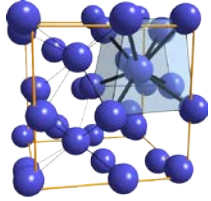


FoReVer

The FoReVer MBSE Solution for System Composition Correctness Analysis



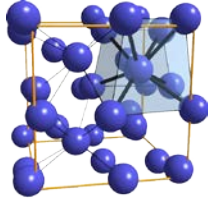
Silvia Mazzini, Intecs



The FoReVer Project

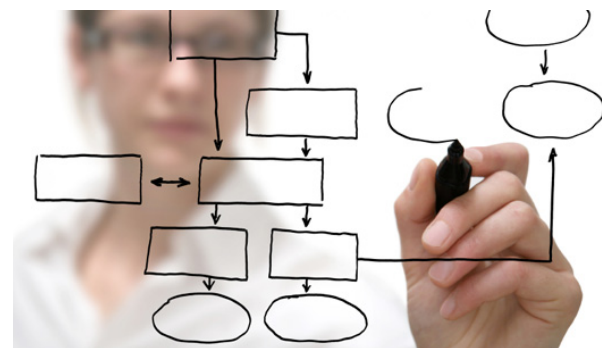
- Functional Requirements and Verification Techniques for the Software Reference Architecture (FoReVer) is an ESA/ESTEC project
 - Consortium led by Intecs
 - Partners
 - Thales Alenia Space (Cannes)
 - Fondazione Bruno Kessler (Trento)
 - Running in the period January 2012 - May 2013

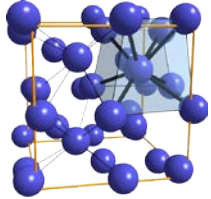




The FoReVer Formal Methodological Framework

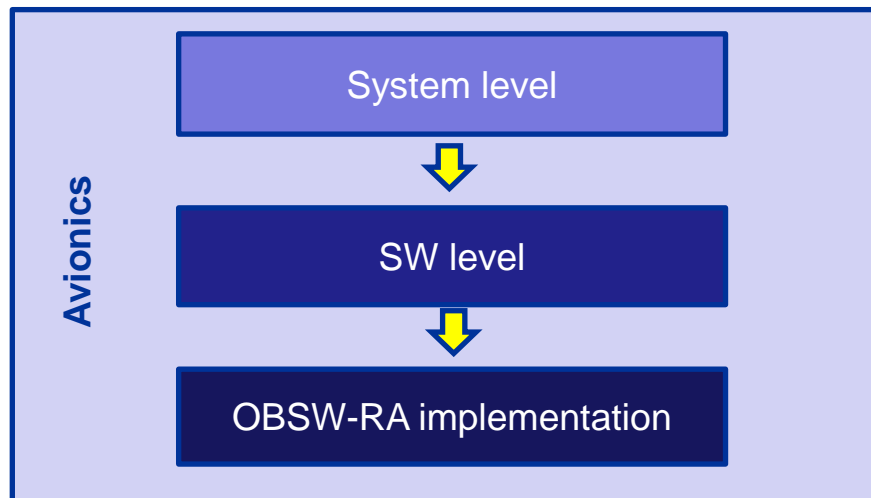
- Model Based System Engineering (MBSE)
Methodology and Technology to support system avionic development across phases O, A, B, and C
- Early apply Formal Verification techniques in the context of MBSE for
 - Specification of requirements
 - Formal properties
 - Formal reasoning
 - Formal verification of properties
 - Step wise refinement from System down to SW

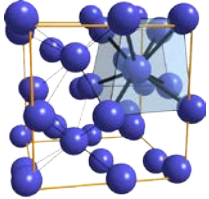




Software Reference Architecture

- Exploit the Software Reference Architecture (SRA) concepts from the context of the SAVOIR-FAIRE ESA initiative (COrDeT-2)
 - Refinement of avionics System level properties down to SW level and then to SW implementation on top of the on-board software reference architecture (OBSW-RA)

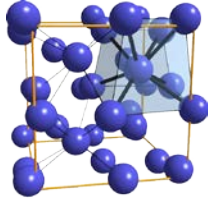




FoReVer Ingredients

Systematic approach to formal verification of space avionics systems properties from the early development phases

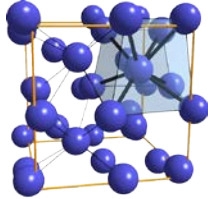




MBSSE

- **Model-Based methodology to support the Space System Engineering (from SSFRT)**
 - based on SysML: graphical, model based support for system modeling of requirements, functional decomposition, behavior, architecture, step-wise refinement and traceability
- property formalization, formal verification, step-wise refinement with assume-guarantee reasoning for modeling the avionics at system and software level

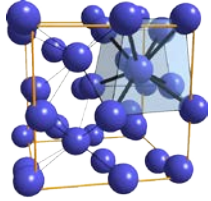
FoReVer



NuSMV3/OCRA

- **NuSMV3** provides formal techniques for:
 - modeling of the avionics for system and software co-engineering and
 - property verification
- **OCRA** package provides methodological framework for:
 - stepwise refinement verification
 - assume-guarantee reasoning
 - traceability
- formal means for verification, step-wise refinement with assume-guarantee reasoning applied to MBSE

FoReVer



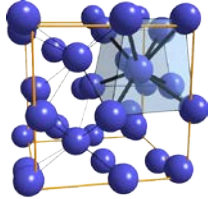
- Model-driven Component Based methodology for high integrity software development from the ARTEMIS CHESS/CONCERTO projects

- modeling of requirements, traceability and properties
- formal verification techniques for **non-functional properties** of software (real-time and dependability)
- correctness-by-construction (by automated code generation)
- SRA component model



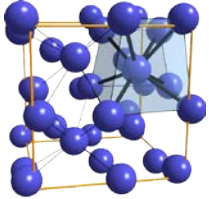
FoReVer

- system level support
- system and software co-engineering support
- stepwise refinement with integration of formal verification means

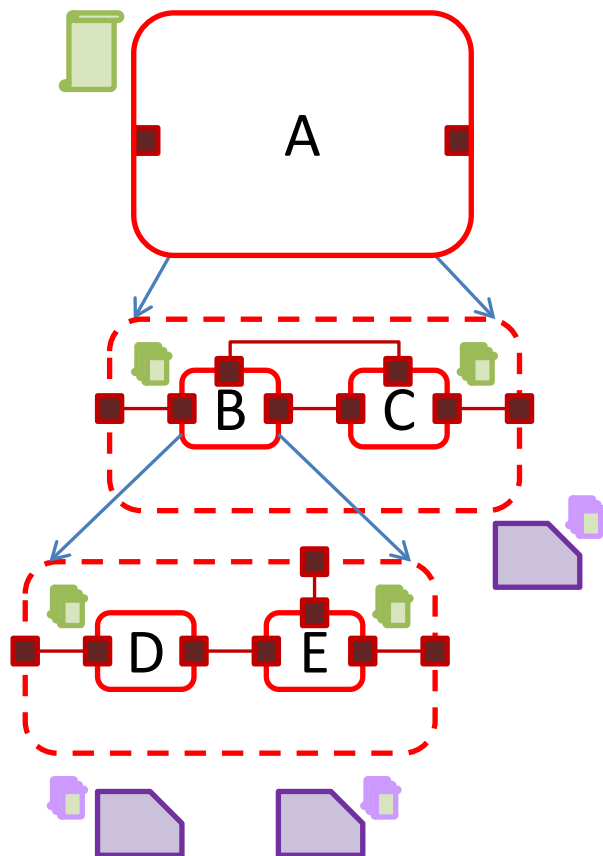


The FoReVer Approach

- **Component-based**
 - The system is described in terms of architectural components with their **well-defined interfaces and related properties**
 - Components are refined into lower levels as black boxes until they are refined
- **Contract-based**
 - Formalize properties of system and components in terms of **component contracts**
 - Formal verification of **contract refinement**



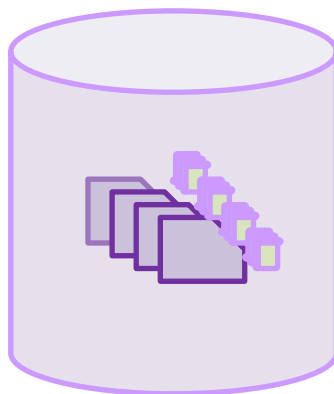
Contract Based Approach Compositional Verification



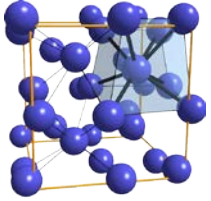
Component's Contract

Contract Refinement

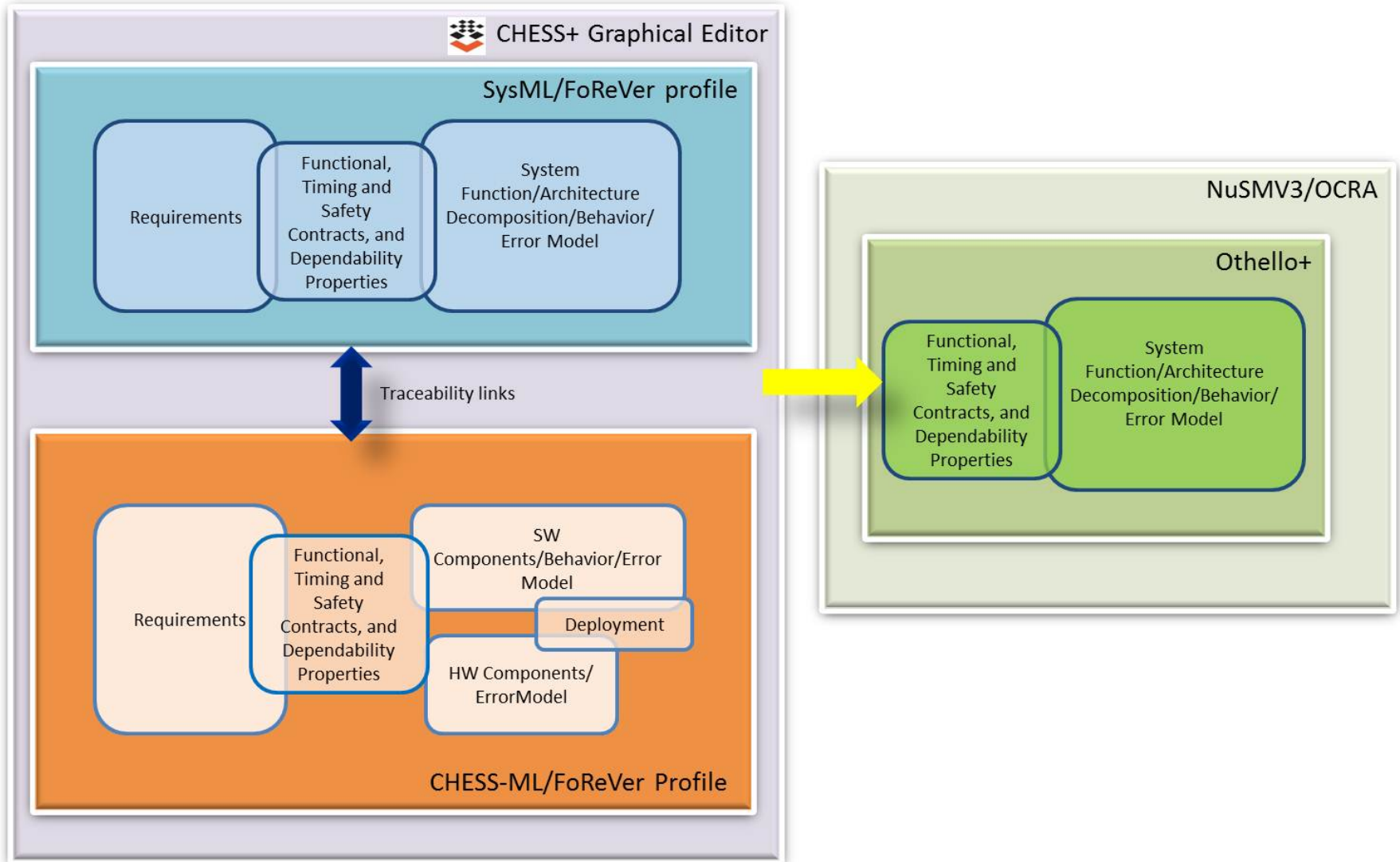
Library of Components with
Contracts

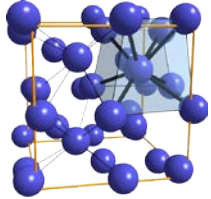


If the refinement steps are proven correct, then any implementation of the leaf components that satisfies the component contracts can be used to implement the system.



The FoReVer Integrated Environment





The FoReVer Toolset At Work

UML/SysML - UseCase_GB2_v4/UseCase_GB2_v4/functional_step2/functional_step2.uml_modified.di - Eclipse Platform

File Edit Diagram Navigate Search Papyrus Project Scripts CHESS Window Help

Tahoma 8 B

Code generation
 Activate ExtraFunctionalView
 Activate RailwayView
 Activate DependabilityView
 Purge user model
 Build Instances
 Filters
 Analysis

Schedulability (VERDE)
 Dependability
 Formal Verification
 Check Contracts Refinement Analysis

Palette
 Nodes
 Edges
 FoReVer
 Contract (Class)
 ContractProperty (Property)
 FormalProperty (Constraint)
 System (ComponentImplementation)
 SRAComponent (ComponentImplementation)

«subSystem»
 PerformAttitudeControl
 + goToSafeMode : Boolean
 + safeModeEnabled : Boolean
 «contractProperty» {RefinedBy=[sensesundir.sunDir_Present, pe...}

«contract»
 CanGoToSafe
 «formalProperty»
 Assumption
 {true}
 «formalProperty»
 Guarantee
 {always(goToSafeMode implies safeModeEnabled)}

«contract»
 SunDirection_AlwaysPresent
 Assumption
 {TRUE}

«subSystem»
 SenseSunDirection
 «contractProperty» + sunDir_Present

OCRA

```

COMPONENT PerformAttitudeControl
INTERFACE
INPUT PORT goToSafeMode : boolean;
OUTPUT PORT safeModeEnabled : boolean;

CONTRACT canGoToSafe
assume:
goToSafeMode=true;
guarantee:
safeModeEnabled=true;

REFINEMENT
SUB sensespacecraftRate : SenseSpacecraftRate;
SUB performcontrolalgorithms : PerformControlAlgorithms;
SUB activatethrusters : ActivateThrusters;
SUB sensesundirection : SenseSunDirection;

DEFINE performcontrolalgorithms.sunDirection_IsPresent := sensesundirection.sunDirection_IsPresent;
DEFINE performcontrolalgorithms.sunDirection := sensesundirection.sunDirection;
DEFINE performcontrolalgorithms.sensedSpeed_IsPresent := sensespacecraftRate.sensedSpeed_IsPresent;
DEFINE performcontrolalgorithms.goToSafeMode := goToSafeMode;
DEFINE activatethrusters.torqueCommand := performcontrolalgorithms.torqueCommand;
DEFINE safeModeEnabled := performcontrolalgorithms.safeModeEnabled;
DEFINE performcontrolalgorithms.sensedSpeed := sensespacecraftRate.sensedSpeed;

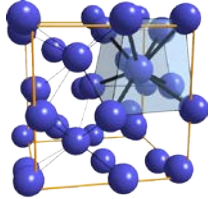
CONTRACT canGoToSafe REFINEDBY sensesundirection.SenseSunDirection_sunDirection_AlwaysPresent, performcontrolalgorithms.engageSafeMode

COMPONENT PerformControlAlgorithms
INTERFACE
  
```

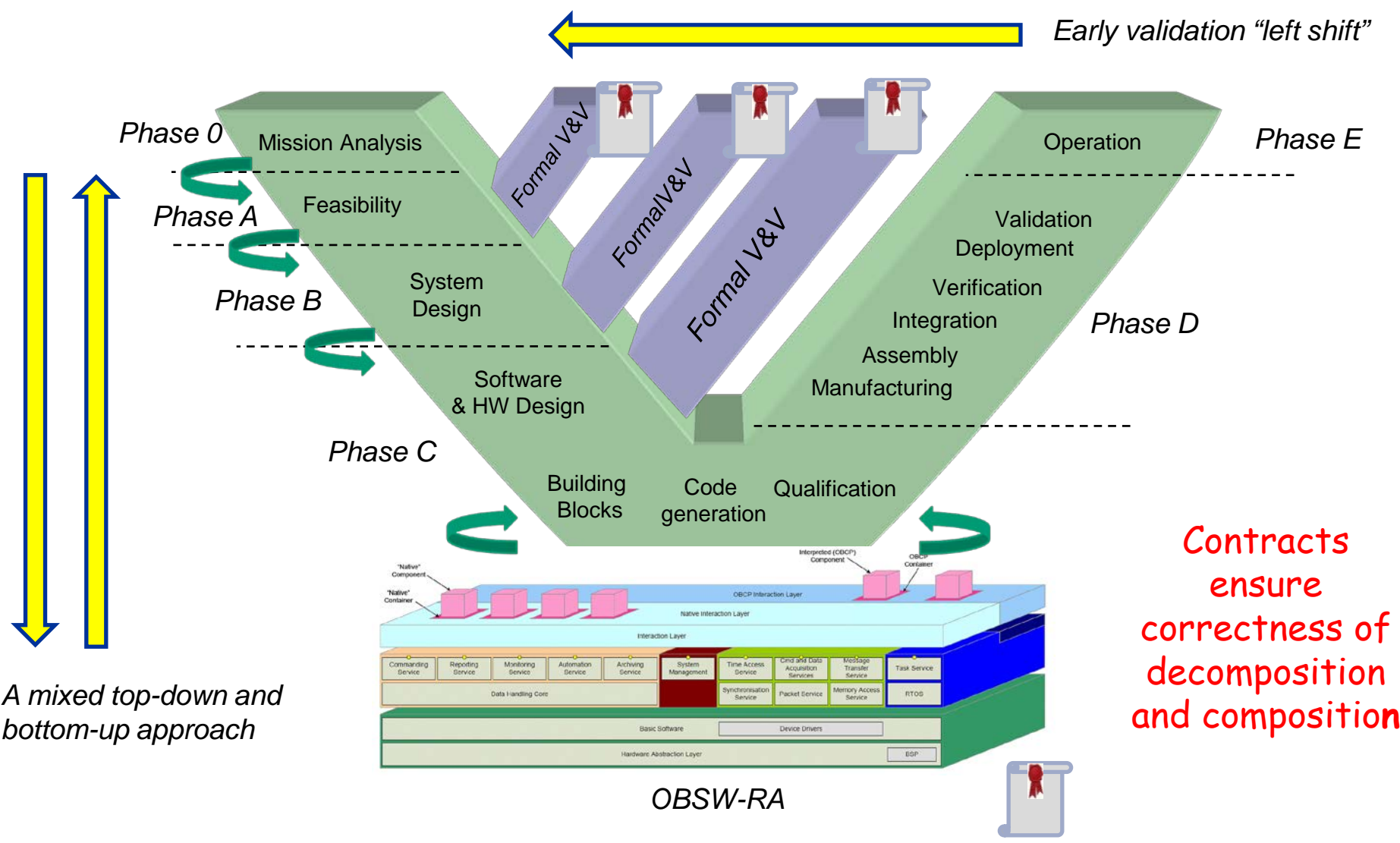
*** Copyright (c) 2010-2012, Fondazione Bruno Kessler
 *** This version of ocra is linked to NuSMV 2.5.devel.
 *** For more information on NuSMV see <http://nusmv.fbk.eu>
 *** or email to <nusmv-users@list.fbk.eu>.
 *** Copyright (C) 2010 by Fondazione Bruno Kessler
 *** This version of ocra is linked to the CUDD library version 2.4.1
 *** Copyright (c) 1995-2004, Regents of the University of Colorado
 *** This version of ocra is linked to the MiniSat SAT solver.
 *** See http://www.cs.chalmers.se/Cs/Research/FormalMethods/MiniSat
 *** Copyright (c) 2003-2005, Niklas Een, Niklas Sorensson

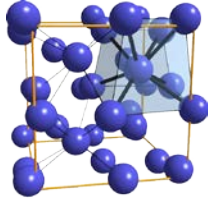
Checking "CONTRACT canGoToSafe REFINEDBY sensespacecraftRate.SenseSpacecraftRate_sensedSpeed_AlwaysPresent, senseundirection.SenseSunDirection_sunDirection_AlwaysPresent"... [OK]
 Checking the correct implementation of "canGoToSafe"... [OK]
 Checking the correct environment of "sensesundirection.SenseSunDirection_sunDirection_AlwaysPresent"... [OK]
 Checking the correct environment of "performcontrolalgorithms.engageSafeMode"... [OK]

Profile
 Appearance
 Is abstract true false
 Is leaf true false



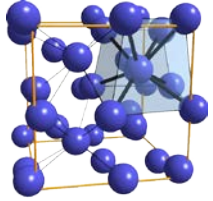
The Vision for a Model Based Systematic Approach





Conclusion

- A MBSSE solution for system composition correctness analysis
- Integration in the CHESSE toolset
 - An front-end for the COMPASS-STAR technology in the OMG UML MBSE world
 - Availability as open-source in the Polarsys/Eclipse open community
 - Increase the potential for other R&D extensions and user experimentation/maturation



Thank you for
your attention!
Questions?