



Invensys®

WESTINGHOUSE®
rail systems

Demonstrating Safety-Critical Properties of an Automatic Train Protection System

Robin Messer

Overview

- Westinghouse are developing a new signalling and train control system
- We must have confidence that the software meets its key safety properties
- Will describe how the key safety properties are identified
- ... and how we assure ourselves that they are met

Distance To Go – Radio (DTG-R)

- Signalling and Automatic Train Control system for underground trains (initially for Victoria Line)
- Signalling information from interlocking transmitted to trains via radio system
- Automatic Train Operation (ATO) drives the train
- Automatic Train Protection (ATP) ensures the train does not exceed speed limits or pass red signals



Automatic Train Protection (ATP)

- Custom hardware manufactured by WRSL
- 3-lane architecture with 2oo3 voting



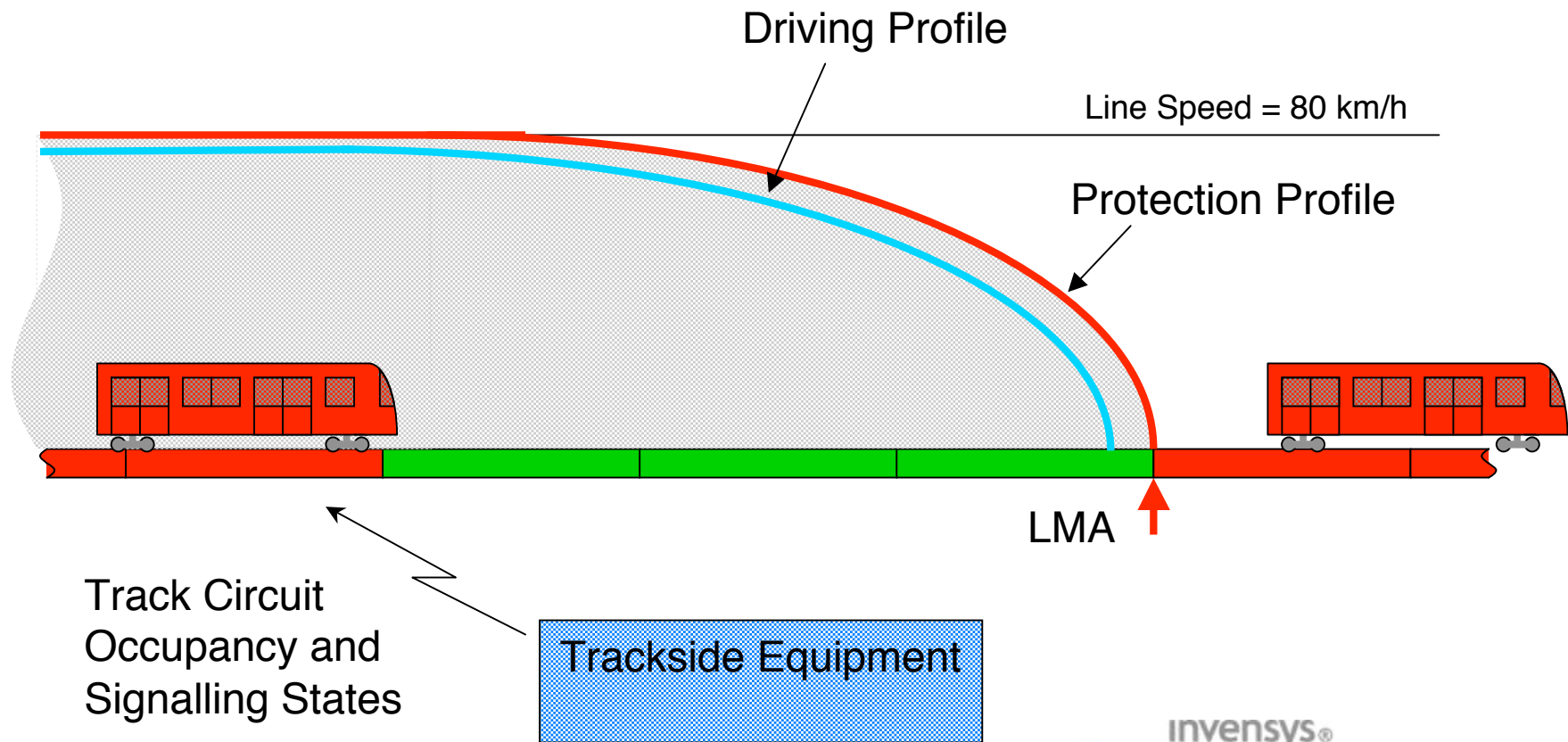
ATP Software

- Software developed by Aerosystems International
- Artisan UML model derived from Westinghouse DOORS requirements
- SPARK implementation developed within Artisan model
- SPARK static analysis and RTE proof done by developers before submitting code for review
- Compiled using Aonix C-SMART (Ada 83)

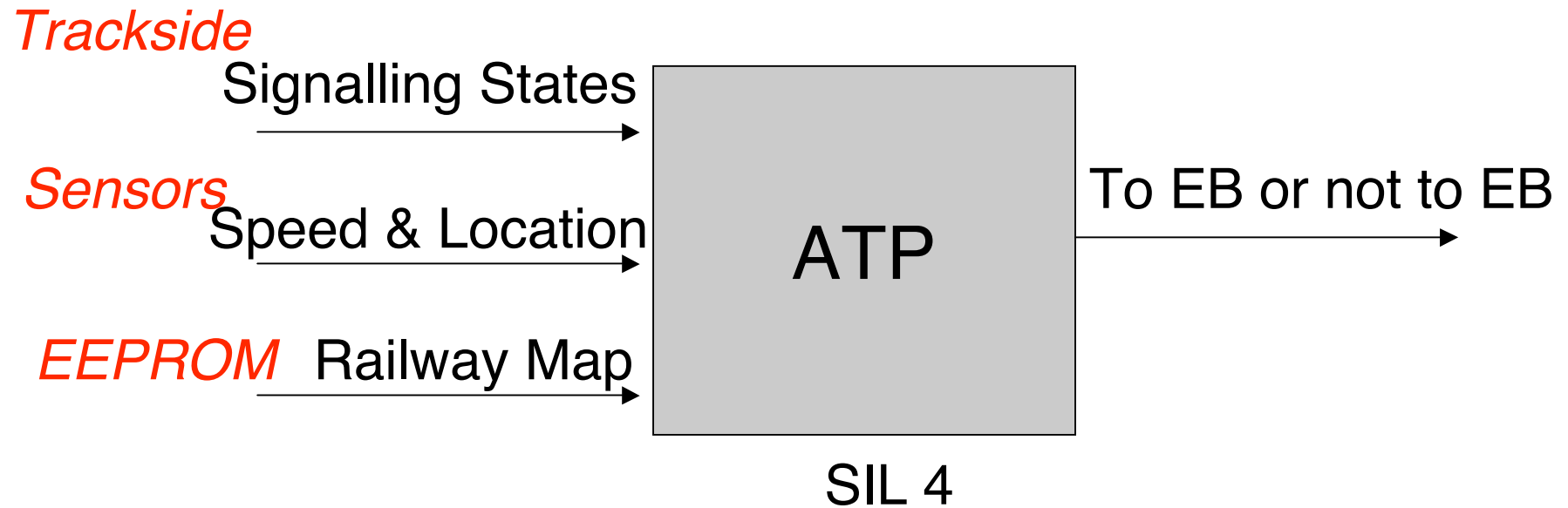
ATP Functionality

- Calculate ‘Movement Authority’ (how far train is permitted to travel) based on signalling states from interlocking
- Apply emergency brakes if train is predicted to exceed movement authority or current speed limit
(prevents **collision** or **derailment**)
- Enable train doors on the correct side when stopped at platform
(prevents passengers falling onto the track)

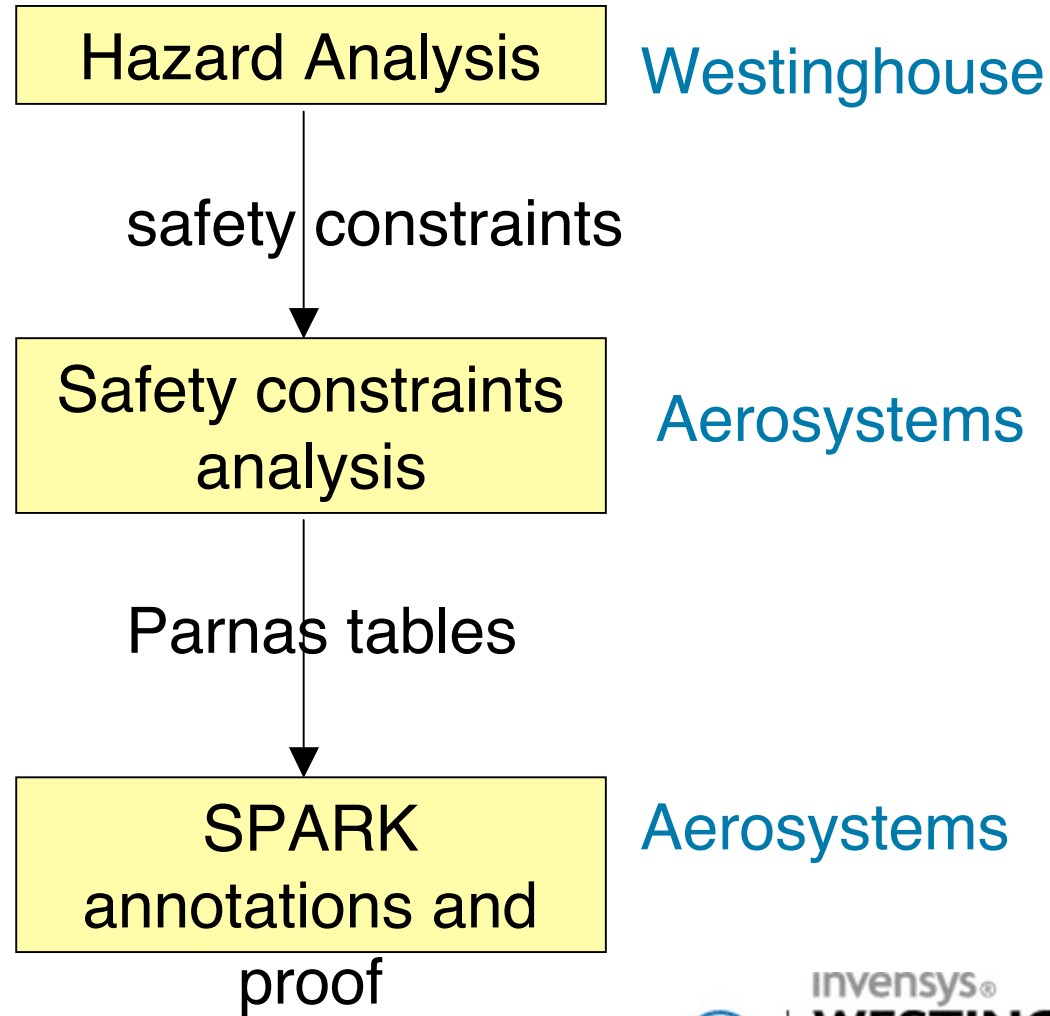
ATP Functionality



ATP Functionality



Overview of Hazard Analysis Process



Hazard Analysis Process

- Functional Failure Analysis of ATP requirements specification used to derive product-level ATP hazards (which are generally causes of system-level hazards)
- Risk assessment exercise performed to assess severity and likelihood of accident associated with each system level hazard
- Iterative process in parallel with developing product requirements

Example System Hazard

Hazard	Consequence	Severity	Mitigation
Overspeeding	Derailment	Catastrophic	ATP will ensure that speed restrictions are met

(assessment of likelihood not shown)

ATP Safety Constraints

- Properties of the ATP which provide mitigation against catastrophic accidents are classified as “safety constraints”
- We must have the highest confidence that these properties hold
- Safety constraint related to previous example is:

The ATP shall apply the emergency brakes unless:

- *train is not exceeding safe speed now*
- *train would not exceed safe speed for track ahead if emergency brakes were applied now*

Safety Analysis of UML

- Aerosystems safety team perform detailed analysis of the UML analysis and design models to determine:
 - which use cases (and which sequence steps) contribute to meeting the safety constraints
 - which operations are critical
 - what must be true about those operations in order for the safety constraints to hold
- These properties of operations are expressed as Parnas tables

Parnas Tables

- Example Parnas table for an operation which contributes to previous example safety constraint:

Train speed > safe speed for current area	Lookahead predicts safe speed will be exceeded	Braking condition (overspeed)
true	X	true
X	true	true

SPARK Annotations

- Parnas tables on operations are used to derive the SPARK proof annotations on the source code subprograms

```
procedure Overspeed_Check (Speed      : in      Train_speed;  
                           Location    : in      Train_loc;  
                           LA_Result   : in      LA_output;  
                           Apply_Brake :        out Brake_cond);  
  
--# global Speed_Limits;  
--# derives Apply_Brake from Speed,  
                           Location,  
                           LA_Result,  
                           Speed_Limits;  
  
--# post ((Speed > Speed_Limits(Location)) or  
--#      (LA_Result = Overspeed_Predicted))  
--#      -> Apply_Brake = Braking_Requested;
```

ATP Safety Constraints

- The SPARK Examiner analyses the code and generates Verification Conditions (VCs) which capture what is known (hypotheses) and what must be shown to be true (conclusions) for each path through the code in order to satisfy the postconditions
- The VCs are then discharged using the SPARK proof tools:
 - Automatic Simplifier – proves the majority of VCs entirely automatically using built-in proof rules
 - Proof Checker – an interactive tool which enables the user to guide the proof of the remaining VCs

How far have we got?

- Software development in progress – will not be able to complete the proof work before the software is complete
- Rigorous analysis process has uncovered faults in the software at an early stage
- Anecdotal evidence is that use of static analysis tools has led to fewer faults being found at integration test

Metrics so far

- 11 safety constraints specified
- 39 safety critical Requirements Model Use Cases (out of 77)
- 19 safety critical Use Cases added at Design stage (out of 33)
- 49 faults (5 major) found by analysis of Requirements Model
- 106 faults (15 major) found by analysis of Design Model
- (Out of approximately 1000 faults raised project-wide)
- Less than 6% of project effort spent on safety analysis so far

Lessons learned

- Anecdotal evidence from developers that performing static analysis (including proof of absence of runtime exceptions) prior to checking code in for review results in fewer issues being raised during review and test
- Analysis that traces safety constraints through design to code can provide valuable independent verification
- A balance has to be struck as to how much effort to put into safety analysis early in the project:
 - If analysis is left to the very end then it may uncover significant problems late in the project.
 - If analysis is done early then it may find problems early and it may influence the way the software is written, but there will be effort to rework it as the software develops.

Conclusions

- Evidence so far is that partial proof of correctness is proving to be practical for a significant (approx 40KLOC so far) software development
- The process is helping to provide confidence that the key safety risks have been reduced to a level that is “As Low As Reasonably Practicable”



Invensys®

WESTINGHOUSE®
rail systems