TransForm
Méthodes formelles pour le transport

systerel
Safe real-time solutions

The Application of Formal Methods to Railway Signalling Software

Laurent Voisin

A word about Systerel

- Techniques
- Tools
- Application
- Feedback

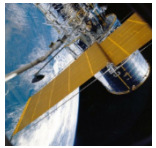
Systerel has been creating, designing and implementing innovative solutions for over 15 years in the area of real time and safety critical systems.

- Formal methods
- System design
- Software development
- RAMS

Creation: 2002
100 engineers
Turnover of 8 M€
+ 15 % R&D
70% of turnover for fixed price projects







systerel
Safe real-time solutions

Objective

Apply mathematically sound techniques

To disambiguate requirements

To make assumptions explicit

To prove that an artefact matches its specification

Similar to calculus for mechanical engineering

But proofs must be machine-checked

Techniques
Tools
Application
Feedback

Formal Techniques (a priori)

Techniques
Tools
Application
Feedback

Event-B

Modeling some aspects of a system by refinement steps and proving the consistency of steps with the RODIN platform

Software-B

Developing a piece of software correct by construction

Proof of specification consistency and proof of implementation/specification compliance with Atelier B

Translation from B code into C code with Systerel B/C Translator

Close to Ada SPARK

Formal Techniques (a posteriori)

Techniques
Tools
Application
Feedback

Formal Data Validation

Modeling configuration data and their properties in B

Automatic evaluation of properties with OVADO² certified double chain

Systerel Smart Solver (S3)

Modeling the specification of a system and its implementation and proving by model checking that they are consistent using the S3 certifiable solver

Proving that a system respects some properties with the S3 solver

Finding solutions to a constrained system with the S3 solver (e.g., test case generation)

systerel
Safe real-time solutions

High-end Tools (1)

Techniques
Tools
Application
Feedback

Rodin Platform

- Open Source Project (developed and maintained by Systerel)
- Event-B platform and provers
- Development techniques: Java, Eclipse plugins, compiler techniques, advanced GUI, Maven, 300 K lines

B to C Translator

- Systerel product
- Double translation chain from B code to C code T3 qualified EN50128
- Development techniques: compiler techniques

OVADO²

- RATP product developed and maintained by Systerel
- Double evaluation of predicates T2 qualified EN50128
- Development techniques:
  - Chain 1: OVADO using AST Rodin plugin and predicate evaluation engine
  - Chain 2: ProB model checker

High-end Tools (2)

Techniques
Tools
Application
Feedback

Systerel Smart Solver

Ada, C, SCADE Front-ends, Expansers, solver, equivalence builder, proof checker

Techniques: C, Ocaml, SAT & compiler techniques

Reusable Tools

Automatic documentation generation of a B-data model (in PDF, MS-Word, Latex)

Techniques: XSLT, XSL-FO, scripts

Simulation kernel with friendly user interfaces

Object-oriented modeling of the environment

Can interface with existing tools (e.g., ControlBuild)

Support for fault injection

Techniques: Python, HTML, SVG, JavaScript

Tools dedicated to projects

ZC CBTC Simulator (wayside of safe metro system)

Techniques: Eclipse, Java, JNI

Example of a Large Project

ZC CBTC

Turnkey project for the development of the main software of a Zone Controller subsystem of a CBTC metro

System design

Systerel required an Event-B study to prove that design choices respect some safety properties

Development of a simulator and a simplified Java software to verify availability

The system design document has been certified SIL4 EN50128

Software development

B-Software development metrics:

200 modules, 70 K lines of B, 35 K lines of C, 21 K Proof Obligations

Development of a qualified double chain B to C translator

B-Data validation of the whole CBTC system

B-Data validation with OVADO

Properties of the ZC B-model where exported as is in the OVADO model

Feedback: Formal Techniques Can Be a Success

Techniques
Tools
Application
Feedback

Successful Use of Formal Techniques

Formal Techniques can be applied successfully and can be efficient

Reaching a very high level of quality for safety critical systems

- Proof completeness
- Abstract model with proven properties that make sense for the target system
- Overall higher quality (as good as it gets)
- Modification/evolution perimeter completely mastered

The global cost (for critical systems) is not higher than with usual techniques

Feedback: But It May Be Difficult

Techniques
Tools
Application
Feedback

Difficulty of Constructing a Good Model

- Picking up the most suitable formal technique
- Defining a methodology to write the best model
- Achieving to model every properties that we would like to
- It requires training, experience and feedback
  - Not as easy as high-end tool development techniques (Internet search, download, documentation, tutorial)

Difficulty of Interactive Proof

- Interactive proof is difficult and costly (Event-B, software-B)
- Engineers always find tool performance too limited
  - Performance level may increase in the future
  - Tools start integrating several external provers
- A part of the model should be constructed at the same time proof is performed
- Techniques with full automated proof (S3, OVADO) are easier to use for engineers

systerel
Safe real-time solutions

Thank you