Model-based system engineering for safety analysis of complex systems
**Motivation**

- **Classical SA techniques: Fault trees, FMEA, etc…**
  - Performed mostly manually
  - Time consuming, costly, high probability of errors…
  - Little tool support
  - No strong links between system engineering and SA

- **Achieved results in formal approaches**
  - Industrial toolsets (Isograph, Item, Relex, etc) : well-elaborated but costly
  - Academic tools (FSAP-NUSMV, ARC/AltaRica, etc) : gap in graphical representation of SA information

- **Goals**
  - To provide a support for SA engineers by integrating SA techniques into model-driven engineering
  - To leverage features of UML/SysML to develop a toolset for model checking and fault tree generation
CONTENTS

- Motivation
- Model-Based Systems Engineering & Safety Analysis
- Safety Analysis Toolset
- Example
- Conclusion & Further work
SA integrated development flow
Parallel V-cycle

SA Methods

Safety Assessment

Hazard Analysis

Preliminary System Safety Assessment

SOPHIA PHA [1]

Evaluate architecture & derive safety requirements

- Classical SA
- Updated HA qualitative FTA
- Simulation stochastic behavior event/fault injection
- Formal methods model checking theorem proving
- Structural redundancy

Evaluate architecture according to safety requirements

- Qualitative & quantitative FTA, FMEA/FMECA/FAE, simulation, formal methods, mitigation techniques

System Development

System Concept & Requirements

System Design & Optimization

Validation

System Acceptance & Maintenance

Module Integration & Test

System Integration & Test

Validation

Implementation

SW/HW Module Design & Optimization

Preliminary Module Safety Assessment

System Safety Assessment

Define & classify failure conditions

- Functional HA
- Preliminary HA
- Failure analysis
- Risk analysis

Converters

SysML2NuSMV
SysML2AltaRica [2]


MBSAW'2012, 11 – 12 Septembre 2012, Bordeaux, France
Engineers are finding that they can save significant time by modeling their system before building a physical prototype.
Engineers are finding that they can save significant time by modeling their system before building a physical prototype.

Why SysML?

- General-purpose modeling language
- Semantics are flexible and expressive
- Global overview of architecture
- Combines HW & SW
- Integrated requirements and life cycle traceability support
- SCADE integrates system view with SysML

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Safety Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open source</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Partners, MAENAD project</td>
<td>Partners, BUILD-IT-Safe project</td>
<td>30 days</td>
<td>30 days</td>
</tr>
<tr>
<td>Hazard Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Failure analysis</td>
<td></td>
<td>✓</td>
<td>Risk analysis</td>
</tr>
<tr>
<td>FT generation</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>FTA &amp; optimization</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Event ordering analysis</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Model checking</td>
<td>BDD/SAT, Plain, CTL, LTL, PSL</td>
<td>BDD/SAT, Plain, μ-calculus, CTL*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FMEA/FMECA/FMEDA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FMEA</td>
<td>FMEA</td>
<td>FMEA FMECA</td>
<td>✓</td>
</tr>
<tr>
<td>Common Cause Analysis</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Establishing strong links and collaboration between different communities

**Benefits**

- Integrate safety analysis techniques into system modeling environment
- Make available different tools for representing both qualitative and quantitative analysis
CONTENTS

- Motivation
- Model Based Systems Engineering & Safety Analysis
- Safety Analysis Toolset
- Example
- Conclusion & Further work
System Modelling Environment for SA

- AltaRica
- FSAP/NuSMV
- HIP-HOPS
- Safety Architect

SysML Modeling Support
Profiles

- Different views of safety perspectives in a system

- Different language support

MBSAW'2012, 11 – 12 Septembre 2012, Bordeaux, France
Model Annotation with Failure Behaviour

Methods

- Profiles (EAST-ADL, etc)
- Analytical way (Safety Architect, HIP-HOPS)
- System modeling diagrams (State Machine SysML diagrams, comments, etc)

System Model + Failure annotations = Global view of failure:

Profiles (EAST-ADL, etc)
Analytical way (Safety Architect, HIP-HOPS)
System modeling diagrams (State Machine SysML diagrams, comments, etc)
Model Annotation with Failure Behaviour

Methods

- Profiles (EAST-ADL, etc)
- Analytical way (Safety Architect, HIP-HOPS)
- System modeling diagrams (State Machine SysML diagrams, comments, etc)

System Model + Failure Annotations of Components = Global View of Failures:
* System
* Component
Model Transformation Methodology

SysML Metamodel

Conforms to

SysML Model

Model to Model Transformation

Target Language Metamodel

Conforms to

Target Language Model

Model to Text Transformation

Target Language Grammar

Conforms to

Target Language Code

Conforms to
SysML2AltaRica converter plug-in

Interface

© CEA LIST 2012

MBSAW'2012, 11 – 12 Septembre 2012, Bordeaux, France
SysML2NuSMV converter plug-in

Interface

MBSAW'2012, 11 – 12 Septembre 2012, Bordeaux, France
CONTENTS

- Motivation
- Model Based Systems Engineering & Safety Analysis
- Safety Analysis Toolset
- Example
- Conclusion & Further work
No train

- G1 excites Relay core, which in turn attracts the Contact, so that Signal Circuit for Green light is closed (Green light is on, Red light is off)

Train arrives

- Track Circuit is short-cut through the Train Axle. Therefore, the Relay is not excited, and Red light is on while Green light is off

Example Train Detection System

Train Detection System
Block Diagram

- Block annotation
- Block output deviations
- Block interconnections
- Propagation of faults through the system
- Each block has an associated State Machine Diagram
- Annotation of the failure modes for each block

```plaintext
StateMachine_Contacts

- closed
  - open
  - close

- open
  - open_fault
  - close
  - close_fault

- fail_to_open
  - open

- fail_to_close
  - close
```

Failure states
Train Detection System

NuSMV Input Model

---

```
MODULE Contacts(flowport1, flowport2)

VAR
  outputs
  flowport3 : boolean;
  flowport4 : boolean;

TRANSITIONS
  open_fault_0 : boolean;
  open_1 : boolean;
  close_fault_2 : boolean;
  close_3 : boolean;

STATES
  state : {closed, fail_to_open, fail_to_close, open};

ASSIGN
  init(state) := closed;
  next(state) :=
  case
    state = closed & open_fault_0 = TRUE : fail_to_open;
    state = closed & open_1 = TRUE : open;
    state = open & close_fault_2 = TRUE : fail_to_close;
    state = open & close_3 = TRUE : closed;
  esac;

  flowport3 := flowport1 | flowport2 | fail_to_open | fail_to_close;
  flowport4 := flowport1 | flowport2 | fail_to_open | fail_to_close;
```
Train Detection System

NuSMV Results

**Declaration of Module Contacts**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDD States</td>
<td></td>
</tr>
<tr>
<td>Arc violation threshold</td>
<td>0</td>
</tr>
<tr>
<td>Cache hit threshold for recycling</td>
<td></td>
</tr>
<tr>
<td>Dead nodes counted in triggering reordering</td>
<td></td>
</tr>
<tr>
<td>Default BDD reordering method</td>
<td>4</td>
</tr>
<tr>
<td>Default ZDD reordering method</td>
<td>4</td>
</tr>
<tr>
<td>Dynamic reordering of BDDs enabled</td>
<td>no</td>
</tr>
<tr>
<td>Dynamic reordering of ZDDs enabled</td>
<td>no</td>
</tr>
<tr>
<td>GA population size</td>
<td>0</td>
</tr>
<tr>
<td>Garbage collection enabled</td>
<td>yes</td>
</tr>
<tr>
<td>Garbage collections so far</td>
<td>0</td>
</tr>
<tr>
<td>Group checking criterion</td>
<td>7</td>
</tr>
<tr>
<td>Hard limit for cache size</td>
<td>1398101</td>
</tr>
<tr>
<td>Limit for fast unique table growth</td>
<td>838850</td>
</tr>
<tr>
<td>Maximum growth while setting a variable</td>
<td>1.2</td>
</tr>
<tr>
<td>Maximum number of variables shifted per reordering</td>
<td>1000</td>
</tr>
<tr>
<td>Maximum number of variable swaps per reordering</td>
<td>2000000</td>
</tr>
<tr>
<td>Memory in use</td>
<td>4137740</td>
</tr>
<tr>
<td>Next reordering threshold</td>
<td>4809</td>
</tr>
<tr>
<td>Number of BDD and ADD nodes</td>
<td>4</td>
</tr>
<tr>
<td>Number of BDD variables</td>
<td>0</td>
</tr>
<tr>
<td>Number of buckets in unique table</td>
<td>256</td>
</tr>
<tr>
<td>Number of cache collisions</td>
<td>0</td>
</tr>
<tr>
<td>Number of cache deletions</td>
<td>0</td>
</tr>
<tr>
<td>Number of cache entries</td>
<td>282144</td>
</tr>
<tr>
<td>Number of cache hits</td>
<td>0</td>
</tr>
<tr>
<td>Number of cache insertions</td>
<td>0</td>
</tr>
<tr>
<td>Number of cache look-ups</td>
<td>0</td>
</tr>
<tr>
<td>Number of crossovers for GA</td>
<td>0</td>
</tr>
<tr>
<td>Number of dead BDD and ADD nodes</td>
<td>0</td>
</tr>
<tr>
<td>Number of dead ZDD nodes</td>
<td>0</td>
</tr>
<tr>
<td>Number of LIVE BDD and ADD nodes</td>
<td>4</td>
</tr>
<tr>
<td>Number of LIVE ZDD nodes</td>
<td>0</td>
</tr>
<tr>
<td>Number of ZDD nodes</td>
<td>0</td>
</tr>
<tr>
<td>Number of ZDD variables</td>
<td>0</td>
</tr>
<tr>
<td>Peak number of live nodes</td>
<td>4</td>
</tr>
<tr>
<td>Peak number of nodes</td>
<td>1022</td>
</tr>
<tr>
<td>Realignment of BDDs to ZDDs enabled</td>
<td>no</td>
</tr>
<tr>
<td>Realignment of ZDDs to BDDs enabled</td>
<td>no</td>
</tr>
<tr>
<td>Recombination threshold</td>
<td>0</td>
</tr>
<tr>
<td>Reorderings so far</td>
<td>0</td>
</tr>
<tr>
<td>Soft limit for cache size</td>
<td></td>
</tr>
<tr>
<td>Symmetry violation threshold</td>
<td>0</td>
</tr>
<tr>
<td>Time for garbage collection</td>
<td>0.00 sec</td>
</tr>
</tbody>
</table>
Train Detection System

NuSMV Results

```
********** Simulation Starting From State 3.1 **********

NuSMV > show traces -v
<|-- 3.1 <--
Trace Description: Simulation Trace
Trace Type: Simulation

control.out_control = TRUE
control.fault_0 = TRUE
control.state = operating
ps.out_ps = TRUE
ps.fault_0 = FALSE
ps.state = operating
pedestal.out_pedestal = TRUE
pedestal.fault_0 = TRUE
pedestal.state = operating
trans1.out_trans = TRUE
trans1.fault_0 = FALSE
trans1.state = operating
recv.out_recv = TRUE
recv.fault_0 = FALSE
recv.state = operating
trans2.out_trans = TRUE
trans2.fault_0 = TRUE
trans2.state = operating

---

<table>
<thead>
<tr>
<th>State: 3.2</th>
<th>control.out_control = FALSE</th>
<th>control.fault_0 = FALSE</th>
<th>control.state = internal_failure</th>
<th>ps.out_ps = FALSE</th>
<th>ps.fault_0 = FALSE</th>
<th>ps.state = operating</th>
<th>pedestal.out_pedestal = FALSE</th>
<th>pedestal.fault_0 = FALSE</th>
<th>pedestal.state = internal_failure</th>
<th>trans1.out_trans = FALSE</th>
<th>trans1.fault_0 = FALSE</th>
<th>trans1.state = operating</th>
<th>recv.out_recv = FALSE</th>
<th>recv.fault_0 = FALSE</th>
<th>recv.state = operating</th>
<th>trans2.out_trans = FALSE</th>
<th>trans2.fault_0 = FALSE</th>
<th>trans2.state = internal_failure</th>
</tr>
</thead>
</table>

---

<table>
<thead>
<tr>
<th>State: 3.3</th>
<th>control.out_control = FALSE</th>
<th>control.fault_0 = FALSE</th>
<th>control.state = internal_failure</th>
</tr>
</thead>
</table>
```

---

Train Detection System

NuSMV Results
Fault tree in Open PSA format
CONTENTS

Motivation

Model Based Systems Engineering & Safety Analysis

Safety Analysis Toolset

Example

Conclusion & Further work
System Modelling Environment for Safety Analysis

- The use of UML/SysML and Papyrus platform for further model checking and fault tree generation
  - Graphical and accessible language
  - Modelling of system architecture, behaviour and failure logic

- The support of two transformation methods and associated SA flows based on ARC(AltaRica) and NuSMV

- Fault tree generation, optimization and analysis

- Graphical representation of failure data using State Machine Diagrams
Integrate different SA tools, plug-ins, profiles and libraries

FMEA generation and analysis

Develop SA profiles for SysML models
- Reflect the results of quantitative and qualitative safety analysis in SysML models
- Graphical representation of Fault Trees
- Distinguish failure modes in State Machines