An industrial application of formal model based development: the Metrô Rio ATP case

A. Ferrari\textsuperscript{2,1} A. Fantechi\textsuperscript{1} M. Papini\textsuperscript{2} D. Grasso\textsuperscript{1}

\textsuperscript{1}University of Florence (DSI)

\textsuperscript{2}General Electric Transportation Systems (GETS)
Florence

2nd International Workshop on Software Engineering for Resilient System (SERENE2010)
Outline

1. Railway Signaling Context and ATP
2. Modelling Guidelines
3. Architecture Definition
4. Conclusion
Safety Critical Systems such as interlocking system and automatic train protection system.

General Electric Transportation System (GETS): 
- Effort to adopt formal methods for own development process
- Collaboration with the Faculty of Engineering of the University of Florence
- Experimentation of different technologies and tools for formal verification
Railway Signaling: ATP

- ATP systems
  - Focus on on-board equipment
  - Control modes logic
  - Message analysis algorithms
- CENELEC EN 50128 based development
  - European norm for railway
  - SIL-4 products (Higher level of integrity)
  - What about generated code?
- Development choices
  - Stateflow
  - Real Time Workshop Embedded Coder
  - MAAB Guidelines
Railway Signaling: ATP

Automatic Train Protection (ATP) systems
Challenges:
Development Choices

Challenges:

- **MAAB Guidelines**: Focused on the design of automotive systems, not for signaling systems
Development Choices

Challenges:

- **MAAB Guidelines**: Focused on the design of automotive systems, not for signaling systems
- **Stateflow**: Has not a rigorous formal semantics
Development Choices

Challenges:

- **MAAB Guidelines**: Focused on the design of automotive systems, not for signaling systems
- **Stateflow**: Has not a rigorous formal semantics
- **Real Time Workshop Embedded Coder**: It is not a EN 50128 certified coder
Challenges:

- **MAAB Guidelines**: Focused on the design of automotive systems, not for signaling systems
- **Stateflow**: Has not a rigorous formal semantics
- **Real Time Workshop Embedded Coder**: It is not a EN 50128 certified coder

**How did we address these shortcomings?**
## MAAB Guidelines

Adaptation of existing Guidelines through priority restrictions:

<table>
<thead>
<tr>
<th>Title</th>
<th>Priority</th>
<th>Restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transitions in Flowcharts</td>
<td>HR</td>
<td>M</td>
</tr>
<tr>
<td>Use of return value from graphical functions</td>
<td>R</td>
<td>SR</td>
</tr>
<tr>
<td>Bitwise Stateflow operators</td>
<td>SR</td>
<td>R</td>
</tr>
<tr>
<td>Use of unary minus on unsigned integers in Stateflow</td>
<td>R</td>
<td>M</td>
</tr>
<tr>
<td>Reuse of variables within a single Stateflow scope</td>
<td>R</td>
<td>M</td>
</tr>
<tr>
<td>Comparison operation in Stateflow</td>
<td>R</td>
<td>M</td>
</tr>
<tr>
<td>Scope of internal signals and local auxiliary variables</td>
<td>SR</td>
<td>M</td>
</tr>
<tr>
<td>.....</td>
<td>.....</td>
<td>.....</td>
</tr>
</tbody>
</table>

**Table: MAAB Guidelines adaptation**
Adaptation of existing Guidelines through priority restrictions:

<table>
<thead>
<tr>
<th>Title</th>
<th>Priority</th>
<th>Restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transitions in Flowcharts</td>
<td>HR</td>
<td>M</td>
</tr>
<tr>
<td>Use of return value from graphical functions</td>
<td>R</td>
<td>SR</td>
</tr>
<tr>
<td>Bitwise Stateflow operators</td>
<td>SR</td>
<td>R</td>
</tr>
<tr>
<td>Use of unary minus on unsigned integers in Stateflow</td>
<td>R</td>
<td>M</td>
</tr>
<tr>
<td>Reuse of variables within a single Stateflow scope</td>
<td>R</td>
<td>M</td>
</tr>
<tr>
<td>Comparison operation in Stateflow</td>
<td>R</td>
<td>M</td>
</tr>
<tr>
<td>Scope of internal signals and local auxiliary variables</td>
<td>SR</td>
<td>M</td>
</tr>
</tbody>
</table>

**Table:** MAAB Guidelines adaptation
Some other guidelines are needed for code generation issue:

- **Events shall not be used in Stateflow diagrams**: to avoid recursive call and then to save from infinite recursion and stack overflow
- **States and junctions shall not be used jointly**: to exclude backtracking without undo
- **Outgoing transitions shall have mutually exclusive conditions on their guards**: to avoid incorrect determinism on transitions evaluated in clockwise rule
Architecture Definition Approach

- Architecture definition must take into account of automatic code generation.
Architecture Definition Approach

- Architecture definition must take into account of automatic code generation.
- Multi-level architecture approach was defined to derive a formal model for the system.
Architecture Definition Approach

- Architecture definition must take into account of automatic code generation.
- Multi-level architecture approach was defined to derive a formal model for the system.
- The Goal is to create formal models that makes sense in terms of the architecture of the software system.
Architecture level

Operation Mode Manager

Information Point Manager

HMI Manager

Controls

Speed Limit Manager

Brake Manager

Speed Analyzers

Ferrari, Fantechi, Papini, Grasso

Model based development: Metrô Rio ATP case
Detail level

HMI Manager Design

HMI

Ferrari, Fantechi, Papini, Grasso

Model based development: Metrô Rio ATP case
Unit Requirements Decomposition

Req: When the rain function button (CH) is pressed and released within trainfunction milliseconds and the train is standing, the icon of the rain function shall be lighted on and the target speed when approaching the platform shall be reduced to 40 km/h if the train is positioned outdoor.

1. If the rain function button (CH) is pressed and released within trainfunction milliseconds the rain event shall be raised: HMI Manager
2. If the rain event is raised and the train is standing the rain function shall be activated: Platform Control
3. If the rain function is active and the train is positioned outdoor the target speed shall be reduced to 40km/h: Platform Speed Analyser
4. If the rain function is active the icon of the rain function button shall be lighted on: HMI Manager
Results

- Definition of 438 unit requirements that led to 13 statecharts
- 14 source generated files, one for each chart and one to manage the integration of the other units
- Approximately 120K lines of code

<table>
<thead>
<tr>
<th>Project</th>
<th>#Modules</th>
<th>LOC</th>
<th>#Bugs</th>
<th>Man/H</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSC Metro Rio</td>
<td>13</td>
<td>120K</td>
<td>33</td>
<td>16</td>
</tr>
<tr>
<td>SSC BL1Plus</td>
<td>12</td>
<td>40K</td>
<td>114</td>
<td>105</td>
</tr>
</tbody>
</table>

Table: Bug detection and correction costs for comparable project which required a modeling cost of approximately 4 man/months
Conclusion

- Notable reduction of bugs
- The novel design approach permitted to strongly reduce bugs detection time
- Design time increases approximately of 30%
- Verification time with model based testing and abstract interpretation for generated code reduced by 70%