Building Validation Suites with Eclipse for Model-based Generation Tools

24.6.2010
Dr. Oscar Slotosch
Validas AG
Content

‣ Model-based Development (of Embedded Systems)

‣ ISO 26262

‣ Validation Suite
  – Architecture
  – Used Eclipse Technologies for building
  – Experiences

‣ Conclusion
Validas A G

- Founded 2000
- 20 Employees
- Motto: Validated Quality
  - Model-based Development
  - Model-based Testing
  - Tool Qualifications
- We solve Your Challenges!
- Clients
  - BMW
  - EADS
  - ESG
  - Giesecke & Devrient
  - MAN
  - Eurocopter
Models in Embedded System Development Process

- Code-Generator
- Test-Generator

Supporting Tools

Model-based Development

Integration Test

Unit Test

Application / Domain Specific ("System") Models

"Unified" Models

Model-based Testing
Model-based Development

- Development
  - Model Creation => "Graphical Tool"
  - Code Generation => "Generator"
  - Code Compilation => Compiler

- Important Verification Methods for Models
  - Simulation (MIL, SIL, PIL)
  - Rule Checking ("Subset-Checker")
  - Property Verification (Formal Methods)
  - Model Coverage
  - Back-To-Back-Testing: MIL = SIL = PIL

Out1 = (In1 + In2) * In3;

... maddf a15,a4,d15,#0 ...

...
ISO 26262-8, Chapter 11: Qualification of software tools

- **Classification in “Tool Confidence Level (TCL)”**

- **Tool Impact (TI)**
  - TI0: no impact => Tool is TCL1
  - TI1: some impact

- **Tool Error Detection/prevention probability (TD)**
  - TD1: high confidence => Tool is TCL1
  - TD2: medium confidence => Tool is TCL2
  - TD3: low confidence => Tool is TCL3
  - TD4: other => Tool is TCL4
Example: Tool Chains in Model-based Development

- Customer: can verify tool results

- Tool provider
  - Has to require verification of his tools ("redundancy")
  - or
  - Validate his tool (Validation Suite / Proven in use)
Process of Tool Qualification

- Requirements / Standard
- Manual of Tool with
  - Used Functions
- Tool Developer Guide (Process)
- Validation and Verification Plan for
  - Requirements
  - Functions
  - Compliance with Process
- Validation and Verification Report

Quote from Ada-Test Suite (ACATS): “...the ACATS tests the normal usage of these features, not unusual corner-cases.”
Test Method for Generators
Construction of Validation Suites

Test Specification: Domain/Tool Specific Language

- Test Inputs: (Model-)Generator
- Test Outputs: Tool Interpreter (Reference Tool)
- Test Automation of
  - Code Generation, Compilation and
  - Execution (on Target)
- Analysis tools:
  - Report Generation
  - Root Cause Analysis
- Until:
  - All Model/Features are covered -> Model Coverage
  - All Tests have been Executed
  - All Deviations have been analyzed
Eclipse-Modeling for the Construction of Suites

**Models/Code**

- **Model**
- **Code**
- **Compiler**

**Validation Suite**
- **Models**
- **Test Data**

**Meta Model**

```
uint8 Out;
int16 In1, In2;
Out = In1 % In2;
```

**DSL**

```
Param Type;
Param Op;
Out uint8;
In1, In2 Type;
Out = Op(In1, In2);
```

**OCL**

```
Projection
Semantik
```

**xText / ATL**

```
Functions?
End?
Correctness?
Specification?
```
Coverage: Model-Projektion

- Input: Set of Reference
  - Models / Elements of Meta Models
  - XML-Structures
  - Projection Results (hierarchy)

- Output: List of used
  - Model Elements
  - Model Properties
  - Settings / Configurations

- Filter Mechanism for not relevant properties (color,..)

- Comparison of two Sets
  - Validation Suite
  - Application models
Example of Meta Model

Oriented on Model Elements / Functions
Used Eclipse Tools

- DSL (xText) for Specification
  - Parameter: Operators, Types, Constants
- ATL for Translation xText → M1
- M2M zur Generation of Models
- M2M Definition of Semantic
- OCL for Modeling Rule Checker
- Model-Projektion on EMF-Models for Coverage/Functions
DSL for Test Specification and Generation

Specification = Model Description + Parameter
Eclipse Experiences

- M1: approx. 150 Classes
- M2: approx. 350 Classes
- M2M: approx. 13000 Lines of ATL Transformations
- DSL: approx. 4000 Lines of Test Specification
- Results (Testsuite), generated in some hours:
  - 10000 Models with
  - 600000 Test sequences
- Stable & performant solution
- ...
- We would do it again using these Eclipse tools
Vision: Validation Framework for Different C Compilers

Compiler (incl. Source-Code)
- Target-Semantic
- Libraries
- Target-Spezifika

Opt. Restrictions
- Configuration
- Features

Compiler Validation Framework

Validation Suite for Compiler C and Target T

Bugs /
Restrictions

Qualification Report

Qualification Report

C Code

Static Checker

OK / NOK

Test Data

Dynamic Test-Checker

OK / NOK
Summary

- Models in Development of Embedded Systems
- ISO 26262 Tool Classification
- Validation Suites
- Eclipse is suitable for building Suites
Thank You!

Your partner for innovation in embedded quality

Arnulfstraße 27
80335 München
www.validas.de
info@validas.de