Overview

• 1. formal mind and its tools
• 2. data validation
• 3. the future
background

• gmbh (limited), based in düsseldorf, germany
• spin-off from university of düsseldorf
• expert services: formal verification, requirements management & engineering
• open source software: ProB & ProR
Systems Development today

Goal → Requirements → Specification → Implementation

Requirements Engineering:
- formalmind

Formal Specifications:
- formalmind

Requirements Engineering Platform:
- ProR

Formal Validation Platform:
- ProB
Requirements Engineering

- ProR
  - Optimizing Communications
  - Integration into existing processes
  - Interoperability with the ReqIF standard
ReqIF support

Traceability to B
Classifying informal and formal artefacts

Support for classifying informal and formal artefacts as W (domain properties), R (requirements) and S (specification).
Manual creation of traces between requirements and formal model elements is supported via drag and drop. The right column “Link” of the specification editor summarizes the number of outgoing (target) and incoming (source) traces. Selecting an outgoing trace shows the targets properties in the Properties View. Furthermore, traces can be annotated if additional information is necessary.
Tracing of phenomena used in artefacts

In order to add a uses-trace for a phenomenon to an artefact, the corresponding text passage is put in square brackets.

Red marked text passages reminds the user that an undeclared phenomena is used.

Unmarked, recognised phenomena are highlighted as well to warn the user about a possible omission.

Blue marked text passages are recognised phenomena.
Change management

When traced formal model elements change, the trace is marked as “suspect” by showing a small icon. Two columns exist for the source and the target of the trace, respectively. The user sees at a glance which requirements or formal model elements need to be revalidated. This is particularly useful if the requirements document becomes large. By double-clicking on the “suspect” icon, the user can mark the trace as “revalidated” and the icon will be removed.
now: Eclipse Foundation Project!

http://www.eclipse.org/rmf/
Formal Specifications

- More efficiency in validating formal specifications
- Optimizing existing tool chains
- Supports compliance with safety standards
Validation tool for high-level formal models

\[
\begin{align*}
&\text{Animation} & & \text{Model Checking} & & \text{Constraint-Based Checking} \\
\end{align*}
\]
what can ProB do for me?

my model

- Model Checking
- Constraint-Based Checking
- Test Generation
- Animation
- Visualization

- B, Event-B, Z, TLA

- Infinite Functions
- Large Data Values
- Constraint Solving
- Extensive Testing & Validation

Infrastructure
Coverage Reports

Industrial Usage: Aistom, Siemens, ...
BMotionStudio

• on top of ProB

• Editor:
  • link graphical elements with B expressions and predicates

• Important so that **domain experts** can detect errors in your models
B MotionStudio
2. Data Validation
M
1
s'automatiser
Worldwide implementations (2012) of systems embedding software generated from B models.
<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>IP</th>
<th>Type</th>
<th>UpLink</th>
<th>DownLink</th>
<th>Length</th>
<th>GPS 1</th>
<th>GPS 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route_tx_001</td>
<td>243</td>
<td>R</td>
<td>Route_tx_005</td>
<td>Route_tx_006</td>
<td>345</td>
<td>291</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Route_vx_002</td>
<td>128</td>
<td>R</td>
<td>Route_vx_002</td>
<td>EndLine_002</td>
<td>128</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Switch_w_003</td>
<td>256</td>
<td>192.16.4.55</td>
<td>S</td>
<td>Route_vx_128</td>
<td>Route_tx_006</td>
<td>23</td>
<td>N 50.85 963</td>
<td>O 6.84 201</td>
</tr>
<tr>
<td>Relay_s_004</td>
<td>12</td>
<td>192.16.4.10</td>
<td>Y</td>
<td>Route_vx_006</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relay_s_001</td>
<td>55</td>
<td>192.16.4.125</td>
<td>Y</td>
<td>Route_vx_006</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Route_tx_005</td>
<td>3</td>
<td>R</td>
<td>Route_tx_006</td>
<td>Route_vx_128</td>
<td>291</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relay_s_003</td>
<td>55</td>
<td>192.16.4.125</td>
<td>Y</td>
<td>Route_vx_006</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Route_vx_006</td>
<td>22</td>
<td>R</td>
<td>EndLine_001</td>
<td>Route_vx_002</td>
<td>110</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Route_vx_128</td>
<td>127</td>
<td>R</td>
<td>Route_tx_006</td>
<td>Route_vx_006</td>
<td>145</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Switch_w_009</td>
<td>242</td>
<td>192.16.4.10</td>
<td>S</td>
<td>Route_vx_128</td>
<td>Route_tx_005</td>
<td>34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EndLine_000</td>
<td>0</td>
<td>E</td>
<td>Route_vx_002</td>
<td>Route_vx_002</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EndLine_001</td>
<td>1</td>
<td>E</td>
<td>Route_vx_002</td>
<td>Route_vx_002</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Signal_xs_002</td>
<td>32</td>
<td>192.16.4.12</td>
<td>G</td>
<td>Route_vx_128</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Signal_xs_003</td>
<td>33</td>
<td>192.16.4.13</td>
<td>G</td>
<td>Route_vx_128</td>
<td>51</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balise_b_001</td>
<td>301</td>
<td>B</td>
<td>Route_vx_128</td>
<td>Route_vx_006</td>
<td>0 N 50.85 933</td>
<td>O 6.84 508</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balise_b_002</td>
<td>302</td>
<td>B</td>
<td>Route_tx_005</td>
<td>Route_tx_005</td>
<td>0 N 50.86 123</td>
<td>O 6.84 550</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Assertion:**
- Q3 |F = Q2: 0m00.026s, 0m00.763s
- Q2 |F = Q1: 0m00.025s, 0m00.765s
- Q3_DIV |F = DIV: 0m00.026s, 0m00.764s
- DIV |F = Q3_DIV: 0m00.026s, 0m00.704s
- Q2 |F = Q4: 0m00.026s, 0m00.710s
- Q4 |F = Q2: 0m00.026s, 0m00.779s
- Q2 |F = Q4: 0m00.027s, 0m00.778s
- Q4 |F = Q2: 0m00.026s, 0m00.749s
- Summarized Time: 0m00.209s, 0m05.611s

**Phone Number:**
- 01-212-555-1234
- 01-212-555-4321

**Fax Number:**
- 01-212-555-1234
- 01-212-555-4321

**E-Mail Address:**
- someone@example.com
Is it consistent?

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>IP</th>
<th>Type</th>
<th>Uplink</th>
<th>Dnlink</th>
<th>Length</th>
<th>GPS 1</th>
<th>GPS 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route tx_001</td>
<td>243</td>
<td>192.16.4.10</td>
<td>Y</td>
<td>Route tx_005</td>
<td>Route tx_002</td>
<td>345</td>
<td>N 50.85 963</td>
<td>O 6.84 201</td>
</tr>
<tr>
<td>Route vx_002</td>
<td>128</td>
<td></td>
<td></td>
<td>Route vx_002</td>
<td>Route vx_128</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Switch w_003</td>
<td>256</td>
<td>192.16.4.55</td>
<td>Y</td>
<td>Route vx_128</td>
<td>Route tx_006</td>
<td>23</td>
<td>N 50.85 963</td>
<td>O 6.84 201</td>
</tr>
<tr>
<td>Relay s_004</td>
<td>12</td>
<td>192.16.4.10</td>
<td>Y</td>
<td>Route tx_006</td>
<td>Route vx_128</td>
<td>291</td>
<td>N 50.85 963</td>
<td>O 6.84 201</td>
</tr>
<tr>
<td>Relay s_001</td>
<td>55</td>
<td>192.16.4.125</td>
<td></td>
<td>Route tx_001</td>
<td>Route vx_002</td>
<td>110</td>
<td>N 50.85 963</td>
<td>O 6.84 201</td>
</tr>
<tr>
<td>Route tx_006</td>
<td>22</td>
<td></td>
<td></td>
<td>EndLine_001</td>
<td>Route vx_002</td>
<td>145</td>
<td>N 50.85 963</td>
<td>O 6.84 201</td>
</tr>
<tr>
<td>Route vx_128</td>
<td>127</td>
<td></td>
<td></td>
<td>Route tx_006</td>
<td>Route vx_002</td>
<td>145</td>
<td>N 50.85 963</td>
<td>O 6.84 201</td>
</tr>
<tr>
<td>Switch w_009</td>
<td>42</td>
<td>192.16.4.10</td>
<td>S</td>
<td>Route vx_128</td>
<td>Route tx_005</td>
<td>34</td>
<td>N 50.85 963</td>
<td>O 6.84 201</td>
</tr>
<tr>
<td>EndLine_000</td>
<td>0</td>
<td>192.16.4.10</td>
<td>Y</td>
<td>Route vx_002</td>
<td>Route vx_002</td>
<td>1</td>
<td>N 50.85 963</td>
<td>O 6.84 201</td>
</tr>
<tr>
<td>EndLine_001</td>
<td>1</td>
<td></td>
<td></td>
<td>Route vx_002</td>
<td>Route vx_002</td>
<td>1</td>
<td>N 50.85 963</td>
<td>O 6.84 201</td>
</tr>
<tr>
<td>Signal xs_002</td>
<td>32</td>
<td></td>
<td></td>
<td>Route vx_128</td>
<td>Route vx_002</td>
<td>22</td>
<td>N 50.85 963</td>
<td>O 6.84 508</td>
</tr>
<tr>
<td>Signal xs_003</td>
<td>33</td>
<td>192.16.4.13</td>
<td>G</td>
<td>Route vx_006</td>
<td>Route tx_005</td>
<td>51</td>
<td>N 50.85 933</td>
<td>O 6.84 550</td>
</tr>
<tr>
<td>Balise b_001</td>
<td>301</td>
<td></td>
<td></td>
<td>Route vx_128</td>
<td>Route tx_005</td>
<td>0</td>
<td>N 50.85 123</td>
<td>O 6.84 550</td>
</tr>
<tr>
<td>Balise b_002</td>
<td>302</td>
<td></td>
<td></td>
<td>Route tx_005</td>
<td>Route tx_005</td>
<td>0</td>
<td>N 50.86 123</td>
<td>O 6.84 550</td>
</tr>
</tbody>
</table>

Is it safe?

Is it correct?
Dream
### Formal Properties

Error location & diagnosis

high-level language (B,...)
MACHINE SearchForFile
USES LibraryFiles, LibraryStrings

DEFINITIONS
4 target == "sicstus";
5 GOAL == (found=TRUE);
6 SET_PREF_MAX_OPERATIONS == 256
7 VARIABLES cur, found
8 INVARIANT
9 cur : STRING & found : BOOL
10 INITIALISATION cur :=="/usr/local" || found := FALSE
11 OPERATIONS
12 r <-- Found MORE target : files(cur) THEN r := cur ||
13 NavigateInto = PRE x:directories(cur) THEN cur := a /
14 IsFile(ex) = f:files(cur) THEN skip END /
15 END
16
17
18
19
20

{} directories("/usr/local/lib")

{"HTTP-3001.1.4","ImageMagick-6.2.9","ImageMagick-6",
"coq","fpc","gettext","graphviz","pkgconfig","xema
0.5.0.0"}

files(cur)

{"sicstus","sicstus-4.1.1","sicstus-4.1.3","spconfi
","spdet-4.1.1","spl","spl","spl","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp","sp"
how it all began

• Deploy Project: Scalability issue at Siemens for data validation

Considerable work

• Generic B model
• Text Data with the entire line (all segments)

Problems:
• Make sure the assumptions in the B model are correct for all segments
147 Assertions
147 Assertions

\[ t_{\text{iti_partiel_acs}} \land bb : cfg_{\text{cdv_aig}} \land aa \rightarrow bb : t_{\text{iti_partiel_acs}} \land bb \land \langle \text{cfg_ipart_cdv_transit_dernier_i} \mid > \text{cfg_cdv_aig} \rightarrow bb : \text{cfg_ipart_cdv_transit_liste_i}(\text{cfg_ipart_cdv_transit_deb}(aa) \ldots \text{cfg_ipart_cdv_transit_fin}(aa))) \rangle \text{cfg_ipart_pc1_adj_i}^{-\{\{\text{TRUE}\}\}} \leq\{\{\text{TRUE}\}\} \land \text{cfg_ipart_pc2_adj_i}^{-\{\{\text{TRUE}\}\}} = \{\}
\]

\[ \text{cfg_ipart_aig_tild_liste_i}^{-\{t_{\text{iti_partiel_acs}}\}} < t_{\text{liste_acs}} \]

\[ \text{cfg_ipart_aig_tild_liste_i}^{-\{t_{\text{iti_partiel_acs}}\}} < \text{NATURAL} \]

\[ \text{cfg_ipart_aig_liste_i}^{-\{t_{\text{aig_acs}}\}} < t_{\text{liste_acs}} \]

\[ \text{cfg_ipart_aig_liste_i}^{-\{t_{\text{aig_acs}}\}} < \text{NATURAL} \]

\[ \text{cfg_ipart_cdv_transit_liste_i}^{-\{\text{cfg_cdv_aig}\}} < t_{\text{liste_acs}} \]

\[ \text{cfg_ipart_cdv_transit_liste_i}^{-\{\text{cfg_cdv_aig}\}} < \text{NATURAL} \]

\[ \text{cfg_ipart_cdv_zdest_sscant_liste_i}^{-\{\text{cfg_cdv_block}\}} < t_{\text{liste_acs}} \]

\[ \text{cfg_ipart_cdv_zdest_sscant_liste_i}^{-\{\text{cfg_cdv_block}\}} < \text{NATURAL} \]
147 Assertions
situation before deploy

san juan: 80 assertions had to be checked manually
current situation

• validation can be done in minutes
• more reliable, better feedback
• much bigger and new problems can be tackled (on-board data)

The work done with ProB is a great success. Thanks to the automatization and ProB, the wayside data validation is quicker, easier and complete.
ProB est utilisé pour la vérification des paramètres de systèmes de signalisation ferroviaire d'Alstom Transport.
Il a permis notamment de réduire le coût et la durée de la vérification des paramètres des systèmes installés à Santiago du Chili, Panama City et Mexico City.
En outre Alstom Transport expérimente ProB pour la génération de paramètres et pour la validation de modèles formels de systèmes de signalisation.
Luis-Fernando Mejia, Alstom
São Paulo line 4

- 210 files, >30,000 lines of B
- >2500 assertions, >32,000 properties to be checked
- some very large sets (e.g., 1..65535), some infinite sets (e.g., INTEGER - {x})
- inconsistencies found!

<table>
<thead>
<tr>
<th>Sector</th>
<th>Predicates</th>
<th>TRUE</th>
<th>FALSE</th>
<th>UNKNOWN</th>
<th>TIMEOUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>pas.as.inv.s036.html</td>
<td>1465</td>
<td>1459</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>pas.as.inv.s037.html</td>
<td>1165</td>
<td>1460</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>pas.as.inv.s038.html</td>
<td>1465</td>
<td>1457</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 1.3 Paris line 1 (ZC)

<table>
<thead>
<tr>
<th>Sector</th>
<th>Predicates</th>
<th>TRUE</th>
<th>FALSE</th>
<th>UNKNOWN</th>
<th>TIMEOUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>pas.as_inv_s011.html</td>
<td>1503</td>
<td>1501</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>pas.as_inv_s012.html</td>
<td>1503</td>
<td>1498</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>pas.as_inv_s013.html</td>
<td>1503</td>
<td>1496</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>pas.as_inv_s014.html</td>
<td>1503</td>
<td>1499</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>pas.as_inv_s015.html</td>
<td>1503</td>
<td>1498</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>pas.as_inv_s016.html</td>
<td>1503</td>
<td>1498</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Paris line 1 (PAL)  . PAL (Pilote Automatique Ligne) is a controller line who realizes the Automatic Train Supervision (ATS) function of CBTC systems. The B models of PAL consisted of 74 files with over 10,000 lines of B. In all 2024 assertions about concrete data of the PAL needed to be checked. ProB found 12 in under 5 minutes. These problems have been examined and confirmed by manual inspection afterward at Siemens.
Why B?

Why ProB?

Challenges
MACHINE SearchForFile
USES LibraryFiles, LibraryStrings
DEFINITIONS
  target == "sicstus";
  GOAL == (found=TRUE);
  SET_PREF_MAX_OPERATIONS == 256
VARIABLES cur, found
INARIANT
  cur : STRING & found : BOOL
INITIALISATION cur := "/usr/local" || found := FALSE
OPERATIONS
  r <-- Found THEN PRE target : files(cur) THEN r := cur ||
  NavigateInto = PRE x:directories(cur) THEN cur := a x:
  IsFile = f:files(cur) THEN skip END /

INVARIANT
  files = %x.(x : STRING|FILES/*EXT:*/(x))
directories = %x.(x : STRING|DIRECTORIES/current_directory = "."
file_exists = %x.(x : STRING|bool(FILE_EXITS)
directory_exists = %x.(x : STRING|bool(DIR
append = %x.(y ,x : STRING & y : STRING)
length = %x.(x : STRING|STRING_LENGTH/
split = %x.(y ,x : STRING & y : STRING|STRING
cur = "/usr/local/sicstus4.1/bin"
founds = TRUE

{} directories("/usr/local/lib")
{"HTTP-3001.1.4", "ImageMagick-6.2.9", "ImageMagick-6
, coq", fpc", gettext", "graphviz", "pkgconfig", "xema
0.5.0.0")

>>> directories("/usr/local/lib")
{"HTTP-3001.1.4", "ImageMagick-6.2.9", "ImageMagick-6
, coq", fpc", gettext", "graphviz", "pkgconfig", "xema
0.5.0.0")

>>> files(cur)
{"sicstus", "sicstus-4.1.1", "sicstus-4.1.3", "spconfi
, spdeg-4.1.1", "spdeg-4.1.3", "spl", "spll-4.1
, spdm-4.1.1", "spdm-

>>> cur = files(cur))

20

---

<table>
<thead>
<tr>
<th>OK</th>
<th>State Properties</th>
<th>Enabled Operations</th>
<th>History</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>invariant_ok</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>files = %x.(x : STRING</td>
<td>FILES/<em>EXT:</em>/(x))</td>
<td></td>
</tr>
<tr>
<td></td>
<td>directories = %x.(x : STRING</td>
<td>DIRECTORIES/current_directory = &quot;.&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>file_exists = %x.(x : STRING</td>
<td>bool(FILE_EXITS)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>directory_exists = %x.(x : STRING</td>
<td>bool(DIR</td>
<td></td>
</tr>
<tr>
<td></td>
<td>append = %x.(y ,x : STRING &amp; y : STRING)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>length = %x.(x : STRING</td>
<td>STRING_LENGTH/</td>
<td></td>
</tr>
<tr>
<td></td>
<td>split = %x.(y ,x : STRING &amp; y : STRING</td>
<td>STRING</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cur = &quot;/usr/local/sicstus4.1/bin&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>found = TRUE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Why B & ProB

The most expressive language in the world
> 2,500 years of human experience
Unambiguous,
Easy to express,
Easy to understand,
Easy to adapt

Intelligible Feedback on your data
Why ProB

In use by Siemens, Alstom,...
Why ProB

Open Source Core
Command-line interfaces
Java-API

Can fit into your process
very expressive

efficient

validated

good feedback

probcli fits into design flow
DTVT

- Tool based on ProB developed by
  - ClearSy, Alstom, Formal Mind
**E_a_trainDynamicDeparture_minimum_speed**

**Description:**
Train dynamic departure minimum speed

**Typage:**
E_a_trainDynamicDeparture_minimum_speed : INT ==> FLOAT

Range Excel du domaine : Train_Dynamics!A7:A27
Range Excel du codomaine : Train_Dynamics!AM7:AM27

---

**Propriété VS_C52**

Règle associée : 104

**Description:**
Les zones de freinage ne se recouvrent pas

**Expression formelle:**

```python
(r1, r2) - r1 == t_regenerativeBraking &
   r2 == t_regenerativeBraking &
   r1 /= r2
   a_regenerativeBrakingArea(r1)
   a_regenerativeBrakingArea(r2) /
   F_areaEnterArea(r1)
```
the future

• B as a high-level
• query language
• constraint-solving language
• programming language
b as high-level query language

- data validation:
  - Siemens
  - Alstom, ClearSy, ...

- many properties can be conveniently expressed in B and now be checked on real data with ProB

- double chain possible: Ovado, ProB-Kodkod, PyProB (in development)
b as a high-level constraint solving language

- very easy to express properties:

@perm p ∈ Nodes → Nodes
@iso ∀x,y • (x ∈ Nodes ∧ y ∈ Nodes → (x ↦ y ∈ graph1 ⇔ p(x) ↦ p(y) ∈ graph2))

@ctype colour ∈ Vtx → 1 · · maxcol
@alldiff (∀i,j • i ↦ j ∈ graph1 ⇒ colour(i) ≠ colour(j))
**THE EVOLUTION OF PROGRAMMING**

**1954 FORTRAN**
C Hello World in Fortran 77
* (lines must be 6 characters indented)

```fortran
PROGRAM HELLO
WRITE(UNIT=*, FMT=*) 'Hello World'
END
```

**1958 LISP**
(print “Hello World”)

**1959 COBOL**

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HelloWorld.
AUTHOR. Fabritius.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
INPUT-OUTPUT SECTION.
DATA DIVISION.
FILE SECTION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.
DISPLAY "Hello World".
STOP RUN.
```

**1962 SIMULA**
```
begin
  OutText("Hello World");
  OutImage
end
```

**1964 BASIC**
```
PRINT "Hello World"
```

**1968 PASCAL**
```
PROGRAM HelloWorld;
BEGIN
  WRITELN('Hello World');
END.
```
2001 C#

```csharp
// Hello World in C#
using System;
class HelloWorld {
    static void Main() {
        Console.WriteLine("Hello World");
    }
}
```

2002 .NET

supports several programming languages which allows language interoperability (each language can use code written in other languages).

2005 RUBY ON RAILS

In Ruby, everything is an object
code = puts "Hello World!"

2009 NODE.JS

Written in JavaScript, reduces overhead on the web server.
```javascript
var http = require('http');
http.createServer(function (request, response) {
    response.writeHead(200, {'Content-Type': 'text/plain'});
    response.end('Hello World\n');
}).listen(8000);
console.log('Server running at http://localhost:8000/');
```

2012+ ????

What will the future bring?
MACHINE SearchForFile
USES LibraryFiles, LibraryStrings
DEFINITIONS
  target == "sicstus";
  GOAL == (found=TRUE);
  SET_PREF_MAX_OPERATIONS == 256
VARIABLES cur, found
INVARIANT
  cur : STRING & found : BOOL
INITIALISATION cur :="/usr/local" || found := FALSE
OPERATIONS
  r <-> Found r=cur
  PRE target : files(cur) THEN r := cur ||
  NavigateInto = PRE x:directories(cur) THEN cur := a
  IsFileEx(x) = f:files(cur) THEN skip END /
EN

Invariant_ok
files = %x.(x : STRING|FILES/*EXT:*/(x))
directories = %x.(x : STRING|DIRECTORIES/current_directory = "."
file_exists = %x.(x : STRING|bool(FILE_EXISTS)
  directory_exists = %x.(x : STRING|bool(DIR
  append = %x,y).(x : STRING & y : STRING)
  length = %x.(x : STRING|STRING_LENGTH/
  split = %x,y).(x : STRING & y : STRING|STR
cur = "/usr/local/sicstus4.1/bin"
found = TRUE

{} directories("/usr/local/lib")
{"HTTP-3001.1.4","ImageMagick-6.2.9","ImageMagick-6","coq","fpc","gettext","graphviz","pkgconfig","xema 0.5.0.0")

files(cur)
sicstus,sicstus-4.1.1,sicstus-4.1.3,spconfi
NavigateInto("sp-4.1.1")
NavigateInto("sp-4.1.3")

cur :="/usr/local/sicstus4.1/bin"

Initialisation("/usr/local",FALSE)
SETUP_CONSTANTS(%%x.(x : STRING|FILES/
b as a high-level programming language

- Alstom ongoing project:
  - large data (cf data validation)
  - find data (cf constraint solving)
  - computation: infinite functions, recursive functions, external functions (sin, cos, ...)
  - efficiency important
conclusion

• move systems engineering to the next level

• move B to next level

• ProB, BMotionStudio, ProR
thank you

backup slides
CBC Deadlock Checking

• successful Deploy case study with Bosch

*BPEL Example:*

SMT Solvers Plugin + Z3:

ProB:

```
$z3 -smt2 dlf1_z3.smt
...WARNING: pulled nested quantifier to be able to find an useable pattern (quantifier id: k!405) unknown
```
Latest Bosch Experiment

- 78 constants, 121 axioms, 62 variables, 59 invariants, 80 events, 855 guards

- card: $79 \times \infty, 1 \times 2^{65}, 1 \times 2^{52}, 11 \times 2^{32}, \ldots$

- 34 pages of A4 formula solved in 1-2 seconds

- So far: no success with SMT/SAT
for Bosch example

kodkod.engine.CapacityExceededException: Arity too large (10) for a universe of size 35  at
kodkod.instance.TupleFactory.checkCapacity(TupleFactory.java:266)  at
kodkod.instance.TupleFactory$IntTuple.<init>(TupleFactory.java:325)  at
kodkod.instance.TupleFactory.tuple(TupleFactory.java:88)  at
de.stups.probkodkod.types<TupleType.createAllTuples(TupleType.java:95)  at
de.stups.probkodkod.KodkodAnalysis.extractTupleSet(KodkodAnalysis.java:725)  at
de.stups.probkodkod.KodkodAnalysis.addRelations(KodkodAnalysis.java:666)  at
de.stups.probkodkod.KodkodAnalysis.caseAProblem(KodkodAnalysis.java:246)  at
de.stups.probkodkod.parser.node.AProblem.apply(AProblem.java:75)  at
de.stups.probkodkod.parser.analysis.DepthFirstAdapter.caseAProblemAction(DepthFirstAdapter.java:55)  at
de.stups.probkodkod.parser.node.AProblemAction.apply(AProblemAction.java:34)  at
de.stups.probkodkod.parser.analysis.DepthFirstAdapter.caseStart(DepthFirstAdapter.java:34)  at
de.stups.probkodkod.parser.node.Start.apply(Start.java:36)  at
de.stups.probkodkod.KodkodInteraction.interaction(KodkodInteraction.java:54)  at
de.stups.probkodkod.KodkodInteraction.main(KodkodInteraction.java:95)