Correctness, Modelling and Performability
of Aerospace Systems

Overview of COMPASS Project

Viet Yen Nguyen

Software Modelling and Verification Group
RWTH Aachen University, Germany

Charles University in Prague, 19 May 2009
Outline

- Who’s Involved
- Scope
- Objectives and Approach
- Modelling Language
- Analyses
- COMPASS Toolset
- Planning
Research Project by European Industry and Academics

Consortium
- RWTH Aachen University
  Software Modeling and Verification Group
- Fondazione Bruno Kessler
  Embedded Systems Group
- Thales Alenia Space
  World-wide #1 in satellite systems

Financial support
- European Space Agency
## Design and Analysis at Systems Engineering Level

<table>
<thead>
<tr>
<th>Design Level</th>
<th>Examples</th>
<th>Formal Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>(component)</td>
<td>communication protocol, heat shield, cruise control</td>
<td>YES!</td>
</tr>
<tr>
<td>(system)</td>
<td>satellite, thrusters, rocket, space station</td>
<td>NO (not coherently)</td>
</tr>
</tbody>
</table>
# Integrated and Coherent Approach for Building Systems

## Modelling Language
- AADL
- Error Annex
- Hardware/Software Error Propagation
- Recovery Mechanisms
- Timing, Probability, Hybrid
- Formal Semantics

## COMPASS Toolset
- NuSMV
- FSAP
- RAT
- Sigref
- MRMC

## Analyses
- Model Checking
- SAT-Solving
- Performance Evaluation
- FTA
- FMEA

## Case Studies
- Satellite Thermal Regulation Manager
- Satellite FDIR
- Fault Detection, Identification and Recovery
- European Train Control System Level 3
## Integrated and Coherent Approach for Building Systems

### Modelling Language
- AADL + Error Annex
- Hardware/Software
- Error Propagation
- Recovery Mechanisms
- Timing, Probability, Hybrid
- Formal Semantics

### COMPASS Toolset
- NuSMV
- FSAP
- RAT
- Sigref
- MRMC

### Analyses
- Model Checking
- SAT-Solving
- Performance Evaluation
- FTA\(^a\)
- FMEA\(^b\)

### Case Studies
- Satellite Thermal Regulation Manager
- Satellite FDIR\(^a\) Effectivity
- European Train Control System Level 3

\(^a\)Fault Detection, Identification and Recovery
\(^b\)Failure Modes and Effects
System-Level Integrated Modelling Language

Inspired by AADL:
  - Component-oriented
System-Level Integrated Modelling Language

Inspired by AADL:
- Component-oriented
- Sub-/supercomponents
System-Level Integrated Modelling Language

Inspired by AADL:
- Component-oriented
- Sub-/supercomponents
- Event/data ports
System-Level Integrated Modelling Language

Inspired by AADL:

- Component-oriented
- Sub-/supercomponents
- Event/data ports
- (Functional) nominal behavior

2009, Viet Yen Nguyen
Inspired by AADL:

- Component-oriented
- Sub-/supercomponents
- Event/data ports
- (Functional) nominal behavior
- (Probabilistic) error behavior (Error Model Annex)
Inspired by AADL:
- Component-oriented
- Sub-/supercomponents
- Event/data ports
- (Functional) nominal behavior
- (Probabilistic) error behavior (Error Model Annex)
- Hybrid behavior (new)
SLIM Example: Battery

Component type and implementation

```slim
device type Battery

end Battery;

device implementation Battery.Imp

end Battery.Imp;
```
SLIM Example: Battery

Type defines the interface

device type Battery
  features
    empty: out event port;
    voltage: out data port real initially 6.0;
end Battery;

device implementation Battery.Imp

end Battery.Imp;
SLIM Example: Battery

Adding modes behavior

device type Battery
  features
    empty: out event port;
    voltage: out data port real initially 6.0;
end Battery;

device implementation Battery.Imp

  modes
    charged: activation mode

    depleted: mode

  transitions
    charged -[]-> charged;
    charged -[empty]-> depleted;
    depleted -[]-> depleted;
end Battery.Imp;
SLIM Example: Battery

Adding hybrid behavior

device type Battery
  features
    empty: out event port;
    voltage: out data port real initially 6.0;
end Battery;

device implementation Battery.Imp
  subcomponents
    energy: data continuous initially 100.0;
  modes
    charged: activation mode
      while energy'=-0.02 and energy>=20.0;
    depleted: mode
      while energy'=-0.03;
  transitions
    charged -[then voltage:=energy/50.0+4.0]-> charged;
    charged -[empty when energy<=20.0]-> depleted;
    depleted -[then voltage:=energy/50.0+4.0]-> depleted;
end Battery.Imp;
SLIM Example: Redundant Power System

Power system with battery subcomponents

```slim
system Power
  features
    voltage: out data port real;
end Power;

system implementation Power.Imp
  subcomponents
    batt1: device Battery.Imp
    batt2: device Battery.Imp
end Power.Imp;
```
SLIM Example: Redundant Power System

Adding dynamic reconfiguration

```
system Power
  features
    voltage: out data port real;
end Power;

system implementation Power.Imp
  subcomponents
    batt1: device Battery.Imp in modes (primary);
    batt2: device Battery.Imp in modes (backup);

modes
  primary: initial mode;
  backup: mode;

transitions
  primary -[batt1.empty]-> backup;
  backup -[batt2.empty]-> primary;
end Power.Imp;
```
Adding port connections

```
system Power
  features
    voltage: out data port real;
end Power;

system implementation Power.Imp
  subcomponents
    batt1: device Battery.Imp in modes (primary);
    batt2: device Battery.Imp in modes (backup);
  connections
    data port batt1.voltage -> voltage in modes (primary);
    data port batt2.voltage -> voltage in modes (backup);
  modes
    primary: initial mode;
    backup: mode;
  transitions
    primary -[batt1.empty]--> backup;
    backup -[batt2.empty]--> primary;
end Power.Imp;
```
Specifying Faulty Behavior

```plaintext
error model BatteryFailure
  features
    ok: initial state;
    dead: error state;
    batteryDied: out error propagation;
end BatteryFailure;

error model implementation BatteryFailure.Imp
  events
    fault: error event occurrence poisson 0.01;
  transitions
    ok -> [fault] dead;
    dead -> [batteryDied] dead;
end BatteryFailure.Imp;
```
Specifying Faulty Behavior

```plaintext
error model BatteryFailure
  features
    ok: initial state;
    dead: error state;
    batteryDied: out error propagation;
end BatteryFailure;

error model implementation BatteryFailure.Imp
  events
    fault: error event occurrence poisson 0.01;
  transitions
    ok -[fault]-> dead;
    dead -[batteryDied]-> dead;
end BatteryFailure.Imp;

Fault Injection
In error state dead, voltage:=0
```
Patterns

<table>
<thead>
<tr>
<th>Formalism</th>
<th>Intended Use</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTL, LTL</td>
<td>finite-state machines</td>
<td>Dwyer et al. 1999</td>
</tr>
<tr>
<td>MTL, TCTL</td>
<td>real-time properties</td>
<td>Konrad, Cheng 2005</td>
</tr>
<tr>
<td>PCTL, CSL</td>
<td>probabilistic properties</td>
<td>Grunske 2008</td>
</tr>
</tbody>
</table>

Example

- Pattern: “the system shall have a behaviour where with probability at least \( p \) it is the case that \( \Phi \) holds contiinusly within time bound \([t, t']\)”
- User instantiation \( p = \frac{3}{5}, \ \Phi = voltage \geq 80, \ \text{and} \ t = 0, t' = 10 \)
- CSL formula: \( \mathcal{P}_{\geq \frac{3}{5}} (\square \leq 10 (voltage \geq 80)) \)
Quantitative Analyses

+ probabilistic analysis of dynamic fault trees
Architectural Overview

- **Correctness Verification**
  - Property verification
  - Simulation

- **Diagnosability Analysis**
  - FDIR effectiveness measures
  - Synthesis of Observability Requirements

- **Performability Analysis**
  - Performability measures
  - Probabilistic fault trees

- **Safety Analysis**
  - Dynamic Fault Trees
  - FMEA Tables

- **Requirements Validation**
  - Property Assurance
  - Property Simulation

Diagram:

- Slim2SMV Instantiator
- Slim Property Table
- Symbol Table
- Model Extension
- Slim Property Instantiator
- Slim2SMV
- SMV2Sigref
- Sigref2MRMC
- MRMC
- NuSMV
- Trace Viewer
- Fault Tree Viewer
- RAT

Copyright 2009, Viet Yen Nguyen
Prototype is Up & Running as of April 2009
European Cooperation for Space Standardisation Process

- SLIM Language Design.
- Software Specification.
- Formal Semantics.
- Prototype Tool Implementation.
- Prototype Evaluation.
- Final Implementation. Due: March 2010.
For More Information

Website
http://compass.informatik.rwth-aachen.de

Publications


Hard-copy prints of these publications are available on request.

Demo of Prototype
Come to me after the talk!