V&V Activities within two Brazilian Space Research Institutes

Miriam C. Bergue Alves*
Valdivino Alexandre de Santiago Júnior+
Nandamudi L. Vijaykumar+

NASA IV&V Workshop
Morgantown, WV
September 11-13, 2012

+ National Institute for Space Research - INPE
São José dos Campos, SP, Brazil

* Institute of Aeronautics and Space - IAE
São José dos Campos, SP, Brazil

Objective

This presentation relates some of the research initiatives of the Institute of Aeronautics and Space (IAE) and the National Institute for Space Research (INPE) with respect to the use of formal methods for the improvement of their V&V activities within the software development life cycle.

Outline

– Brazilian Space Program

– Presentation of IAE

– V&V Projects at IAE: Software Engineering Lab

– Presentation of INPE

– V&V Activities (Products/Projects) at INPE: CEA/LAC

– Conclusions

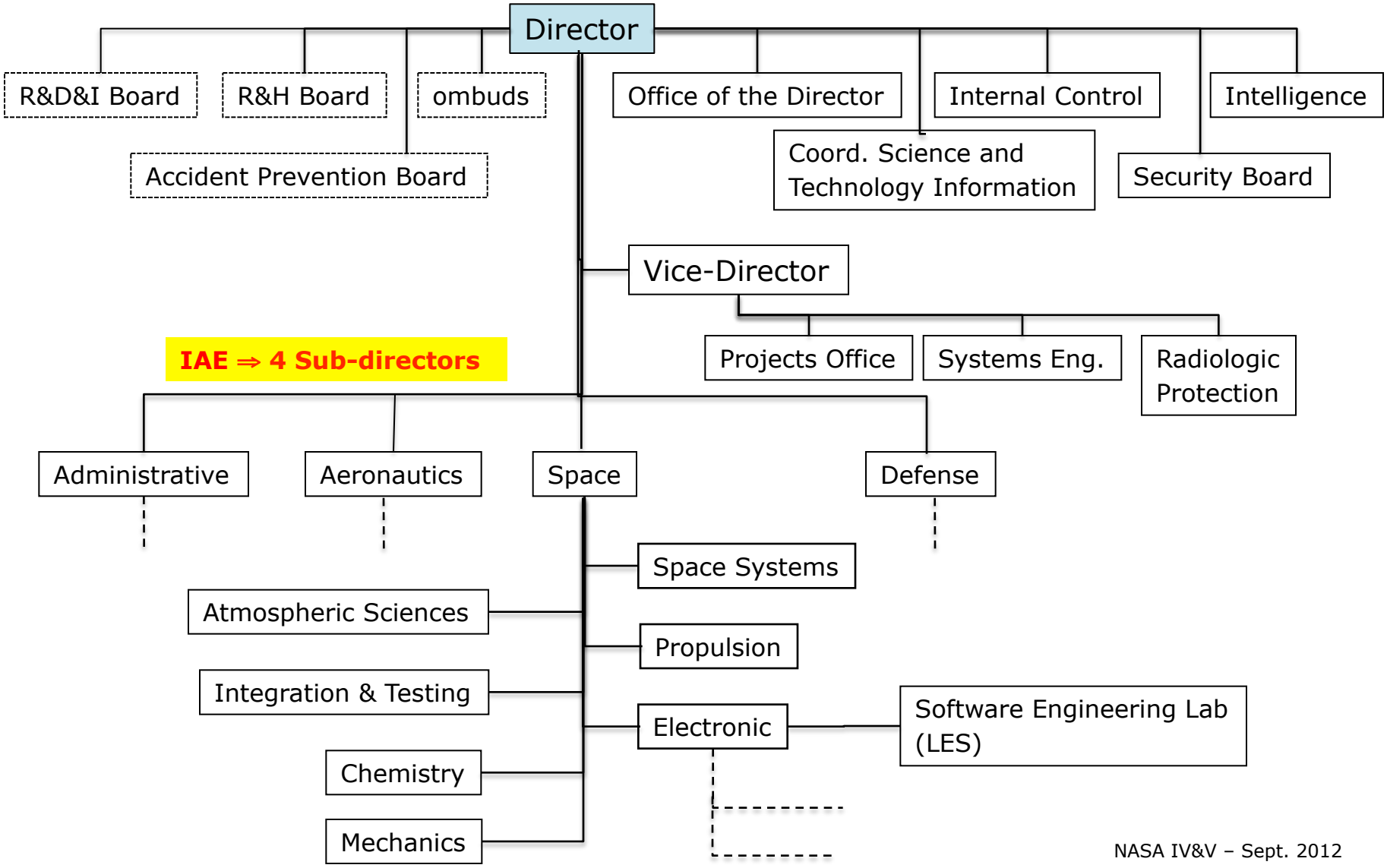
Brazilian Space Program

–Rocketry: launching and sounding rockets (IAE)

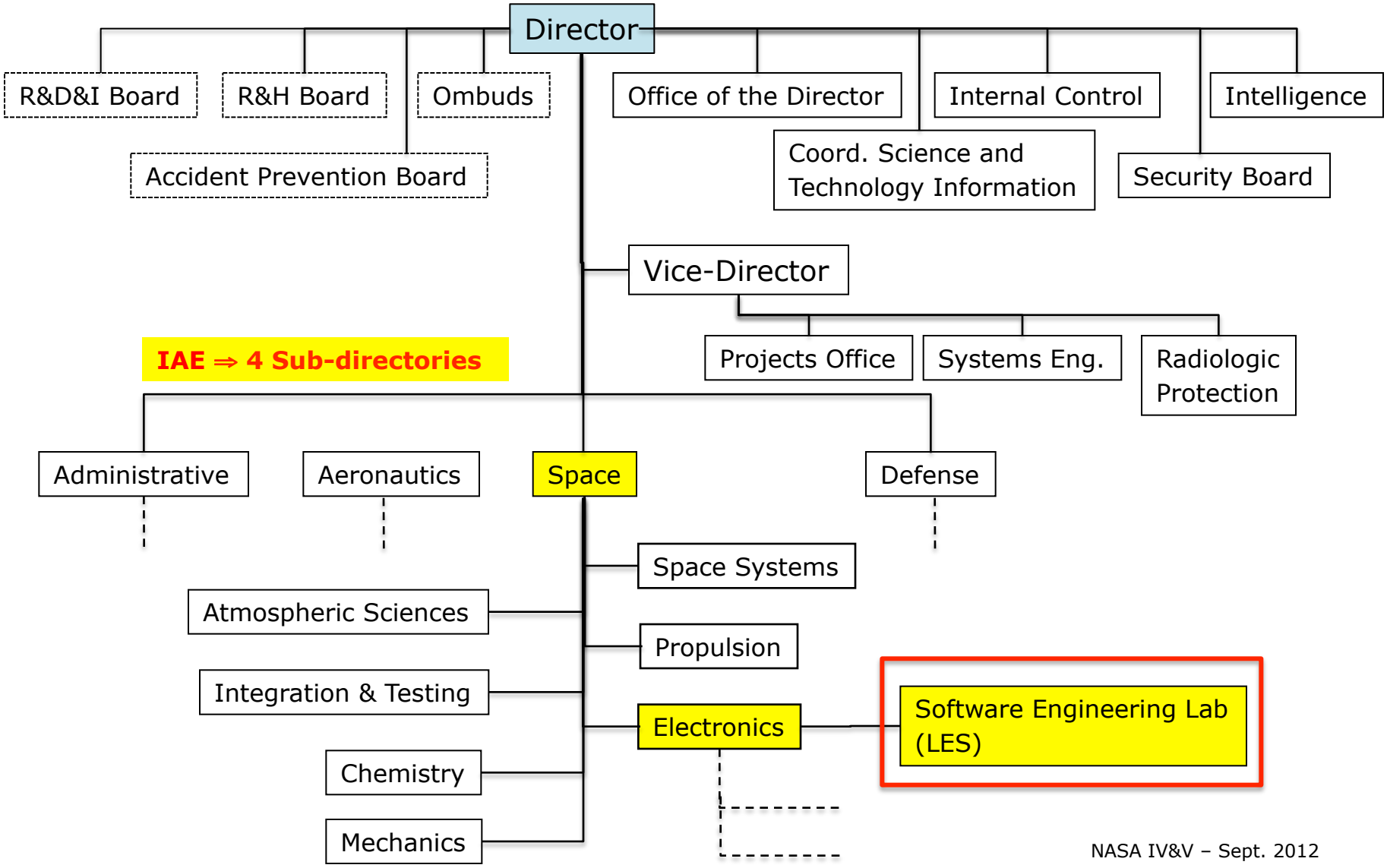
–Space exploration: satellites (INPE)

– Launch sites: Alcantara Launch Center and Barreira do Inferno Launch Center (DCTA)

IAE's Organization Chart



IAE's Organization Chart



V&V Projects at IAE

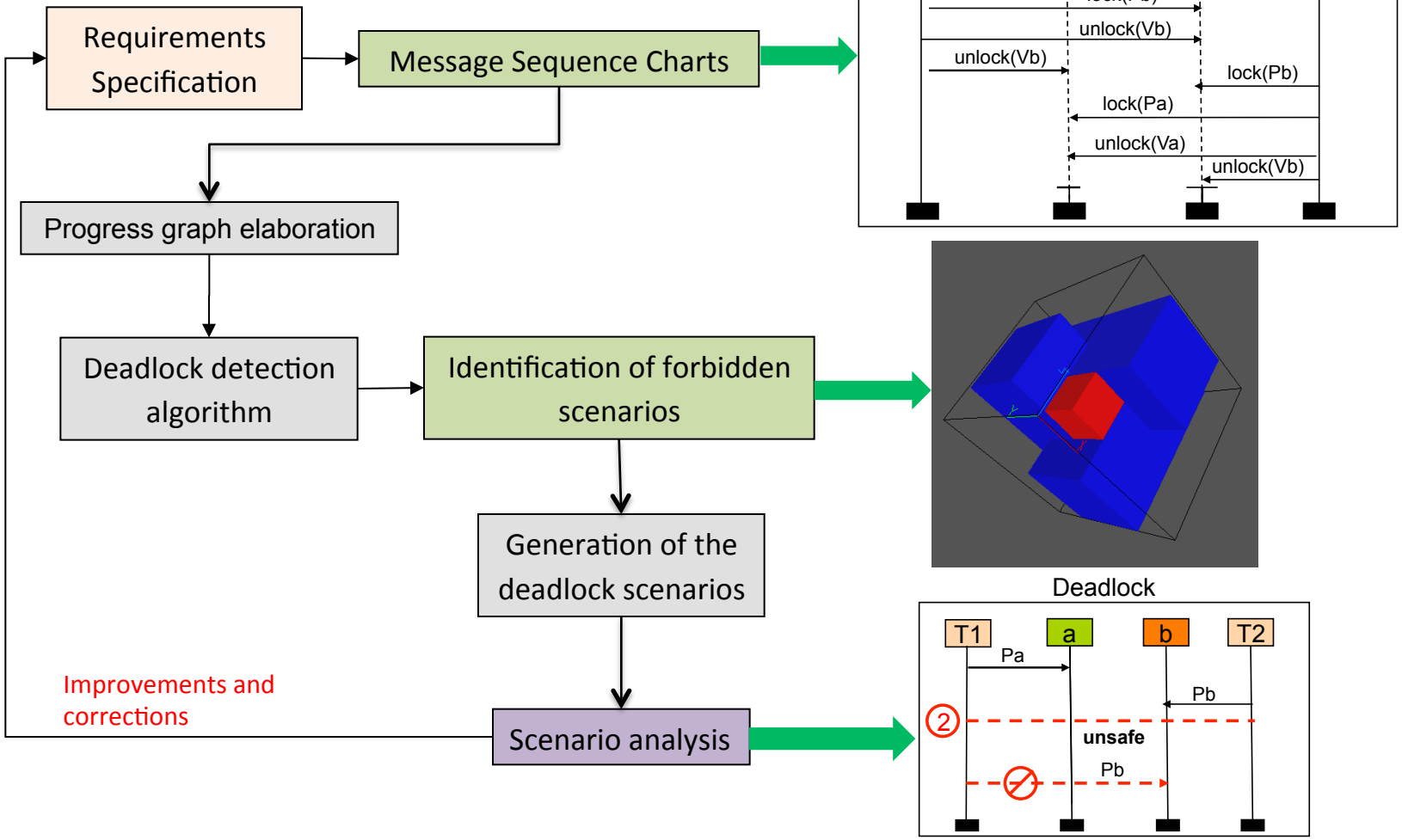
- Use of topology for verification of deadlocks in concurrent systems

  – This project proposes a method that maps scenario-based specifications of concurrent systems, represented formally by MSCs (Message Sequence Charts), to a topological space. This mapping allows to formally verify these specifications for deadlock scenarios.

  – A simple “proof-of-concepts” prototype was constructed.

V&V Projects at IAE

- Use of topology for verification



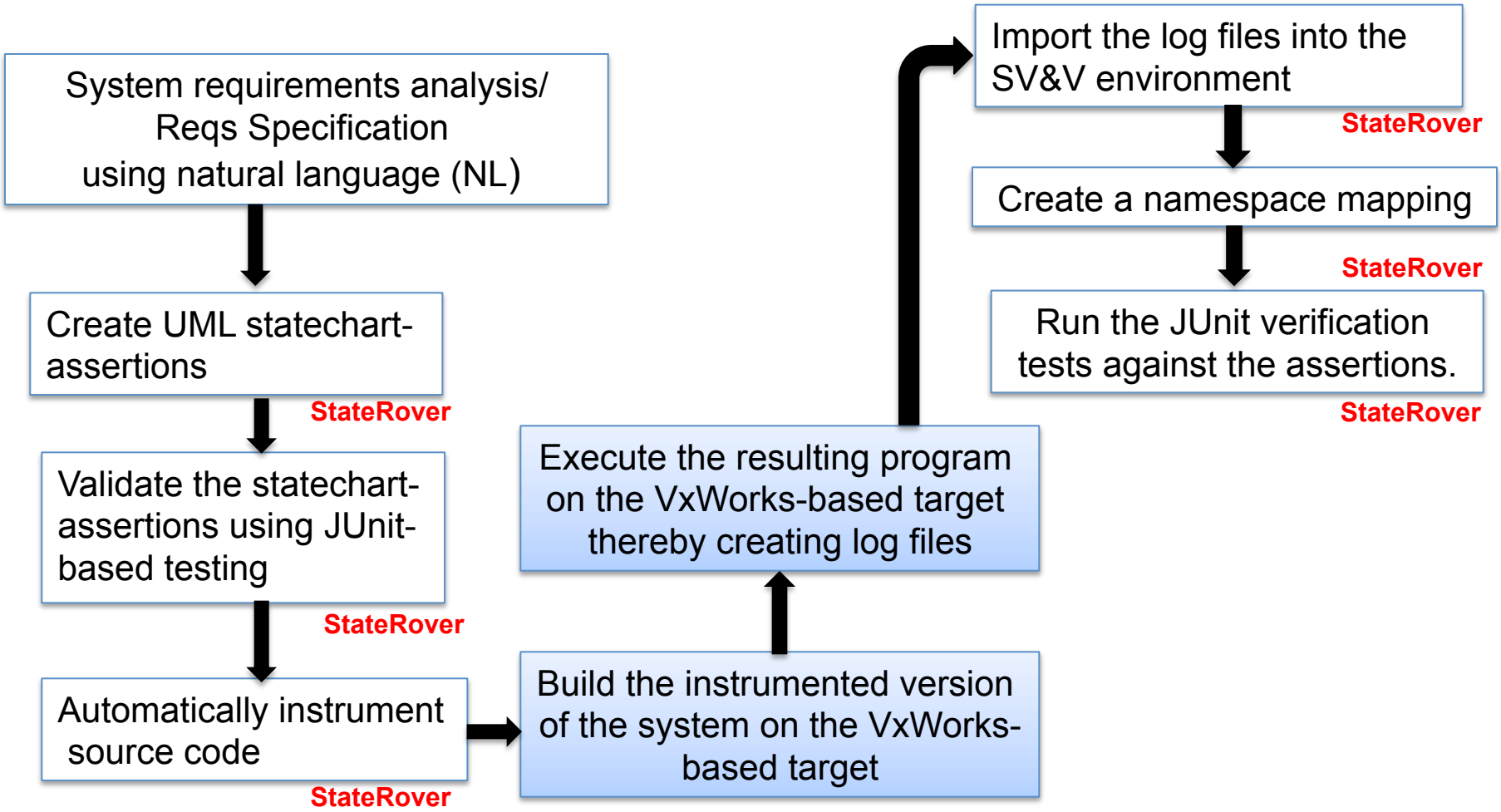
V&V Projects at IAE

- Use of statechart-assertions for requirements specification, validation and verification

  – Formal computer-aided validation and verification of critical time-constrained requirements of the Brazilian Satellite Launcher flight software. It included the entire specification, validation, and verification process based on UML statechart-assertions and log files.

V&V Projects at IAE

- The SV&V process



V&V Projects at IAE

- SV&V – Some results

Validation Tests	Verification Tests
220 tests (around 5 tests per assertion) 220 JUnit classes - 1 JUnit class per test	4 log files (4 tests per assertion) 4 JUnit class- 1 JUnit class per log file
132 success scenarios (around 60% of the scenarios)	31 assertions passed in all tests (around 70% of the assertions)
88 scenarios expect an assertion failure (around 40% of the scenarios)	13 assertions failed at least in one test (around 30% of the assertions)

V&V Projects at IAE

- Use of timed automata for model verification

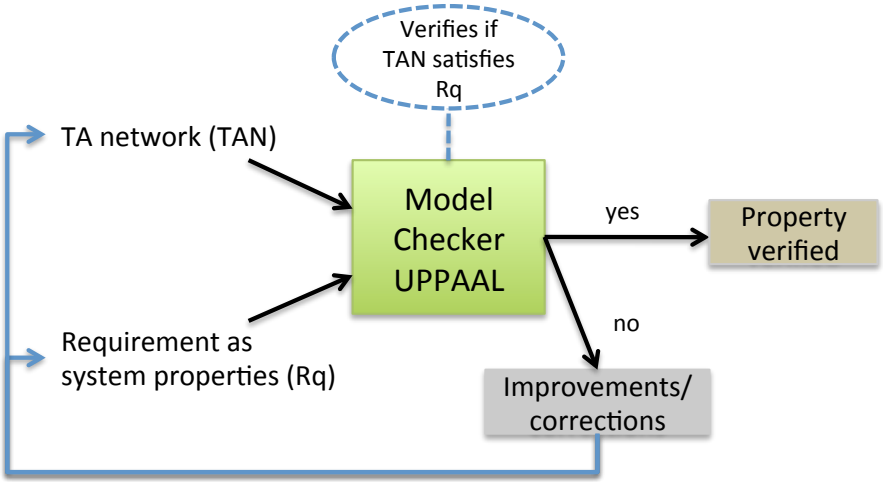
  - A case study of a legacy space flight software system is being conducted, where the flight control and the flight events sequence chain of a satellite launcher are under study.

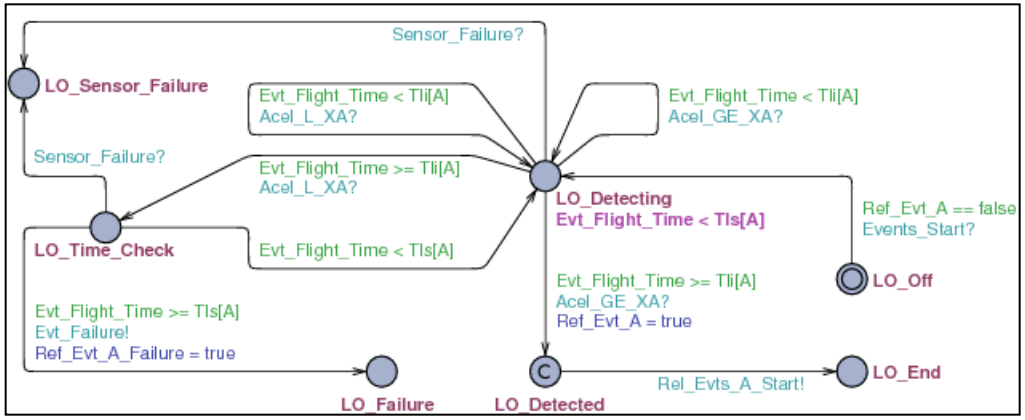
  - Use of model checking and a timed automata (TA) network to model the original requirements specification, incorporating new mission requirements and modifications.

  - Improve reliability in legacy systems evolution.

V&V Projects at IAE

- Use of timed automata for model verification





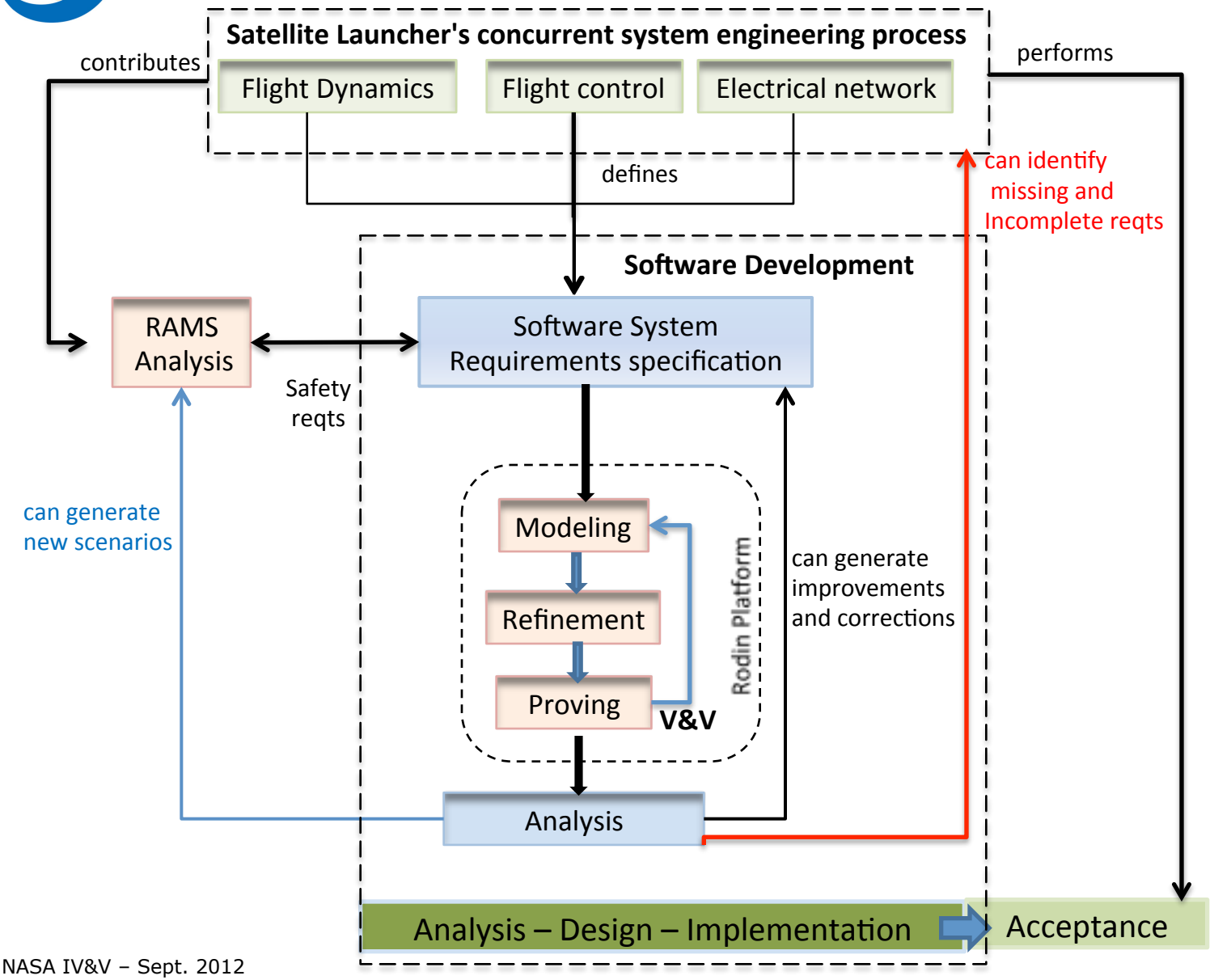
V&V Projects at IAE

- Use of Event-B and Rodin Platform

  – The UML-B and Event-B language are being used for the models elaboration of a case study that involves the control of the first stage of a launch vehicle, with the support of the computer-aided tool Rodin Platform (Rigorous Open Development Environment for Complex Systems).

  – The work is at its initial phases of creating and refining the models, with emphasis to the improvement of the system dependability.

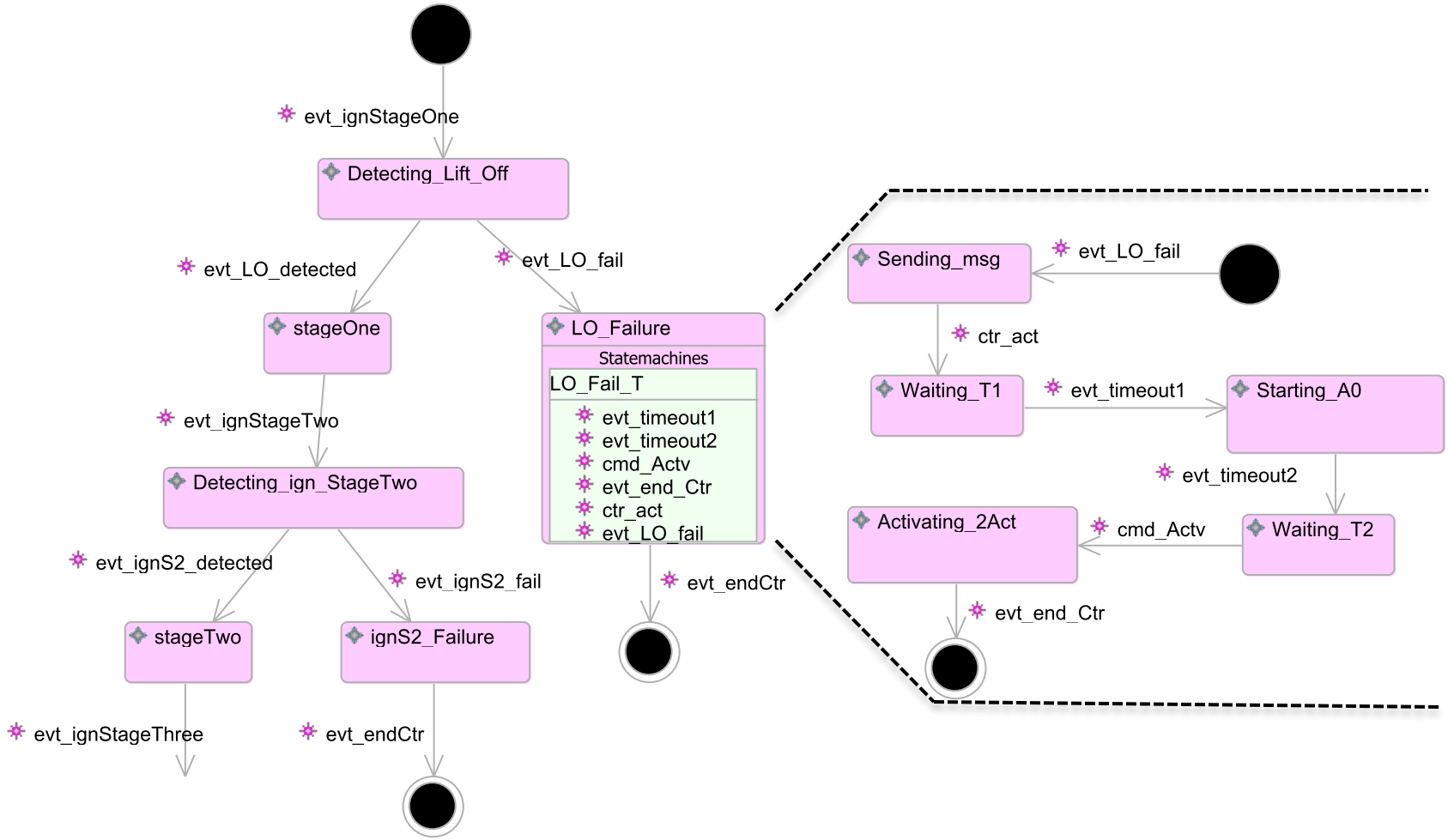
V&V Projects at IAE



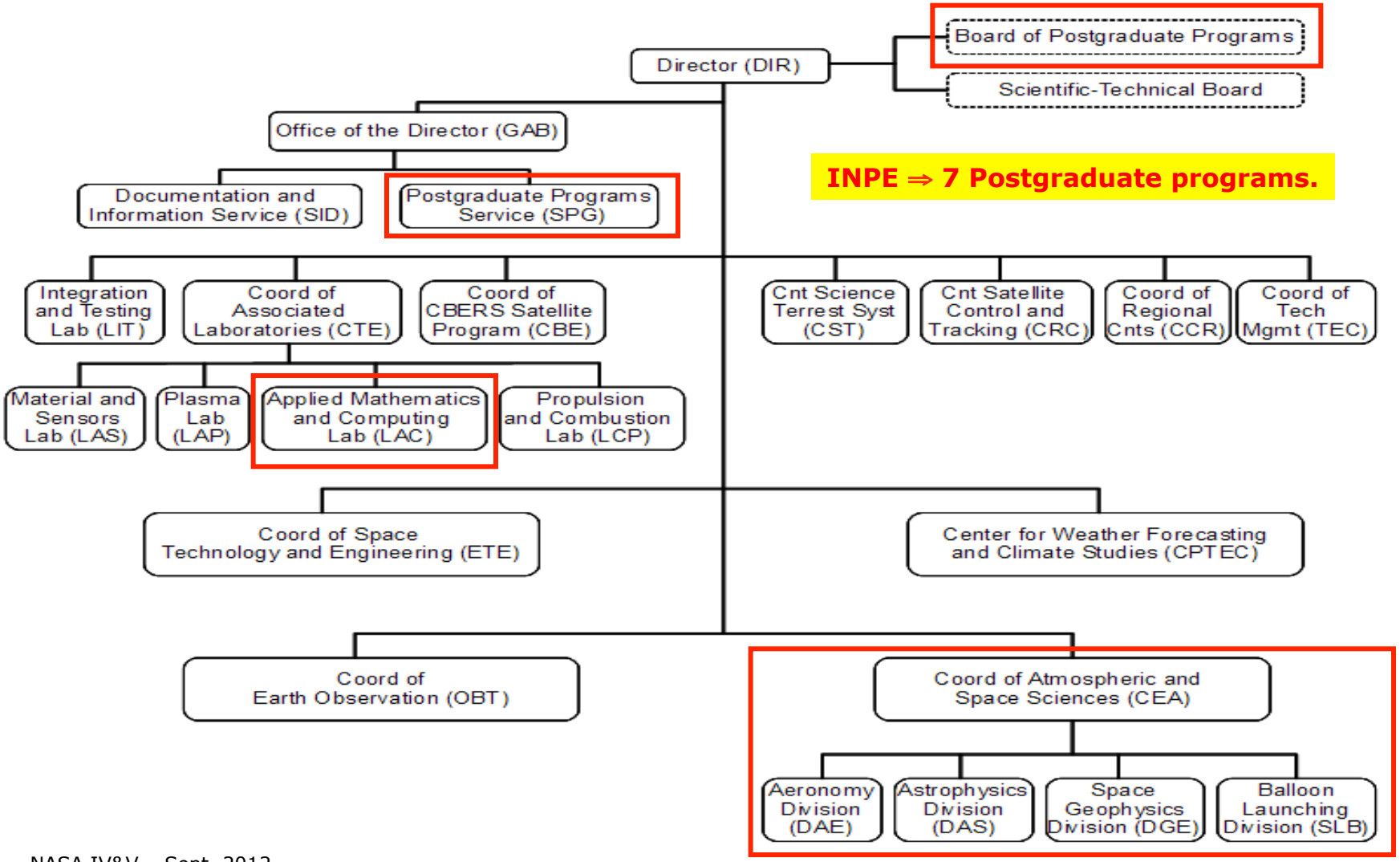
Event-B and Rodin Platform: the process

V&V Projects at IAE

- Use of Event-B and Rodin Platform: example



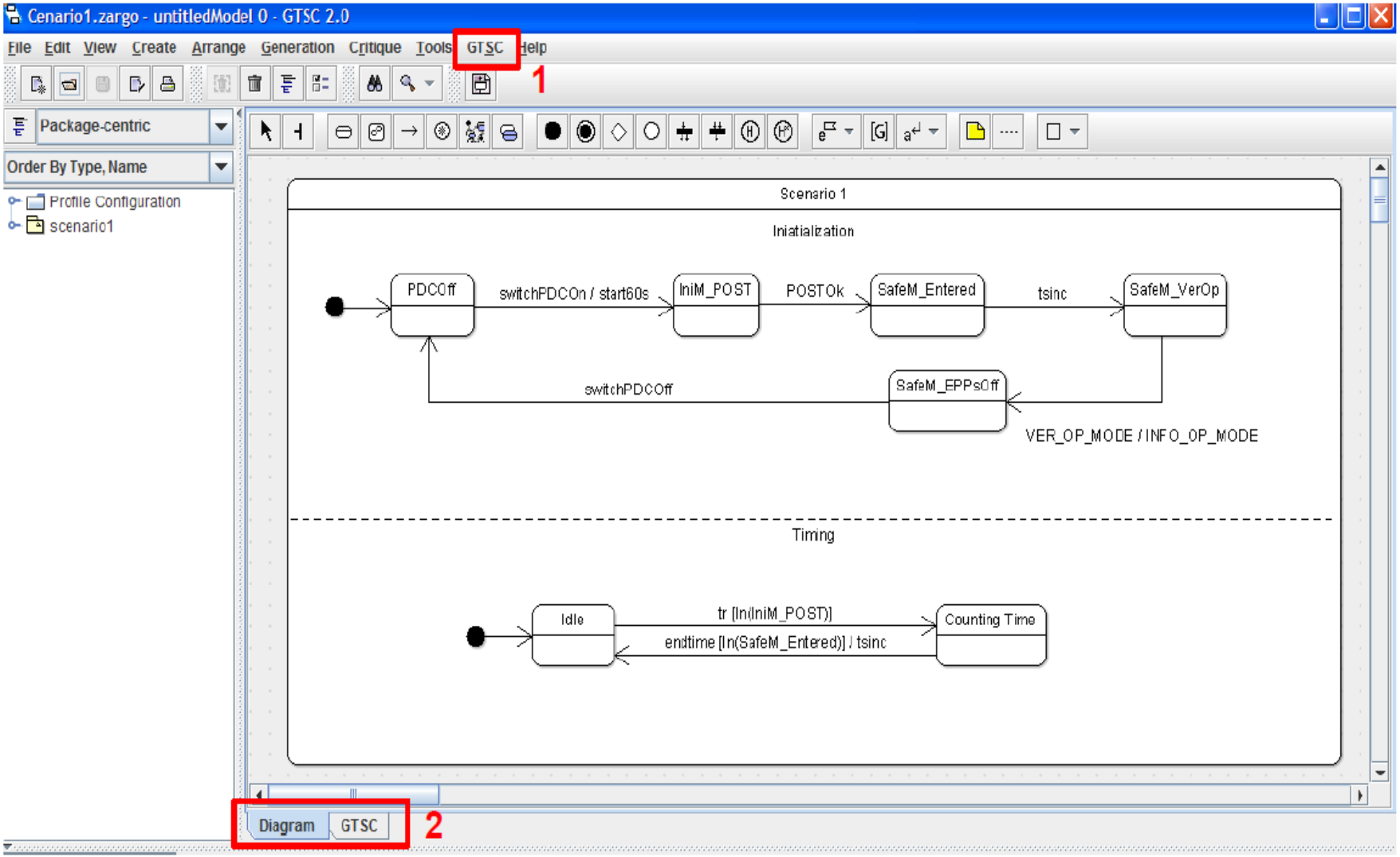
INPE's Organization Chart



V&V Activities at INPE: Products

- Automated Test Case Generation based on Statecharts (GTSC):
  - Model-based test case generation based on Statecharts ⇒ four test criteria (all-transitions, all-simple-paths, all-paths-k-C0-configuration, all-paths-k-configurations) from the Statechart Coverage Criteria Family (SCCF);
  - Model-based test case generation based on FSM ⇒ three test criteria (DS, UIO, H-switch cover) where one (H-switch cover) is a new test criterion.

GTSC 2.0: Main Interface



V&V Activities at INPE: Products

WEB - PerformCharts

Log-out Project in use: **APEX**

Project | Statechart | PerformCharts | FSM | Condado | Test Case Gen. | Help

Quinta, 14 de Junho de 2012 - 11:46 AM

Admin Mode

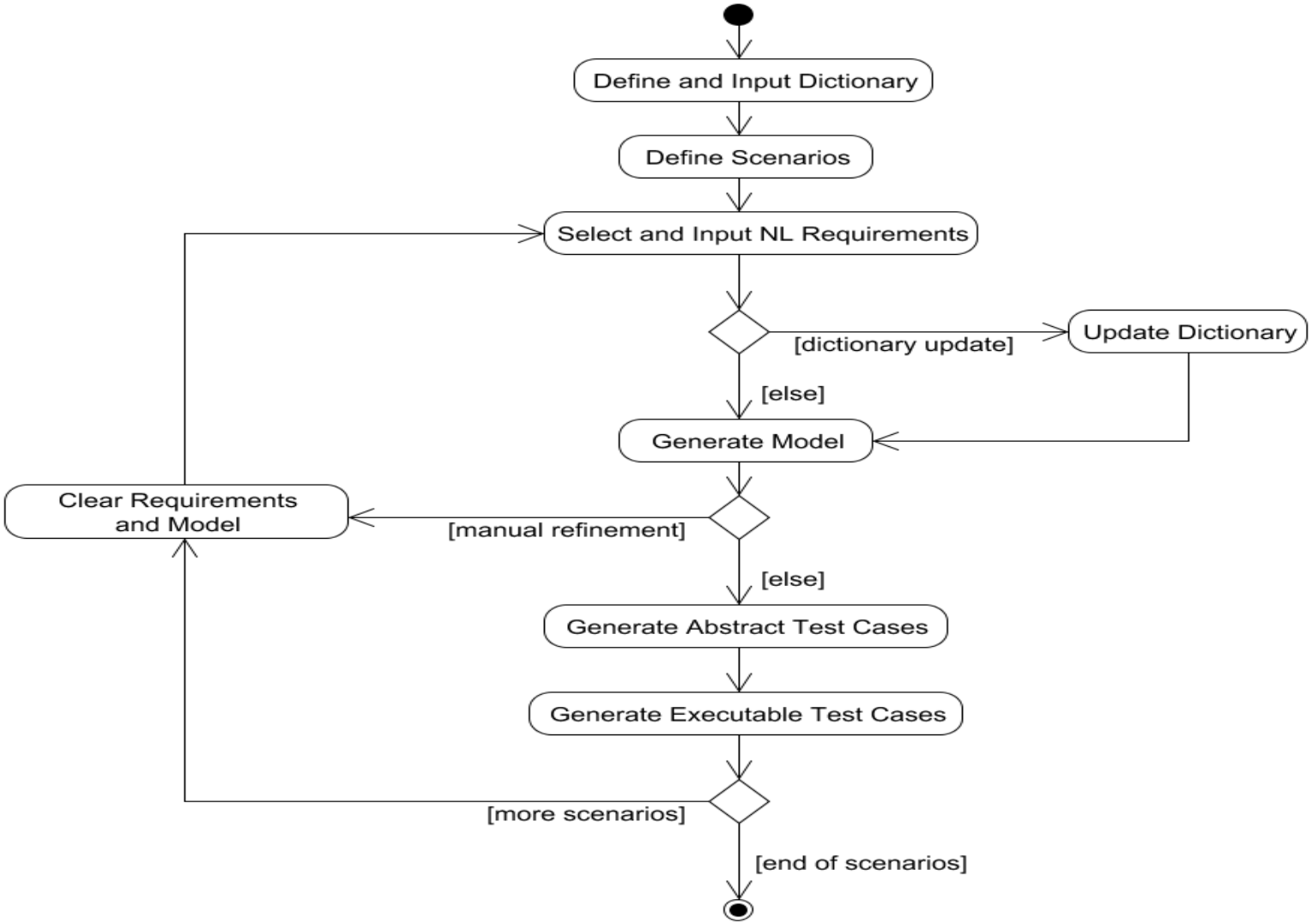
Step	Event	State	Output
1	EB9	CountingTimeWaitingExpid	
2	WaitingTimeExpired	IdleWaitingSync	
-	-	-	-
3	EB9	CountingTimeWaitingExpid	
4	ExpidRec	CountingTimeWaitingType	
5	WaitingTimeExpired	IdleWaitingSync	
-	-	-	-
6	EB9	CountingTimeWaitingExpid	
7	ExpidRec	CountingTimeWaitingType	
8	TypeRec	CountingTimeWaitingSize	
9	WaitingTimeExpired	IdleWaitingSync	
-	-	-	-
10	EB9	CountingTimeWaitingExpid	
11	ExpidRec	CountingTimeWaitingType	
12	TypeRec	CountingTimeWaitingSize	
13	SizeRec	CountingTimeWaitingData	
14	WaitingTimeExpired	IdleWaitingSync	
-	-	-	-
15	EB9	CountingTimeWaitingExpid	
16	ExpidRec	CountingTimeWaitingType	
17	TypeRec	CountingTimeWaitingSize	
18	SizeRec	CountingTimeWaitingData	
19	DataRec	CountingTimeWaitingChecksum	
20	WaitingTimeExpired+ChecksumRec	IdleWaitingSync	
-	-	-	-
21	NotEB9	IdleWaitingSync	
-	-	-	-

Intranet local | Modo Protegido: Desativado

V&V Activities at INPE: Products

- SOLIMVA ⇒ A methodology aiming at:
  - the generation of model-based system and acceptance test cases considering Natural Language (NL) requirements deliverables (artifacts) ⇒ Version 1.0 (software testing);
  - the detection of incompleteness in software specifications ⇒ Version 2.0 (software inspection with the aid of formal verification);
  - Formal Verification (Model Ckecking) of UML-based software ⇒ Version 3.0 (Formal Verification in the traditional approach).

The SOLIMVA methodology 1.0: Workflow



The SOLIMVA methodology 1.0: Workflow

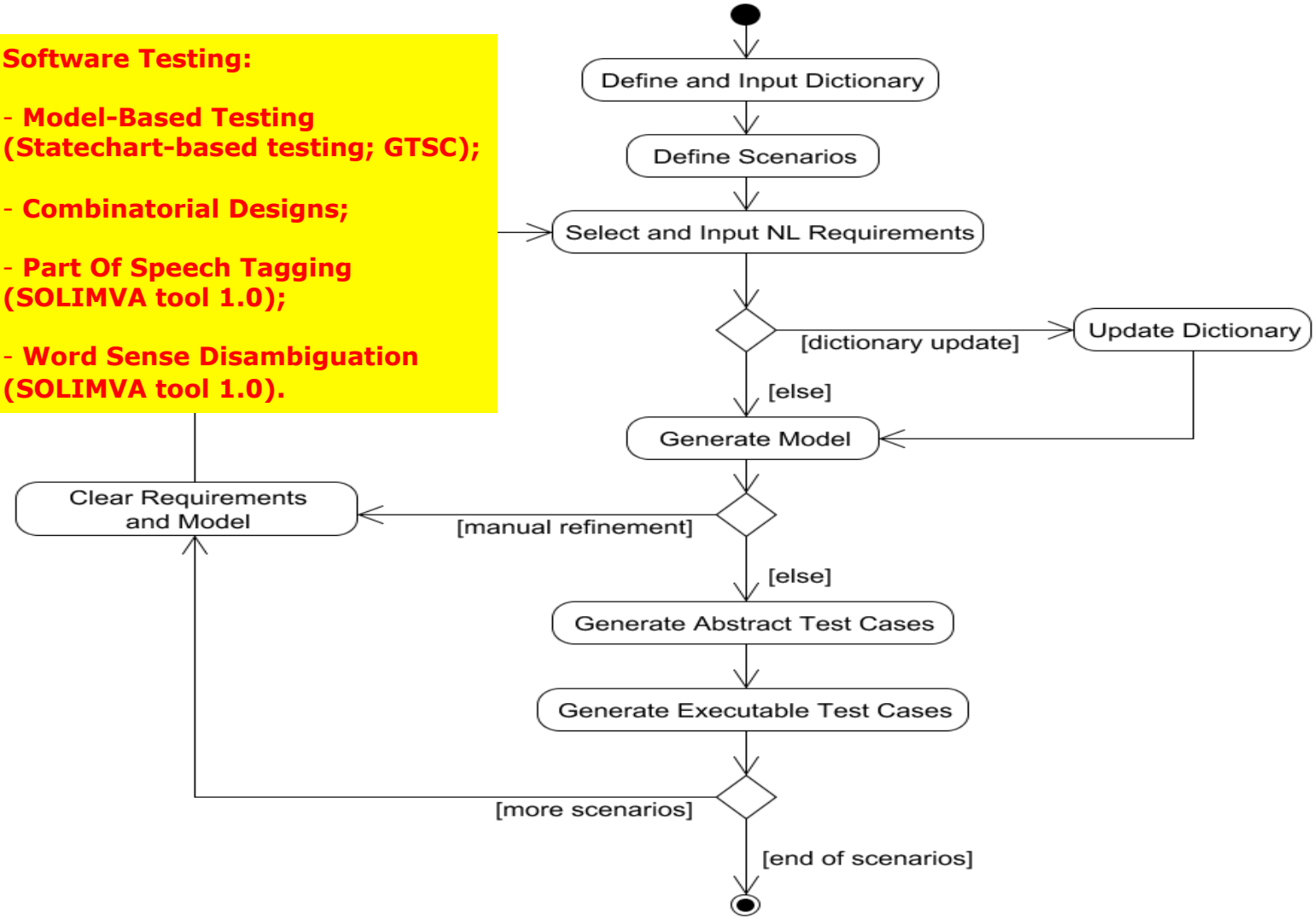
Software Testing:

- **Model-Based Testing (Statechart-based testing; GTSC);**

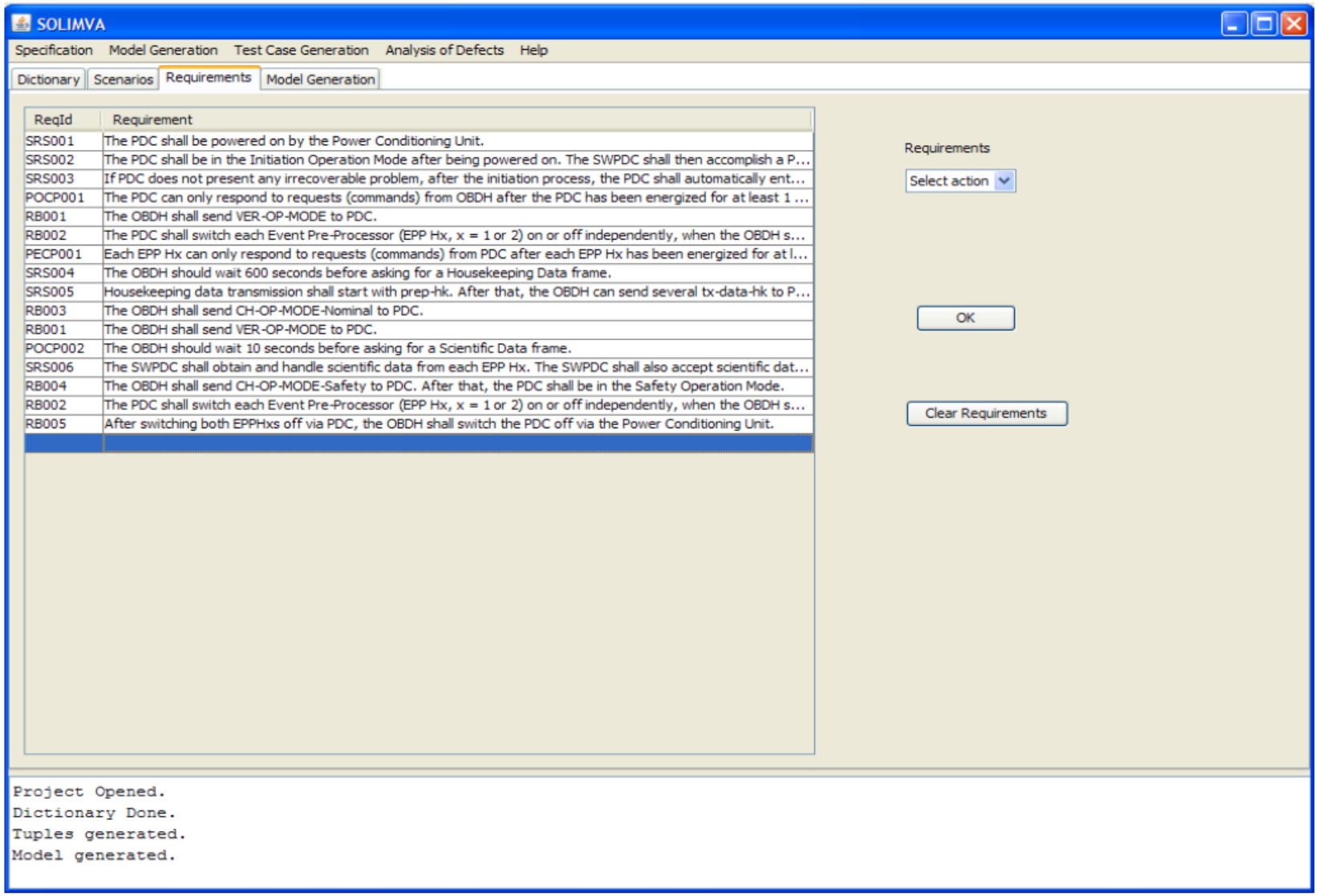
- **Combinatorial Designs;**

- **Part Of Speech Tagging (SOLIMVA tool 1.0);**

- **Word Sense Disambiguation (SOLIMVA tool 1.0).**



The SOLIMVA methodology 1.0: Tool (1.0)



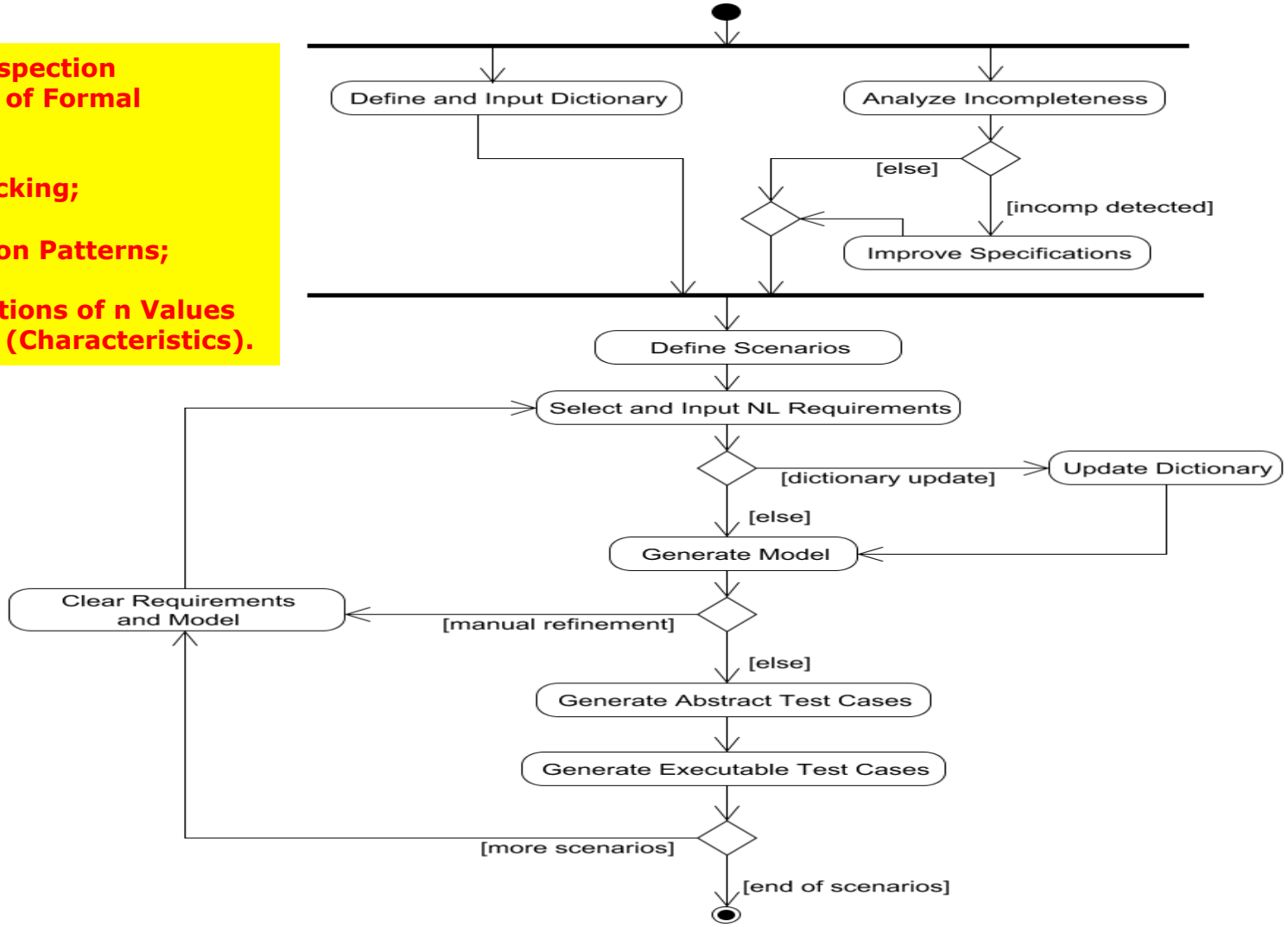
The SOLIMVA methodology 2.0: Workflow

Software Inspection with the aid of Formal Verification:

- **Model Checking;**

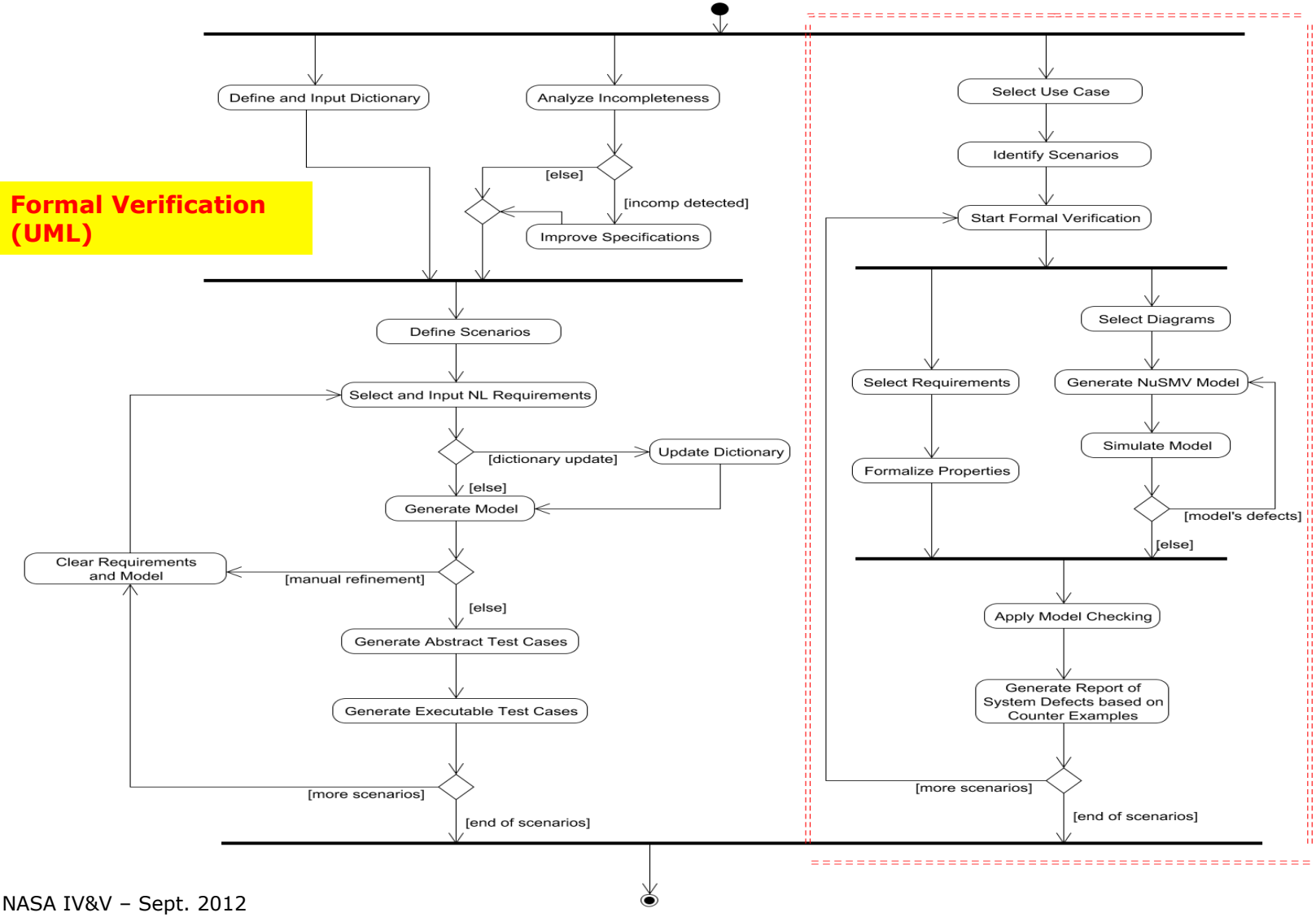
- **Specification Patterns;**

- **k-Permutations of n Values of Variables (Characteristics).**



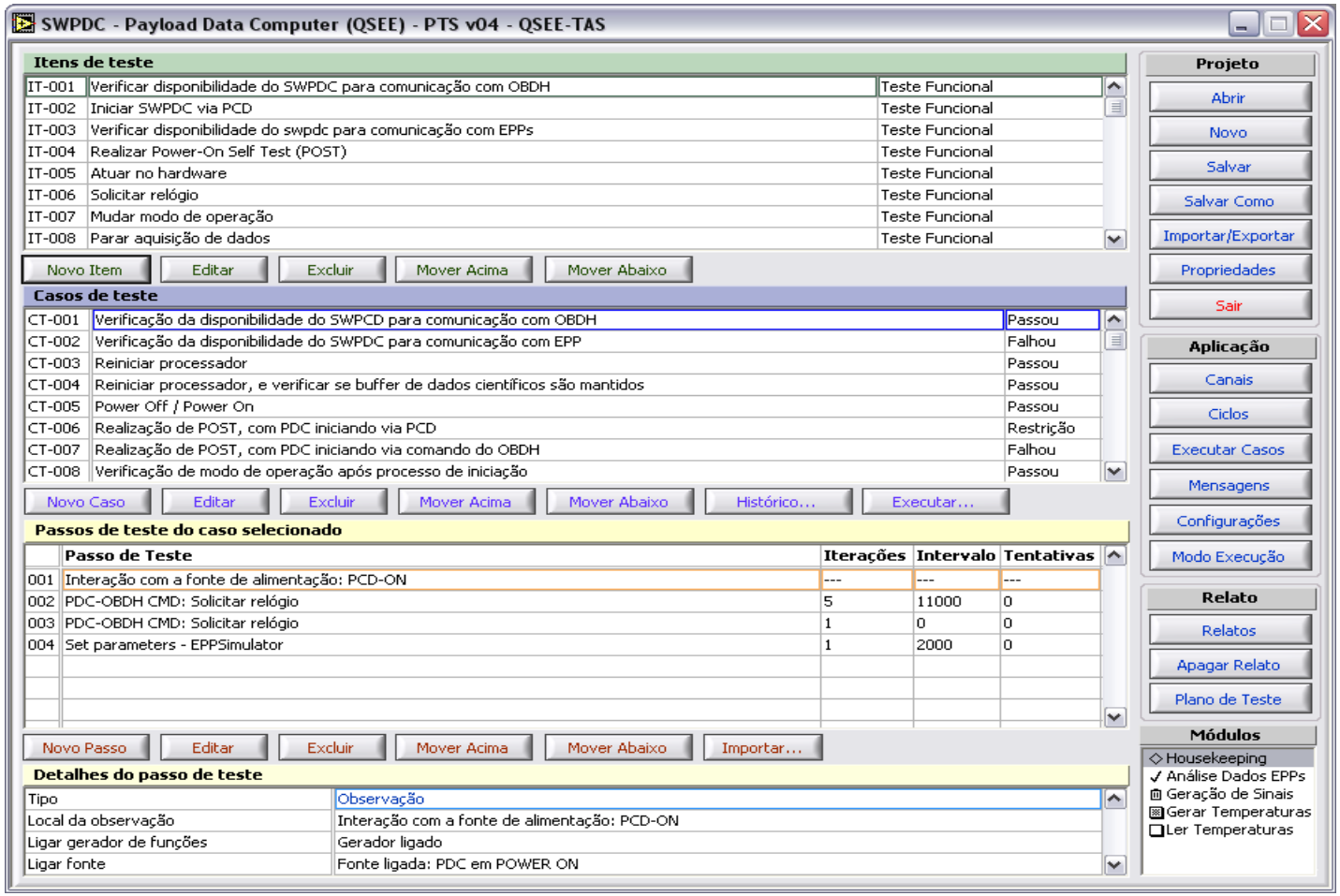
The SOLIMVA methodology 3.0: Workflow

Formal Verification (UML)



V&V Activities at INPE: Products

- Quality of Space Application Embedded Software – Automated Software Testing (QSEE-TAS): Automated test case execution, Automated test process documentation generation.



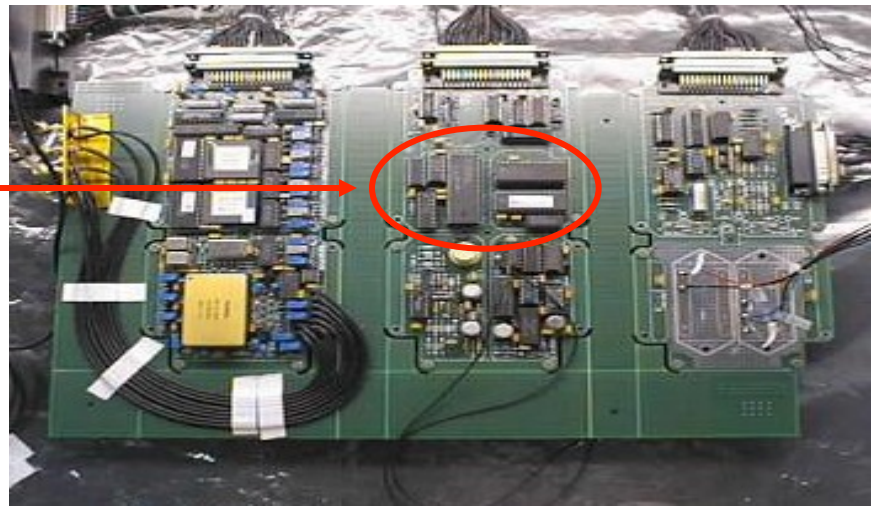
V&V Activities at INPE: Application to Projects

- Alpha, Proton and Electron Monitoring Experiment in the Magnetosphere (APEX).
  - Products ⇒ GTSC, WEB-PerformCharts, QSEE-TAS.
- Quality of Space Application Embedded Software (QSEE) – Software for the Payload Data Handling Computer (SWPDC).
  - Products ⇒ GTSC, WEB-PerformCharts, SOLIMVA, QSEE-TAS.
- protoMIRAX Scientific Experiment (Balloon application).
  - Products ⇒ GTSC, SOLIMVA.

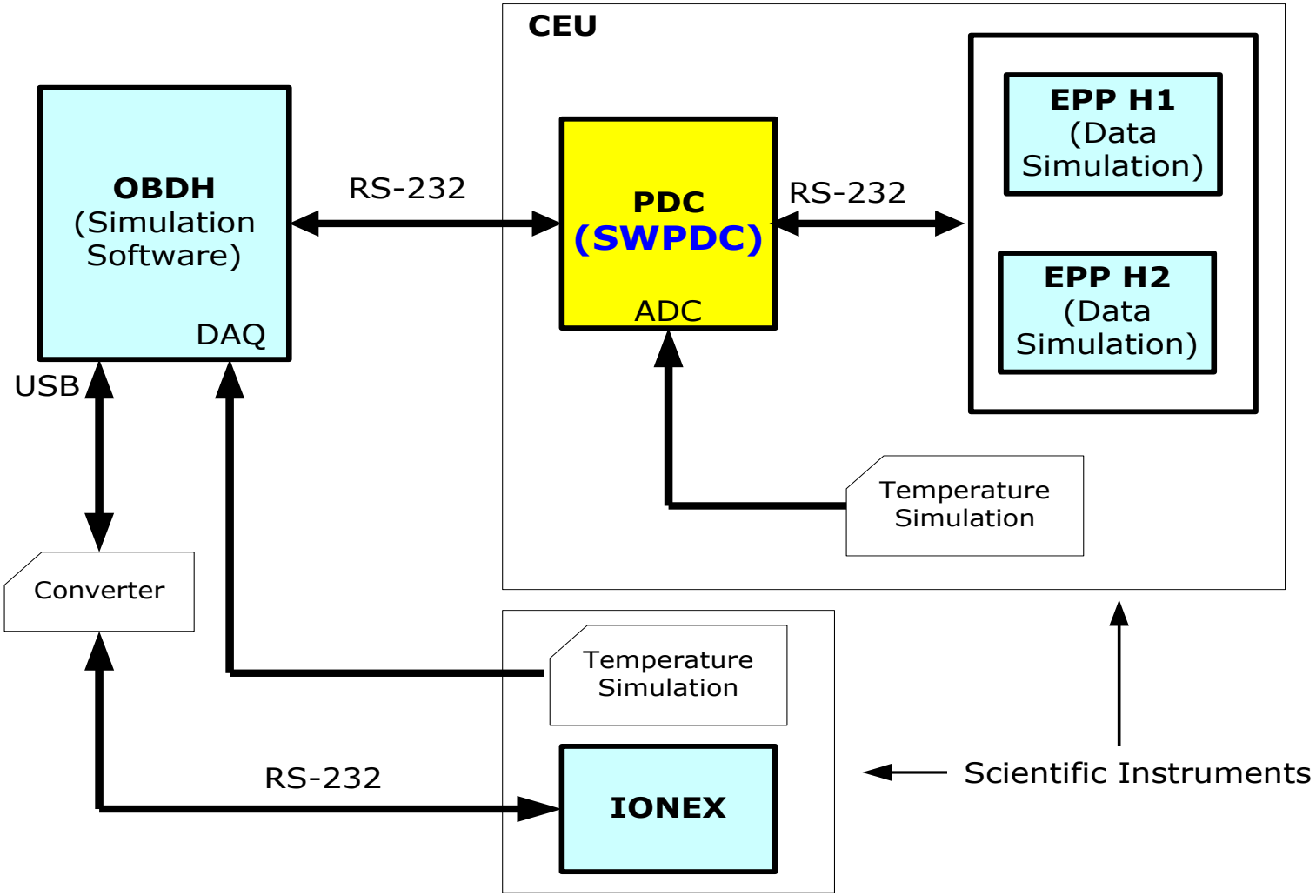
APEX

IUT:
-Command Recognition Component of the APEX embedded software;

- Simulated version (Java).

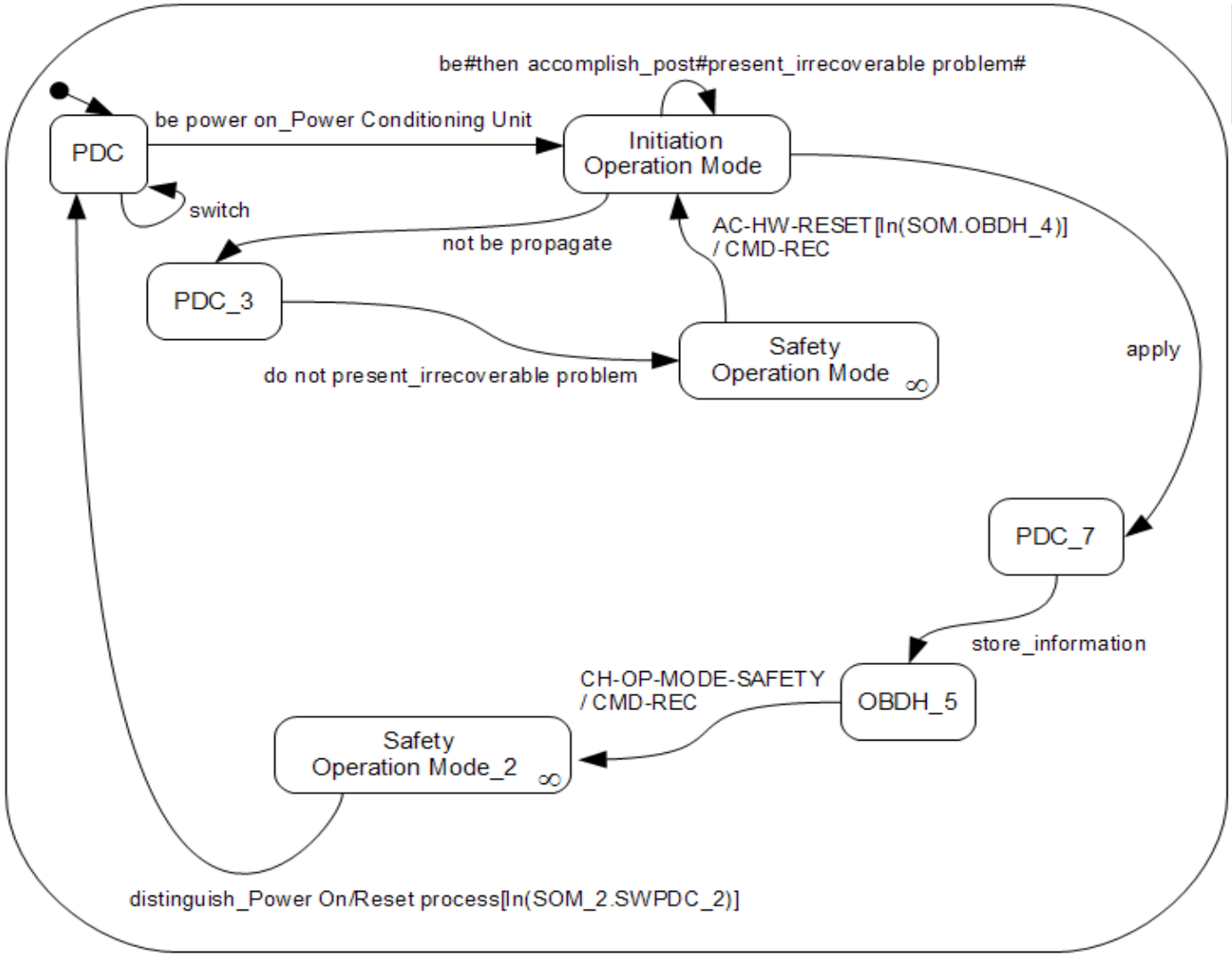


QSEE/SWPDC: Physical Architecture



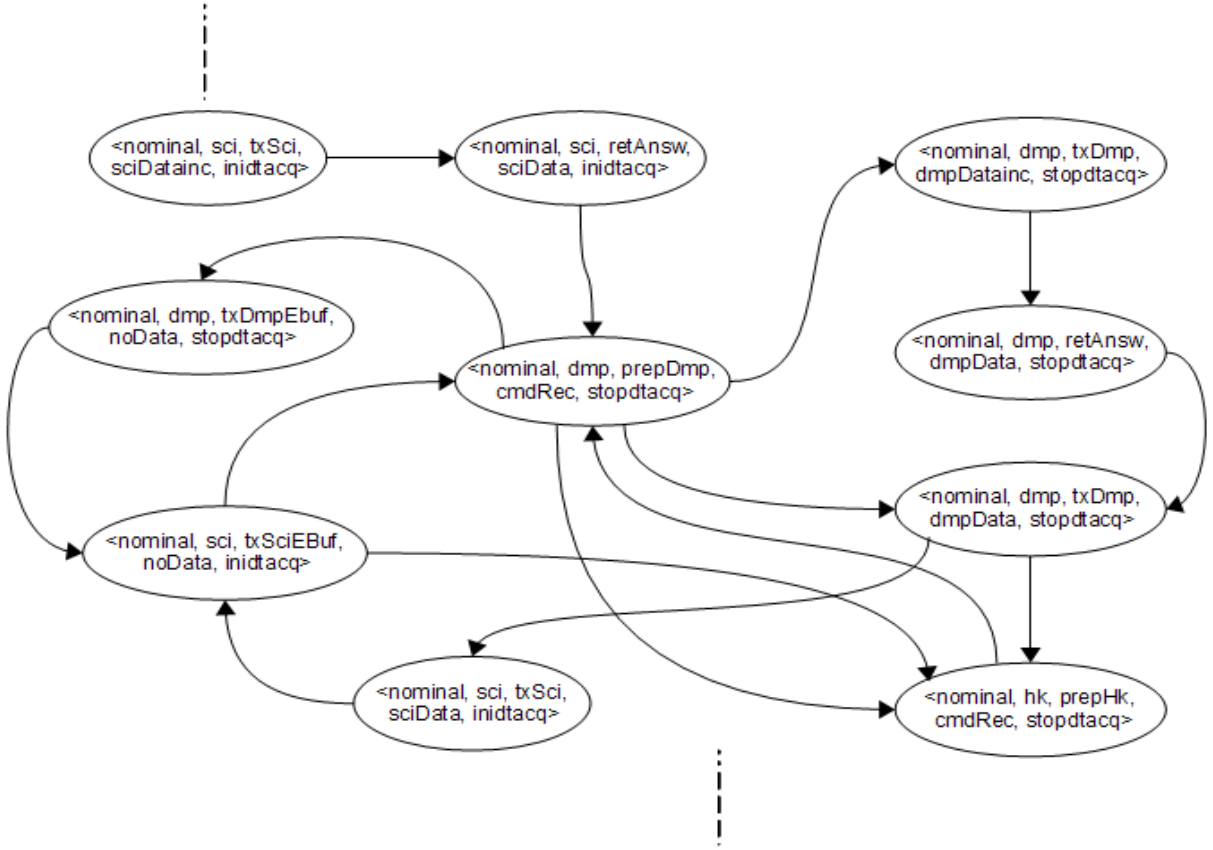
QSEE/SWPDC: Example of Statechart model

SOLIMVA 1.0 (methodology/ tool)



QSEE/SWPDC: Example of CTL properties and NuSMV model (SOLIMVA 2.0)

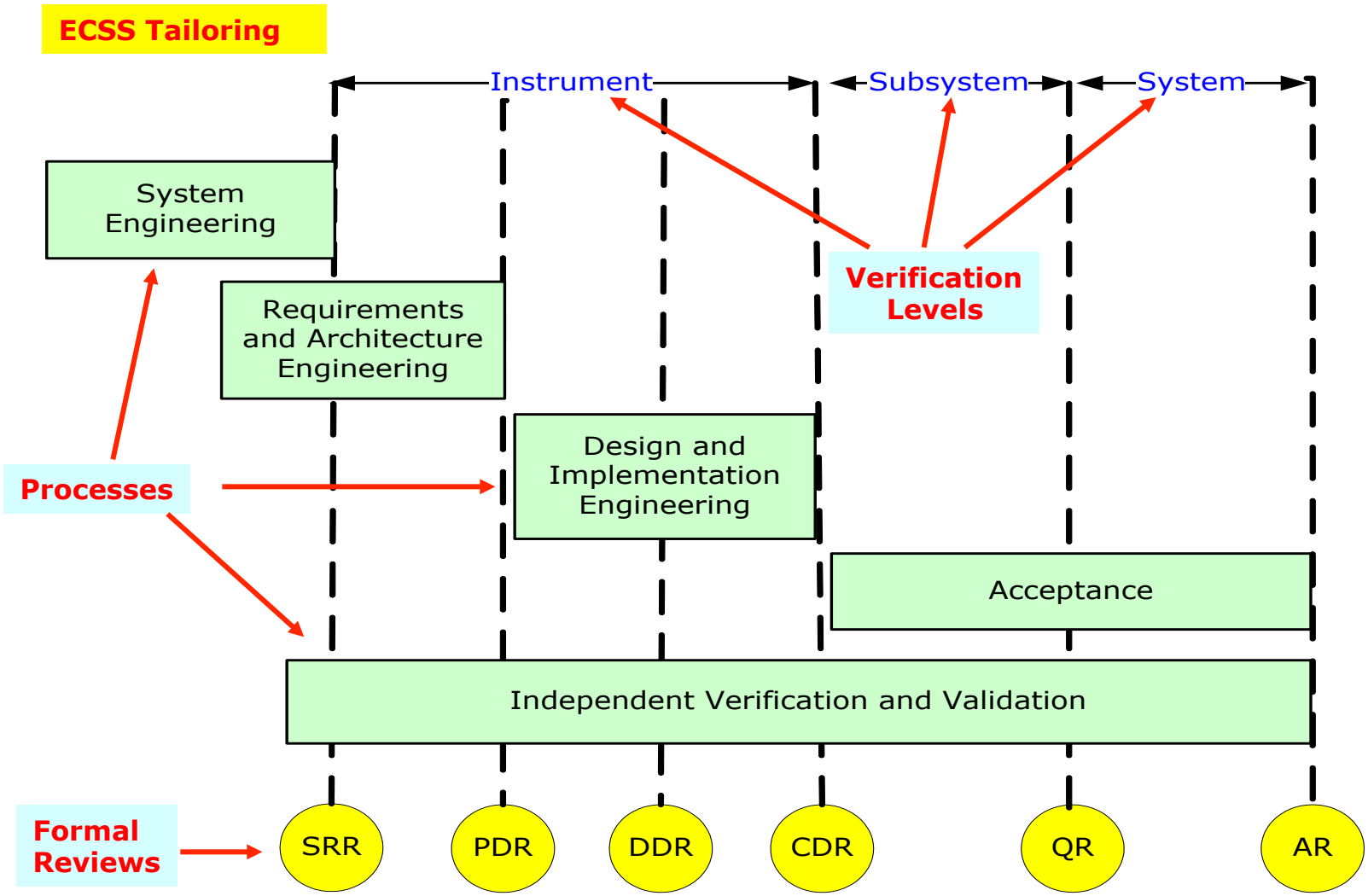
$$\neg\exists[\neg(prim = valprim_i \wedge sec_j = valsec_{t1}) \cup ((prim = valprim_i \wedge sec_j = valsec_{t2}) \wedge \neg(prim = valprim_i \wedge sec_j = valsec_{t1}))]$$



QSEE/SWPDC: Remarks

- GTSC ⇒ test suites with more than 300 test cases.
- SOLIMVA 1.0 ⇒ better strategy with test objectives clearly separated according to the directives of Combinatorial Designs.
- SOLIMVA 1.0 ⇒ Executable Test Cases predicted behaviors that did not exist (Expert's strategy).
- SOLIMVA 2.0 ⇒ 362 CTL properties formalized, 21 incompleteness defects detected.

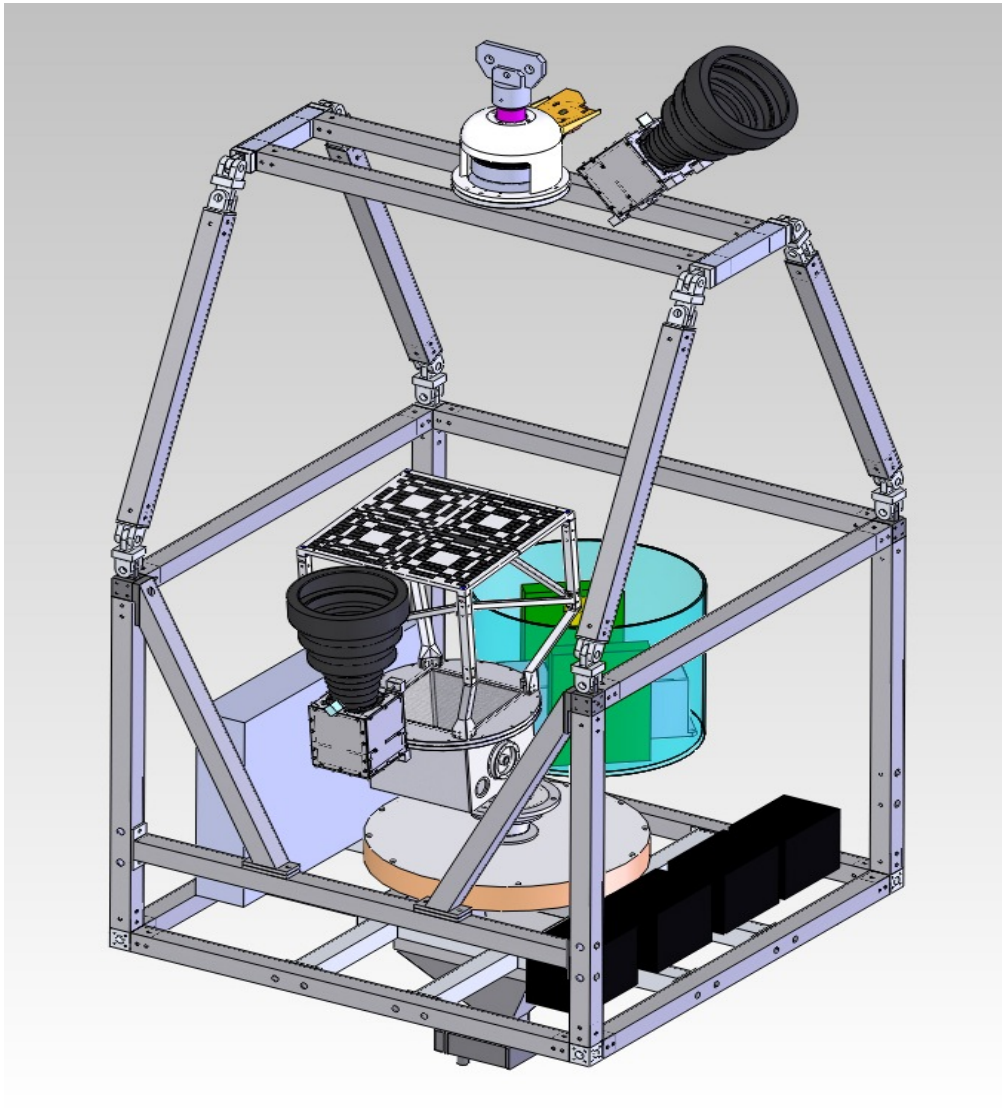
QSEE/SWPDC: Software Development Lifecycle



QSEE/SWPDC: IV&V

- Test Case Generation ⇒ Model-Based Testing (FSM).
- Test Case Execution ⇒ QSEE-TAS tool.
- Test Results Evaluation ⇒ Four-step process:
  - Observation of test results (QSEE-TAS interface);
  - Assignment of a preliminary verdict;
  - Meeting (every week) ⇒ IV&V team and customer representatives at INPE to evaluate the test reports;
  - Final verdict ⇒ Non-Conformance Record (NCR).

protoMIRAX Scientific Experiment



Conclusions

- Main V&V activities, products and projects in the area of formal V&V of safety-critical space software systems within IAE-LES and INPE (CEA/LAC).

- More confidence in the right choice of techniques to be used in each phase of development and in each part or component of the space software.

- Importance of computer-aided tools to support the formal V&V process.

- Efforts to bridge the gap between the state of the art and the state of the practice (application of research results to space projects development).

THANK YOU!

Miriam C. Bergue Alves:
miriammcba@iae.cta.br

Valdivino Alexandre de Santiago Júnior:
http://www.cea.inpe.br/~valdivino/

Nandamudi L. Vijaykumar:
http://www.lac.inpe.br/~vijay/Welcome.html