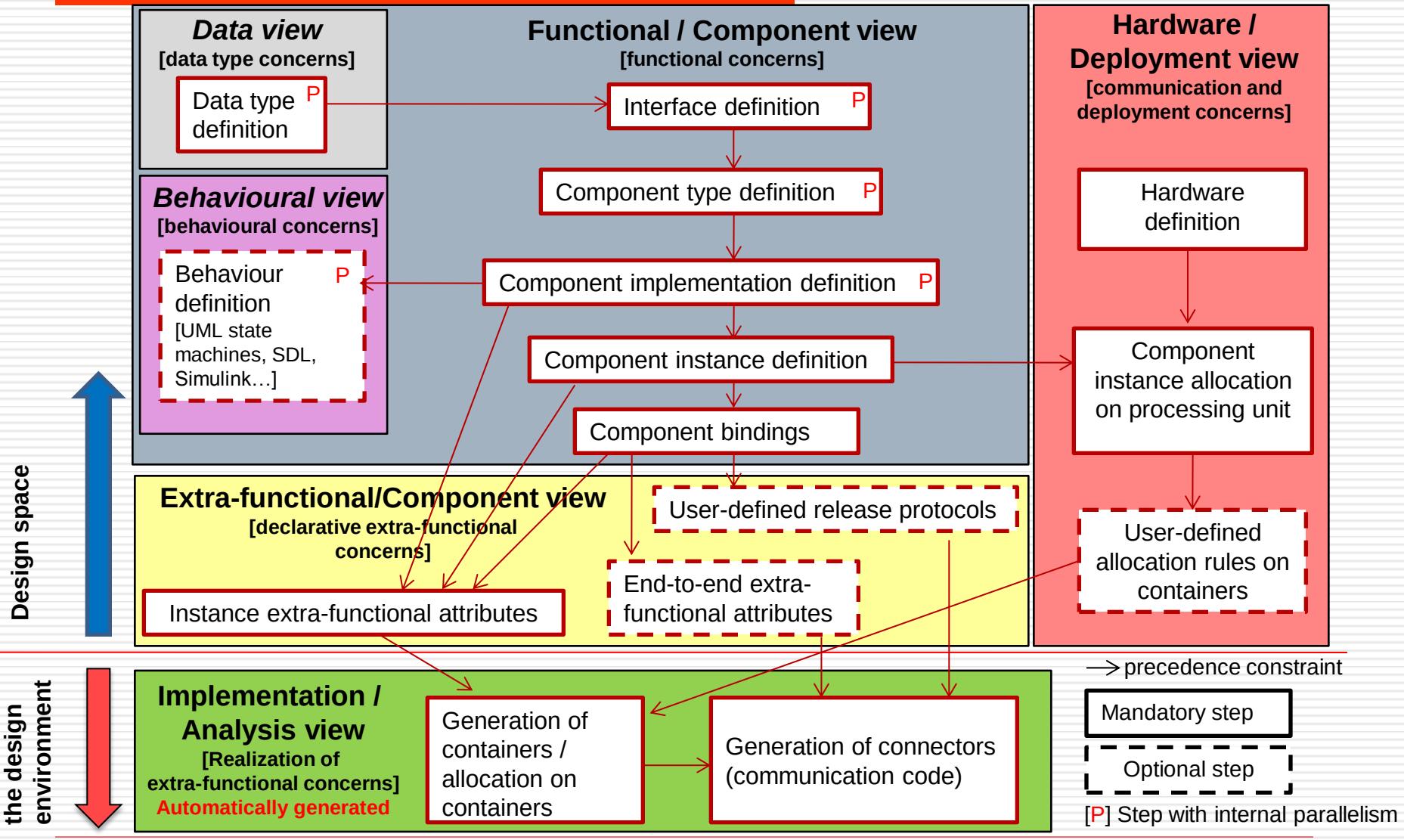
# CHESS Methodology – Views and process

- Multi-view design space
  - "The architectural description of the system is organized in one or more constituents called views" [ISO 42010]
  - Distinct concerns allocated to distinct views
- Incremental and iterative process
  - Incremental by component refinement
  - Iterative by static analysis ⇒ verification ⇒ back propagation cycles
  - Traceability to requirements
  - Automated code generation

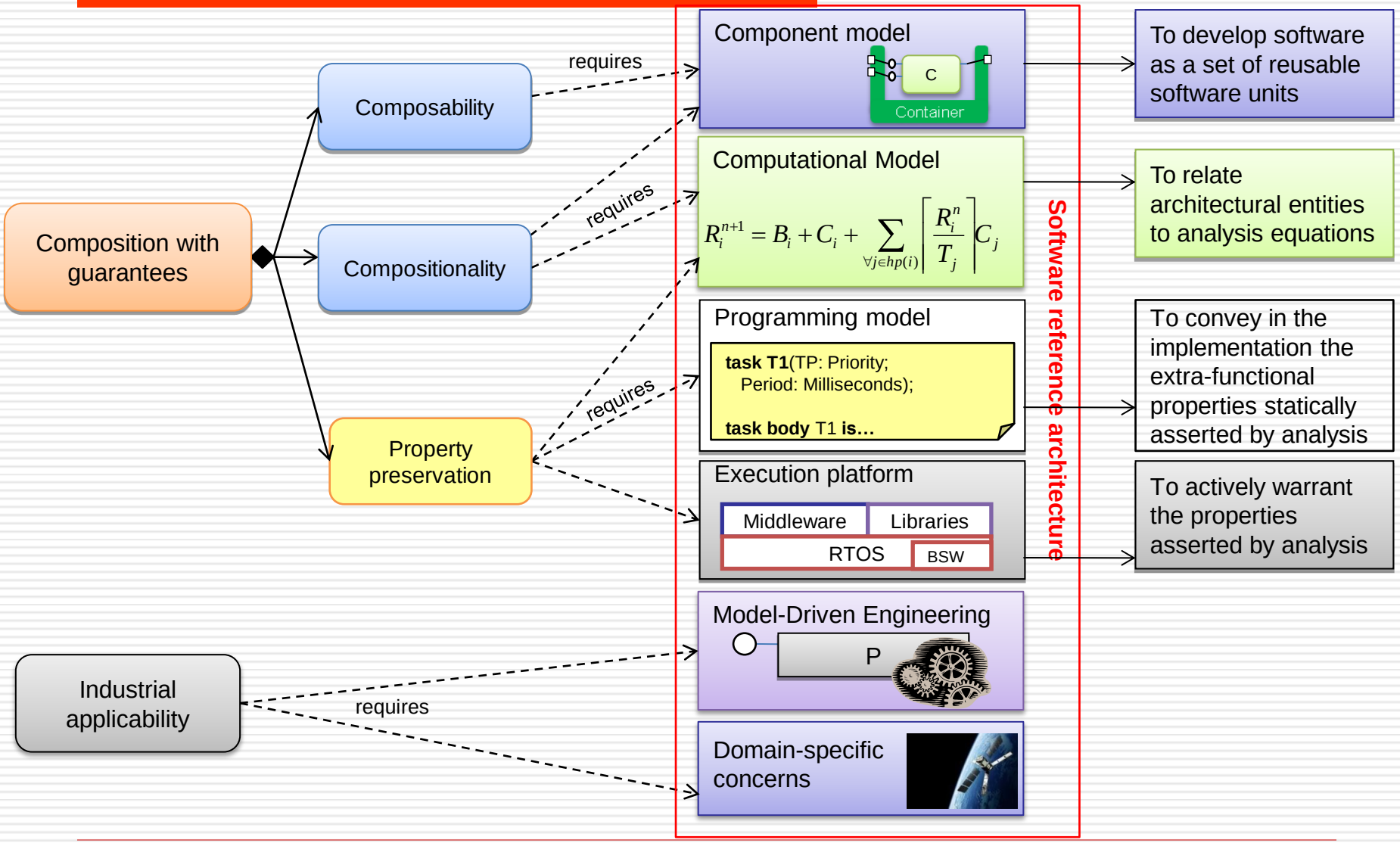
# Design views and design flow



# Design views and design flow



# CHESS reference architecture

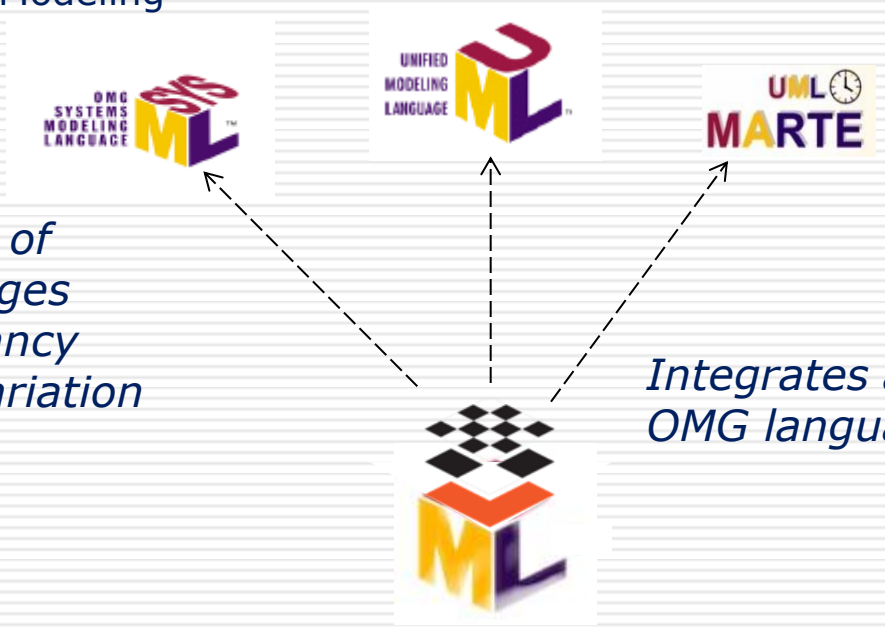


# The CHESS Modeling Language

Standard profile for System (and Requirements) Modeling

Standard Unified Modeling Language

Standard profile for Modeling and Analysis of Real-Time and Embedded Systems




*Imports subsets of standard languages*
*✓ avoid redundancy*
*✓fix semantic variation points*

*Integrates and extends standard OMG languages*

*Introduces a new*
***Dependability Profile***

# CHESS Web Page

www.chess-project.org

Sign Up Sign In Search Chess P

CHESS

*Composition with Guarant*
*Embedded Software Co*

Main CHESS Project Training Videos News Partners Resources Members Contact

ARTEMIS

The CHESS Project is partially funded by the Artemis Joint Undertaking - a public private partnership in the field of embedded systems supported by the European Commission.

The Chess Project

The **CHESS** Project seeks to improve Model Driven Engineering practices technologies to better address safety, reliability, performance, robustnes functional concerns while guaranteeing correctness of component devel composition for embedded systems. System development costs will be extensive use of provable automation and model transformation engines high-integrity applications in the Railway, Space and Telecommunicatior through the identification of feasible solutions to complex system challer development process.

The **CHESS** Project addresses two key embedded system industrial prio reference designs and architectures, and also new design methods and responds to the challenge of reducing the system development cost thro use of provable automation and model transformation engines, specifica the high-integrity application domain. The new **CHESS** technologies will

# CHESS and Polarsys

www.eclipse.org/org/press-release/20111102_polarsys.php

eclipse

Visit other Eclipse Sites

Home Downloads Users Members Committers Resources Projects About Us

About Us »

- Foundation
- Governance
- Legal Resources
- Contact Us

## Polarsys: A New Industry Collaboration to Build Open Source Tools for Safety-Critical Software Development

**Ludwigsburg, Germany – November 2, 2011** - A new open source industry collaboration, called Polarsys, is being created at the Eclipse Foundation to focus on building and maintaining tools for safety critical and embedded system development. Interested parties in Polarsys include Airbus, Astrium Satellites, ATOS, CEA, CS (Communication & Systèmes), Ericsson, IRIT (Institut de recherche en informatique de Toulouse), Inria, Katholieke Universiteit Leuven, Obeo, Universidad Politécnica de Valencia, Tecnalia, Thales, and Xipp. Polarsys will operate as a Eclipse Industry Working Group and be open to any organization interested in participating in the goals of the group.

The goal of Polarsys is to build and maintain an open source tools chain that is used by organizations building safety-critical and software intensive embedded systems. Industries such as aerospace, defense, transportation, telecommunications, energy and healthcare require development tool chains with a number of specific requirements:

# CHESS On-going Extensions

- ESA funded FoReVer study
  - A Component-based Contract-based approach at system level
  - The system is described in terms of architectural components
  - Components are refined into lower levels as black boxes until they are refined
  - Formalize requirements/properties of system and components in terms of component contracts
  - Formal verification of component contracts
    - contract implementation
    - step wise refinement of contracts from System down to SW
- The extensions will be further elaborated within the SafeCer ARTEMIS project.

# The new CONCERTO Project

- "Guaranteed Component Assembly with Round Trip Analysis for Energy Efficient High-integrity Multi-core Systems"- ARTEMIS JU Call 2012
- Recently started to extend the CHESS project achievements with
  - Wider coverage of industrial domains: medical, offshore platforms, avionics other than telecom, space, and automotive
  - Extensions to multicore platforms
  - Model execution

Thank you for your attention

# QUESTIONS?