SIEMENS

Lean Engineering of Interlocking

ProRail

TU Delft

Siemens

Goals

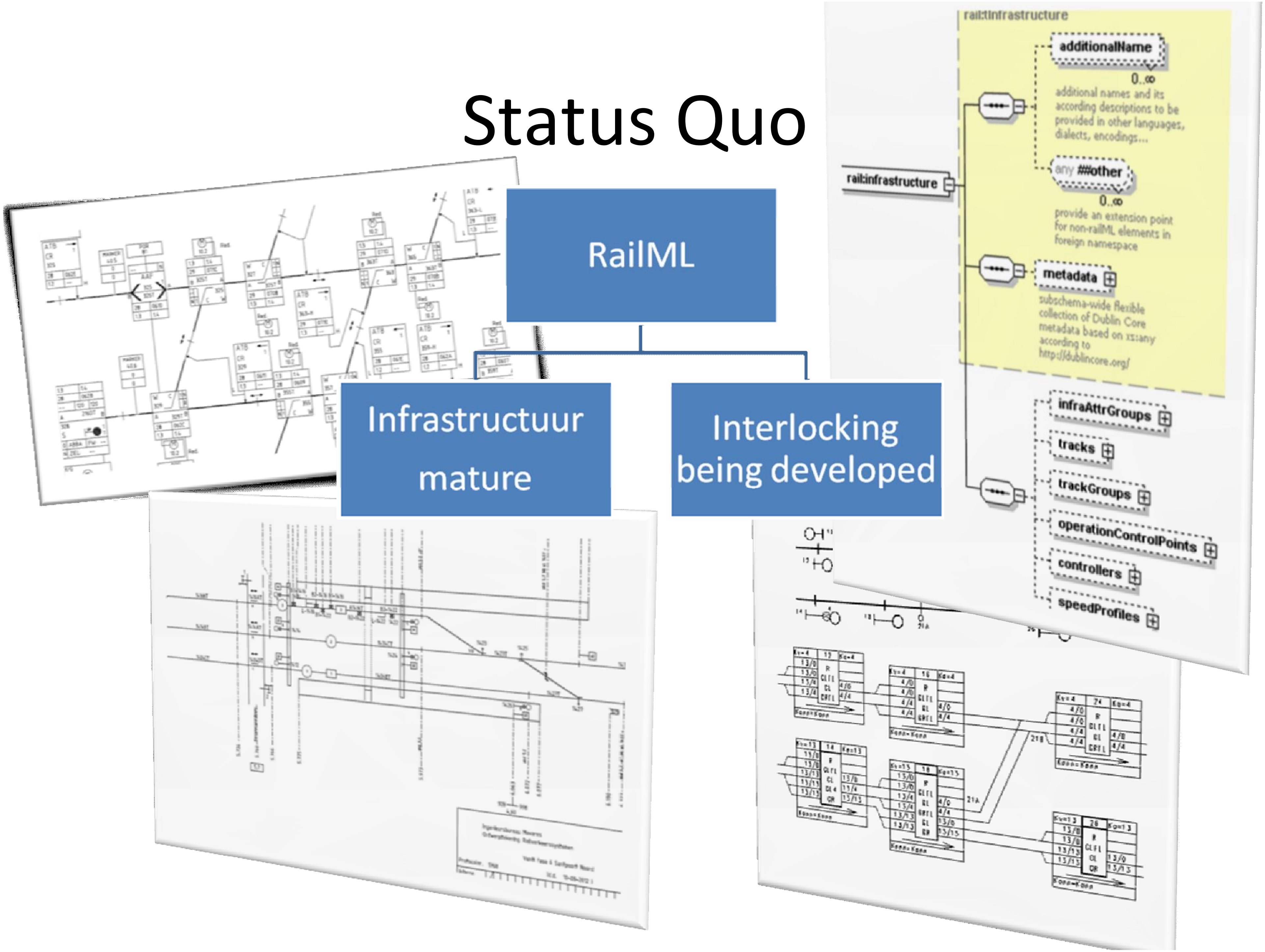
RailML for Lean Engineering

- High quality Interlocking
- At less cost

RailML model of a station

- Can we capture Prorail data in RailML ?
- Capture the data that an *IL needs*
- Anything missing ?

Status Quo


RailML
Infrastructuur mature
Interlocking being developed
rail:tInfrastructure
additionalName
0..∞
additional names and its according descriptions to be provided in other languages, dialects, encodings...
rail:infrastructure
any ##other
0..∞
provide an extension point for non-railML elements in foreign namespace
metadata
subschema-wide flexible collection of Dublin Core metadata based on xs:any according to http://dublincore.org/
infraAttrGroups
tracks
trackGroups
operationControlPoints
controllers
speedProfiles

Modelling Tenets

Interlocking must know topo

- IS schema captures topology well
- Graph approach is common
- e.g. walk a graph for finding route

Avoid manual data capture

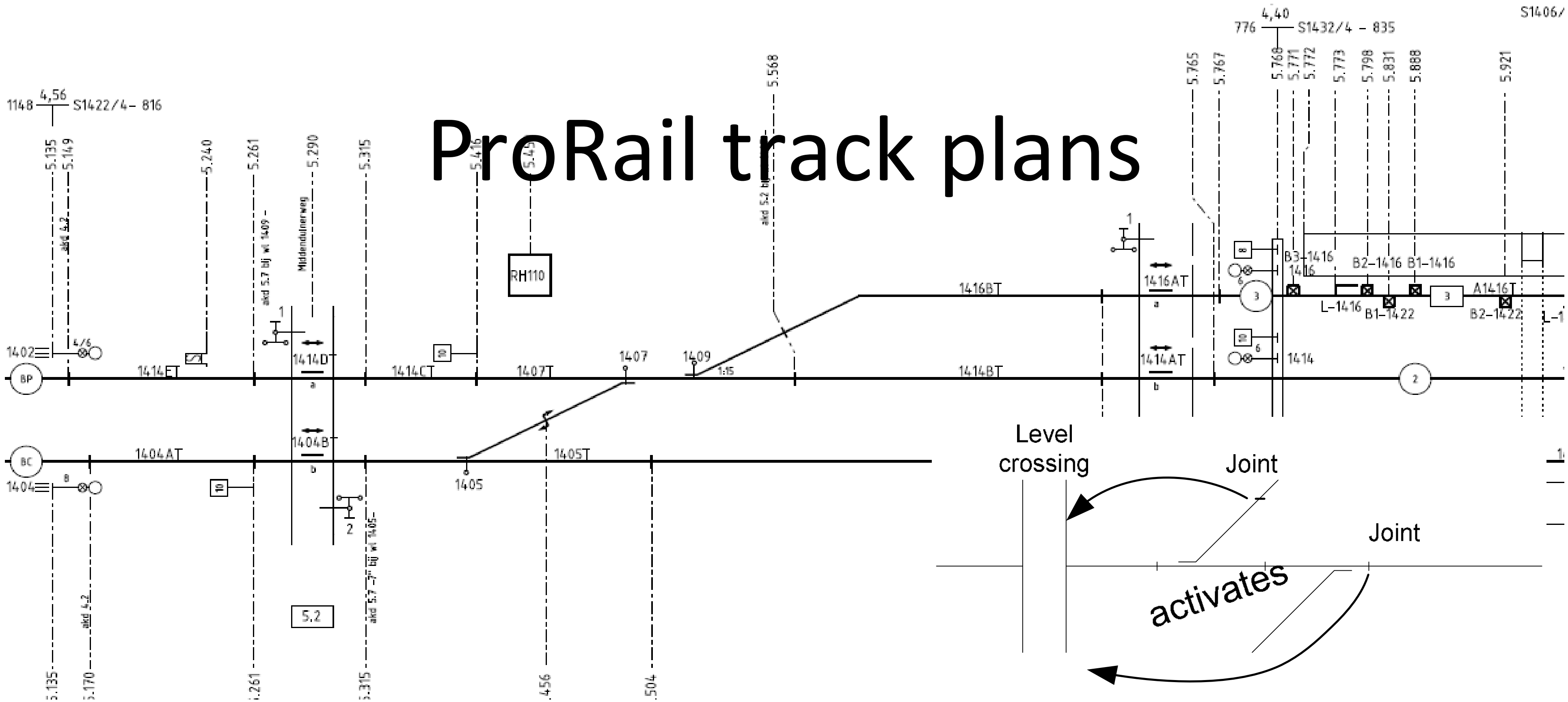
- Error prone
- Creates need for tests

Algorithms

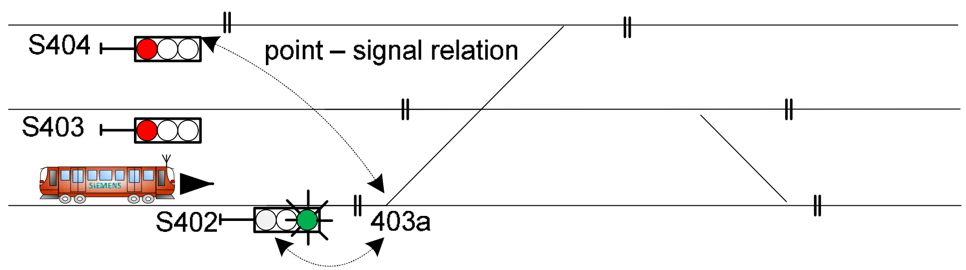
- ProRail already has “a dataset”
- Automatic conversion to RailML w
- Detect and prevents errors


```
description="Sptn verbindingsspoor Zuid">
<trackTopology>
  <trackbegin id="Sptn_1407V"
              pos="0"
              absPos="5.49">
    <connection id="Sptn_1407VR"
                ref="Sptn_1407RV" />
  </trackbegin>
  <trackend id="Sptn_1409V"
            pos="0.03"
            absPos="5.52">
    <connection id="Sptn_1409VR"
                ref="Sptn_1409RV" />
  </trackend>
  <connections>
    <switch id="Sptn_1407"
            pos="0">
      <connection id="Sptn_1407VL"
                  ref="Sptn_1407LV"
                  orientation="incoming"
                  course="right" />
    </switch>
    <switch id="Sptn_1409"
            pos="0,03">
      <connection id="Sptn_1409VL"
                  ref="Sptn_1409LV"
                  orientation="outgoing"
                  course="left" />
    </switch>
```

ProRail track plans



1. Can *mostly* be mapped to the Infrastructure schema
2. Contains *relations for use by the Interlocking schema*
   - LX – announcement
   - Flank protection
   - …



Data quality is essential

Avoid derived data

- e.g. age and date of birth
- Should we store both (x,y,z) and mileage ?

Automated data checks

- Are speedprofiles consistent with signs ?
- Given the topology, is flankprotection provided ?
- Can we transform interlocking design rules into algorithms ?

Precision and reliability

- Are distances precise ?
- Are distances reliable ?

Good data mean fewer tests

Signals are nodes and objects

```
                    pos="x"
                    absPos="!
                    vmax="10(
                    dir="down
        </speedchanges>
        <levelCrossings>
            <levelCrossing id="Hlm_
                     pos="x"
                     absPos='
        </levelCrossings>
    </trackElements>
    <ocsElements>
        <signals>
            <signal id="Sptn_1402"
                    pos="X"
                    absPos="5.135"
                    name="1402"
                    dir="up" />
            <signal id="Sptn_1407R_1600"
                    pos="x"
                    absPos="5.100"
                    dir="down"
                    virtual="true"
                    description="Open track to stati
            <signal id="Sptn_1407R_1601"
                    pos="x"
                    absPos="5.100"
                    dir="up"
                    virtual="true"
                    description="Open track to stati
            <signal id="X"
```

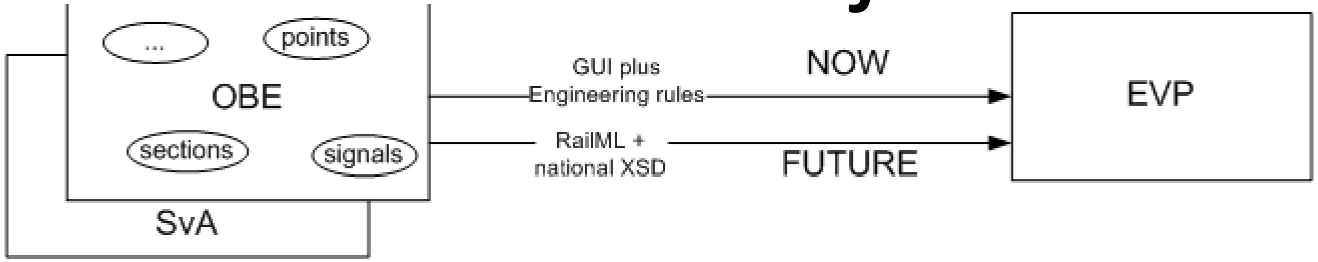


Figure 15 OBE to topology conversion. EVP is Siemens format

The Prorail specifica can be attached to the `any ##other` extension point in RailML. Prorail will need a schema for modelling signal attributes that can't be captured in standard RailML.

```
<!—Namespace xmlns:ns54=http://ergensBijProrail/ns/NS54.xsd>
...
<signals>
   <signal id="Sptn_1404" pos="X" absPos="5.135" name="BED_SEIN" dir="up">
      <ns54:bediendSein type="gloeilamp" lichtbak="true"
    techniek="glasvezel">
      <line switchable="true">
   </signal>
```

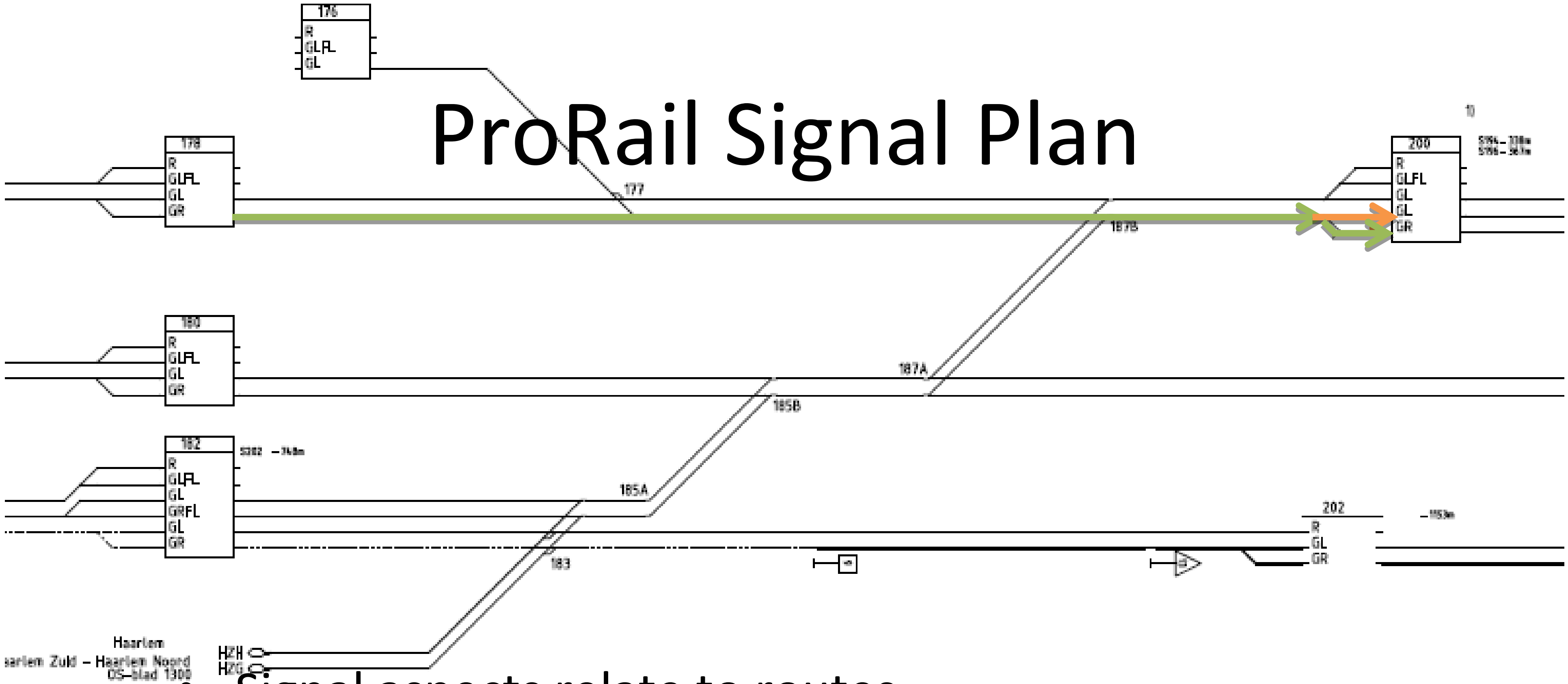
RailML IS provides generic signals

"our" signals will be an extension – IS or IL ?

An interlocking signal is a software object

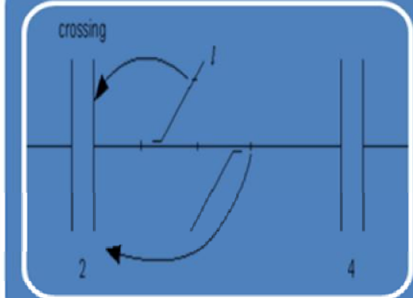
- Propertiese caught in RailML
- Methods *caught in Interlocking software*

ProRail Signal Plan



- Signal aspects relate to routes
- Interlocking sets signal aspect plus speed code according to signal plan
- Routes
  - can be composed from atomic routes
  - very much like linked lists
  - NO need to capture composite routes

Conclusions and recommendations (1)



Build the bridge between IS en IL

- IL needs associations between elements
- Route setting needn't be complex...

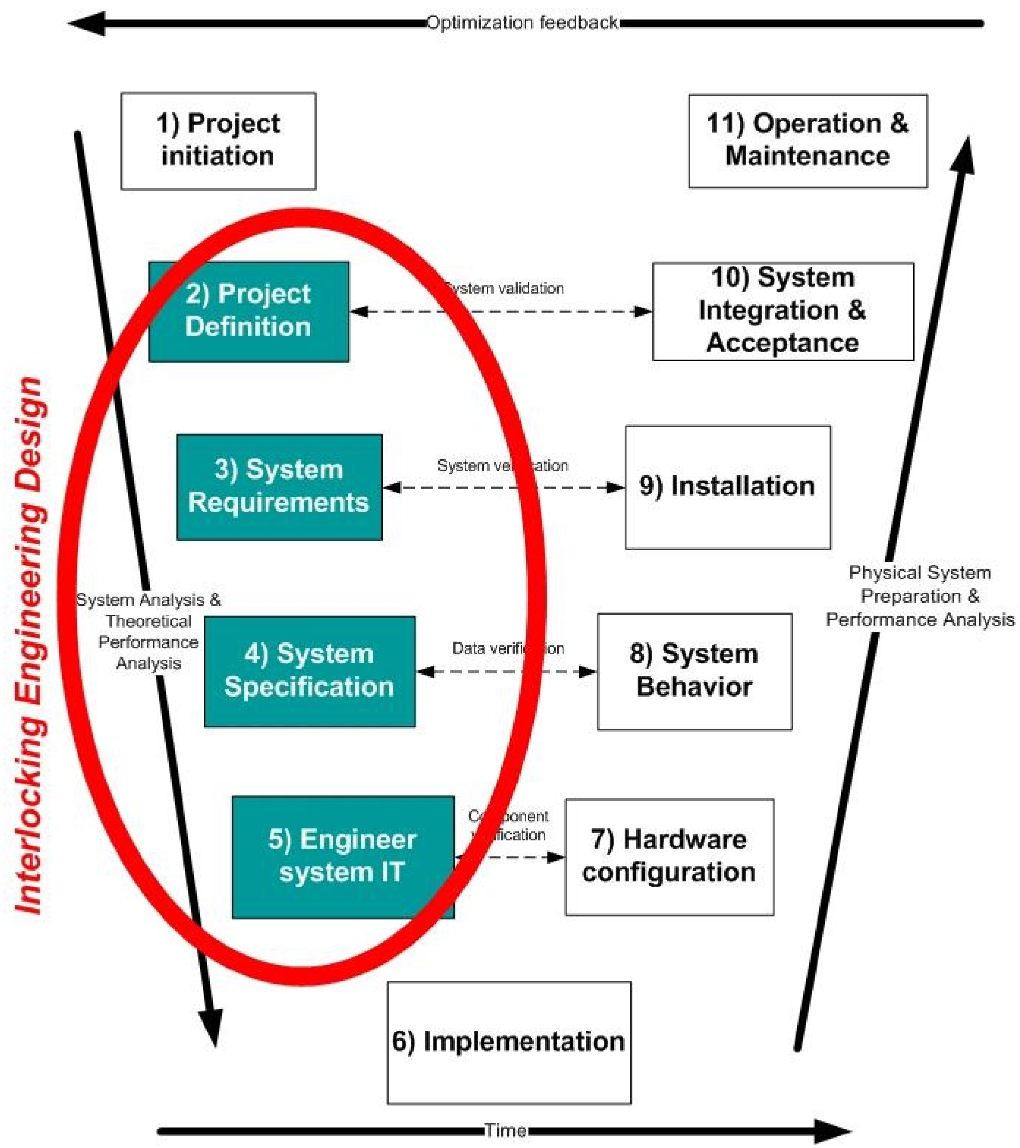
Visualise RailML

- Integration with designtools, GIS
- Users prefer map-based tools to raw data
- RailML can store various coordinatesystems
- Avoid that OCS displays, track plans etc. lead separate lives

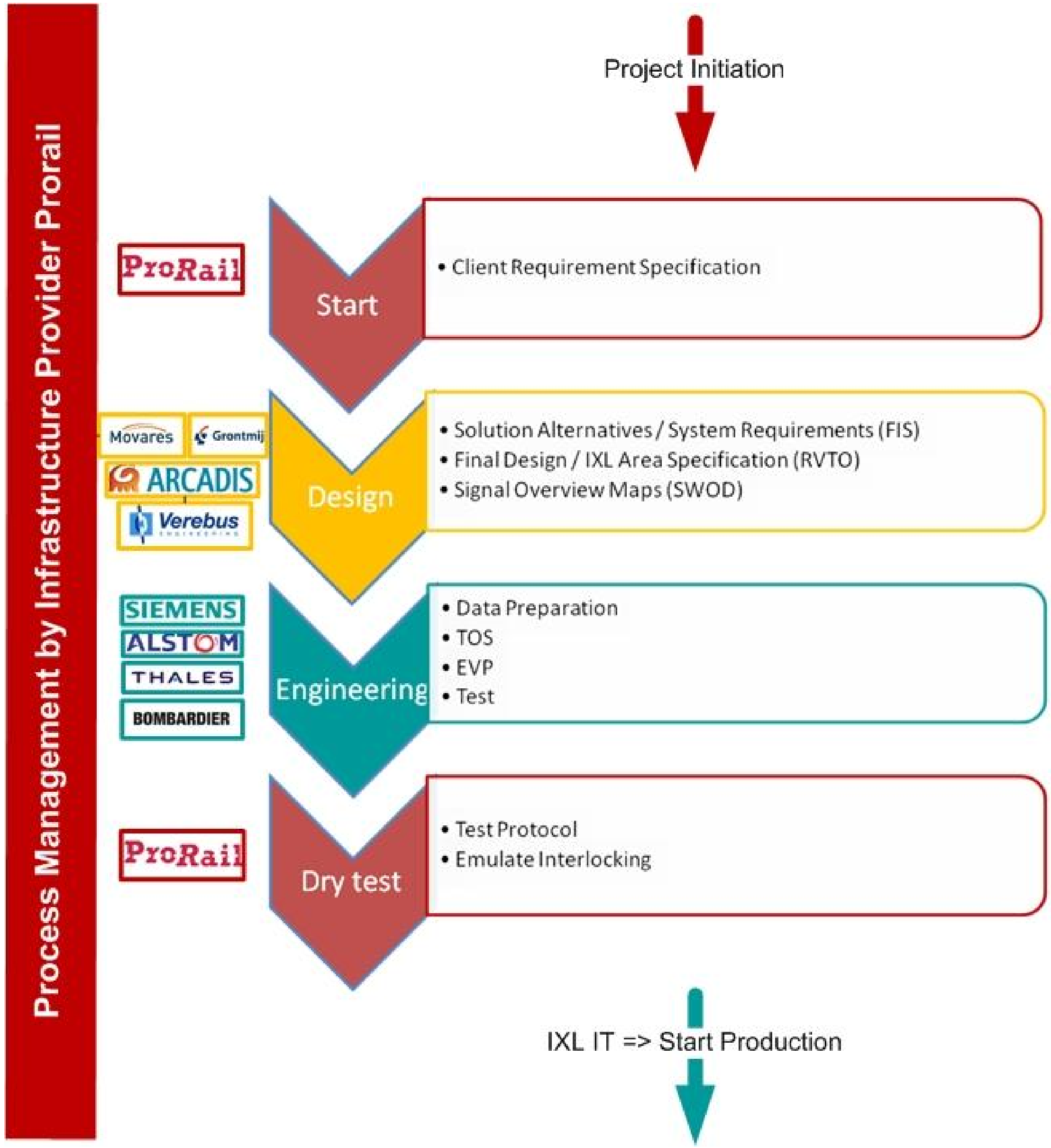
Ensure data quality

- Prerequisite for cost reduction
- Reduces human error
- Complete and precise
- Reliable, i.e. is the probability of erroneous data sufficiently low

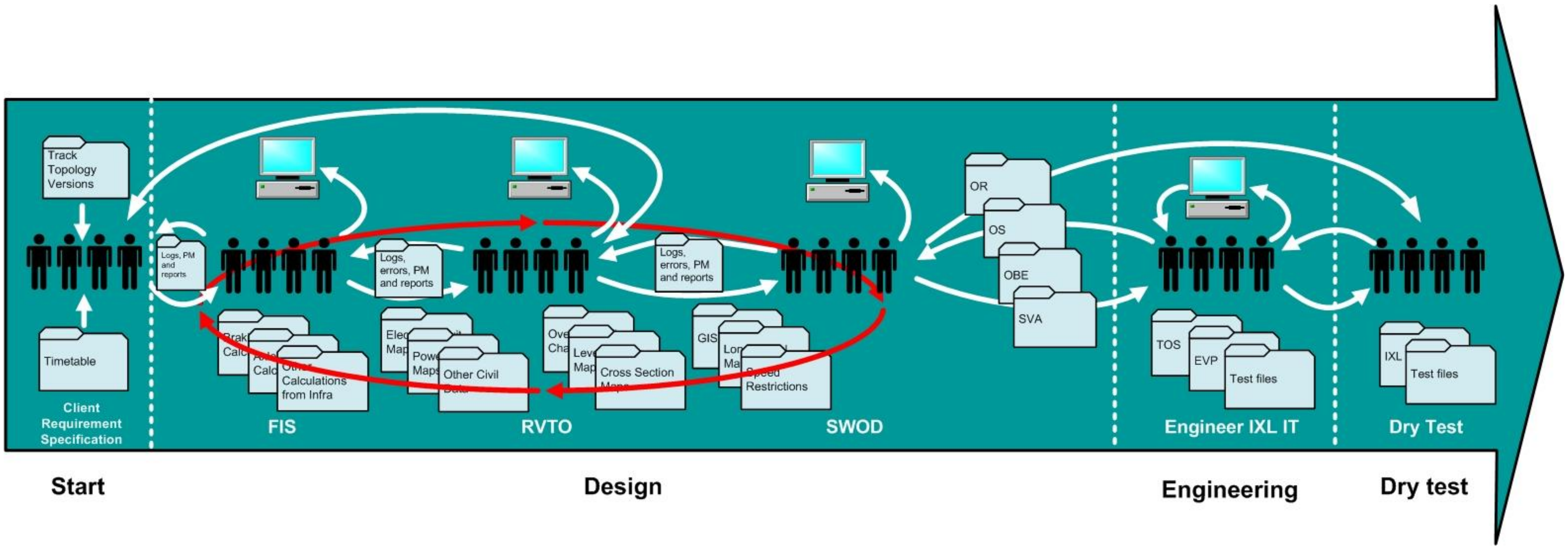
Interlocking Process (V-)Chain



Interlocking Engineering Design



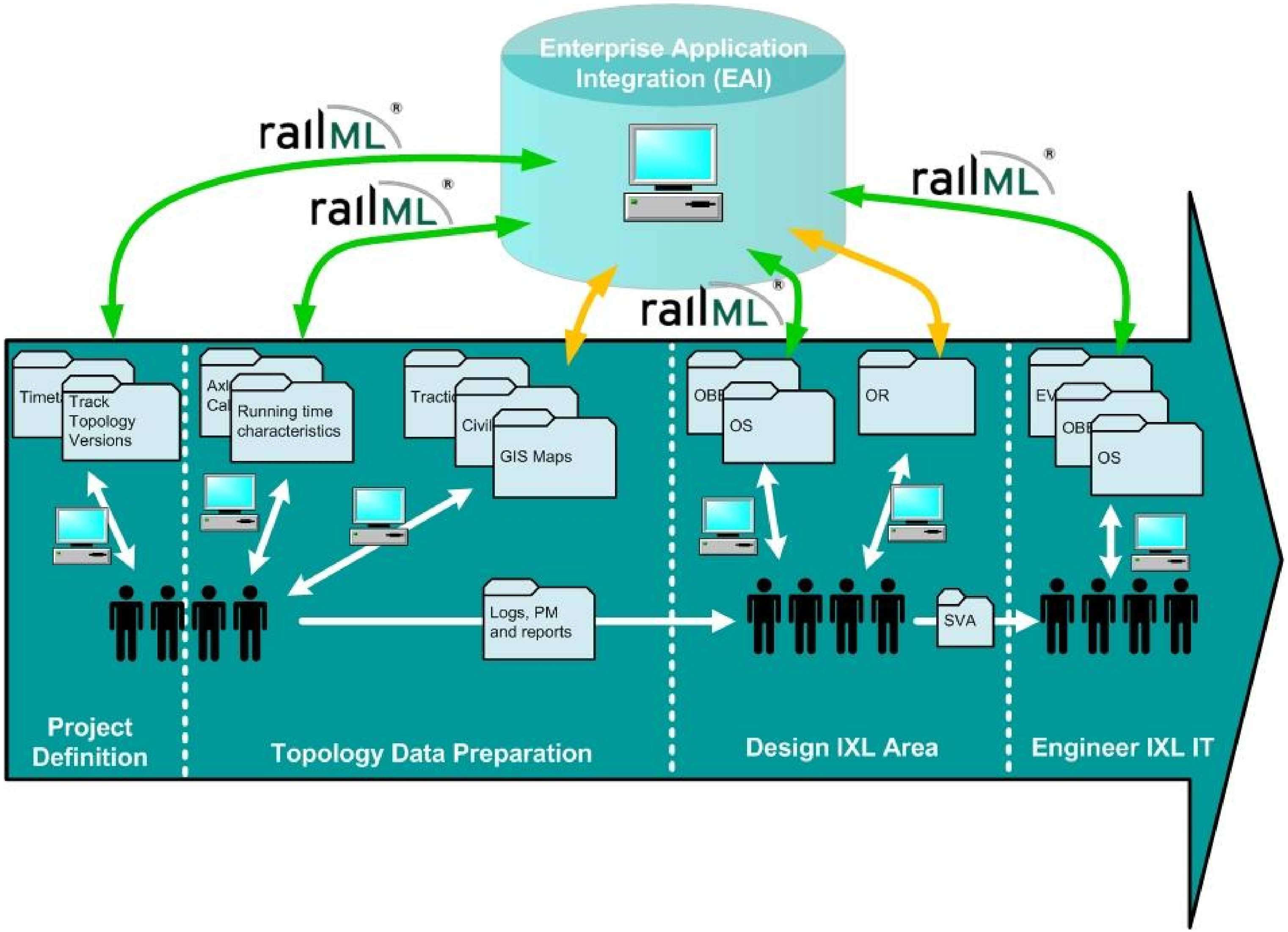
Interlocking Engineering Design
The Status Quo



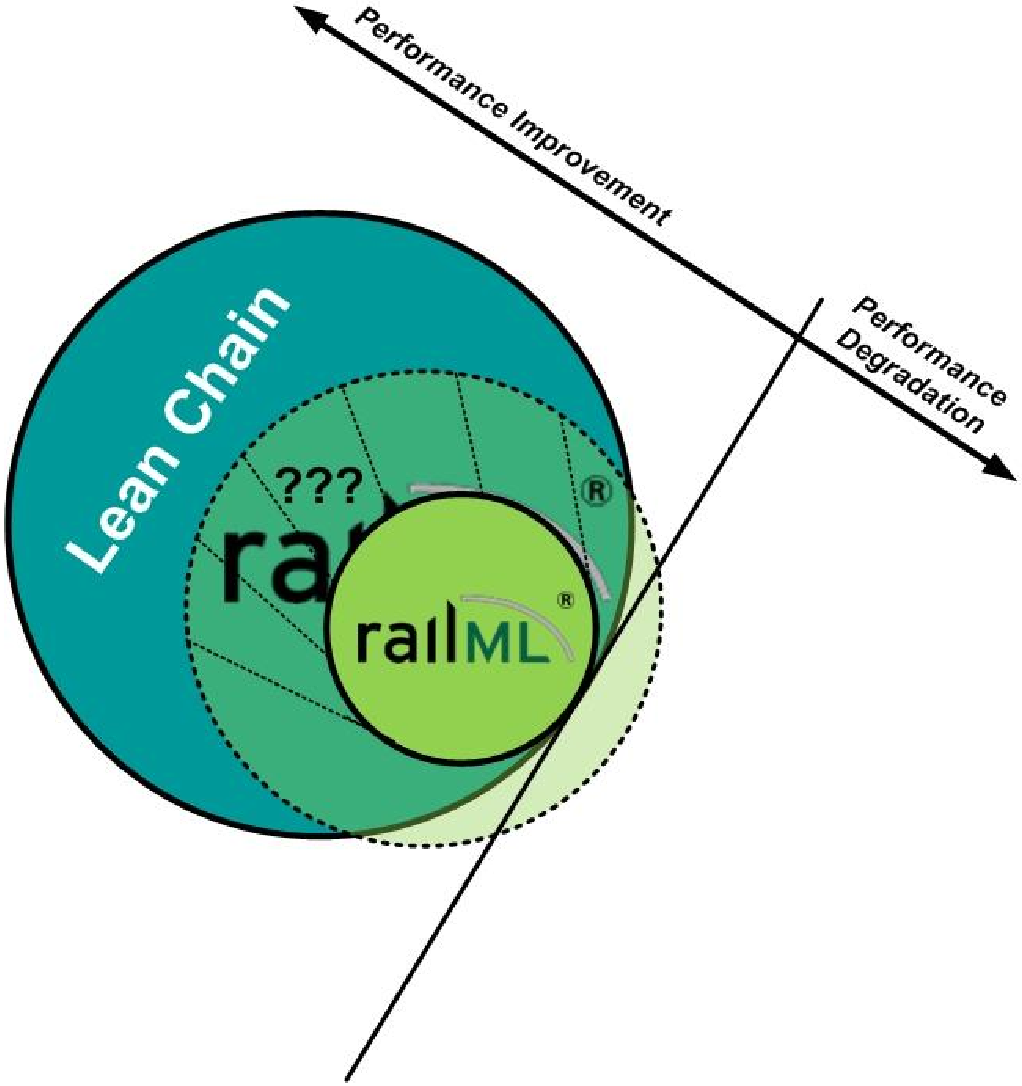
Interlocking Engineering Design

- Design
  - Fluctuating requirements
  - Many and ambiguous processes
  - Error prone, i.e. many parties and manual transfer
  - Design from scratch
- Engineering
  - Considerable fixed costs
  - Input variety => niches and many tests
  - Data translation imposes challenges
- General: Time is cost driver!

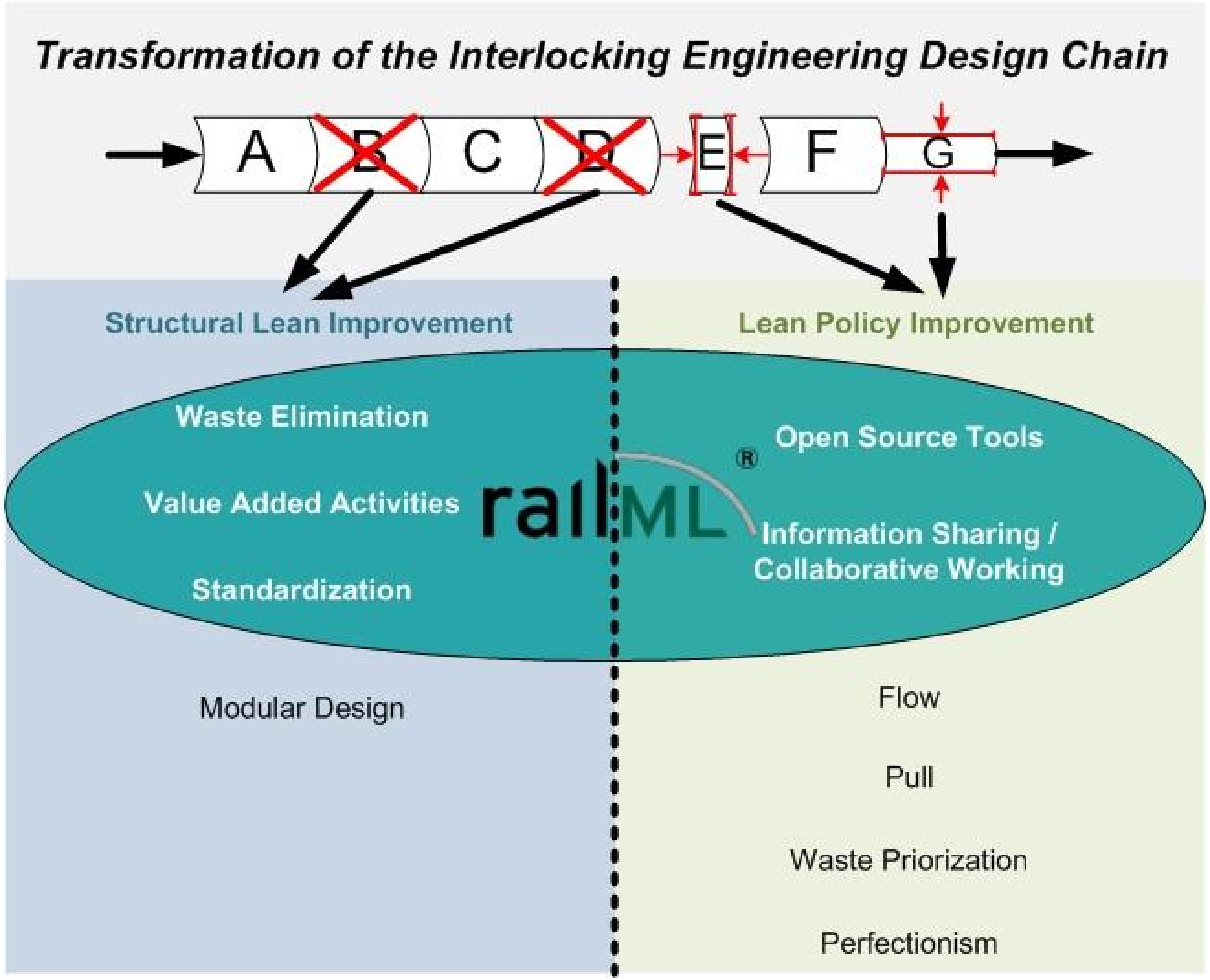
railML's Transformation of Chain



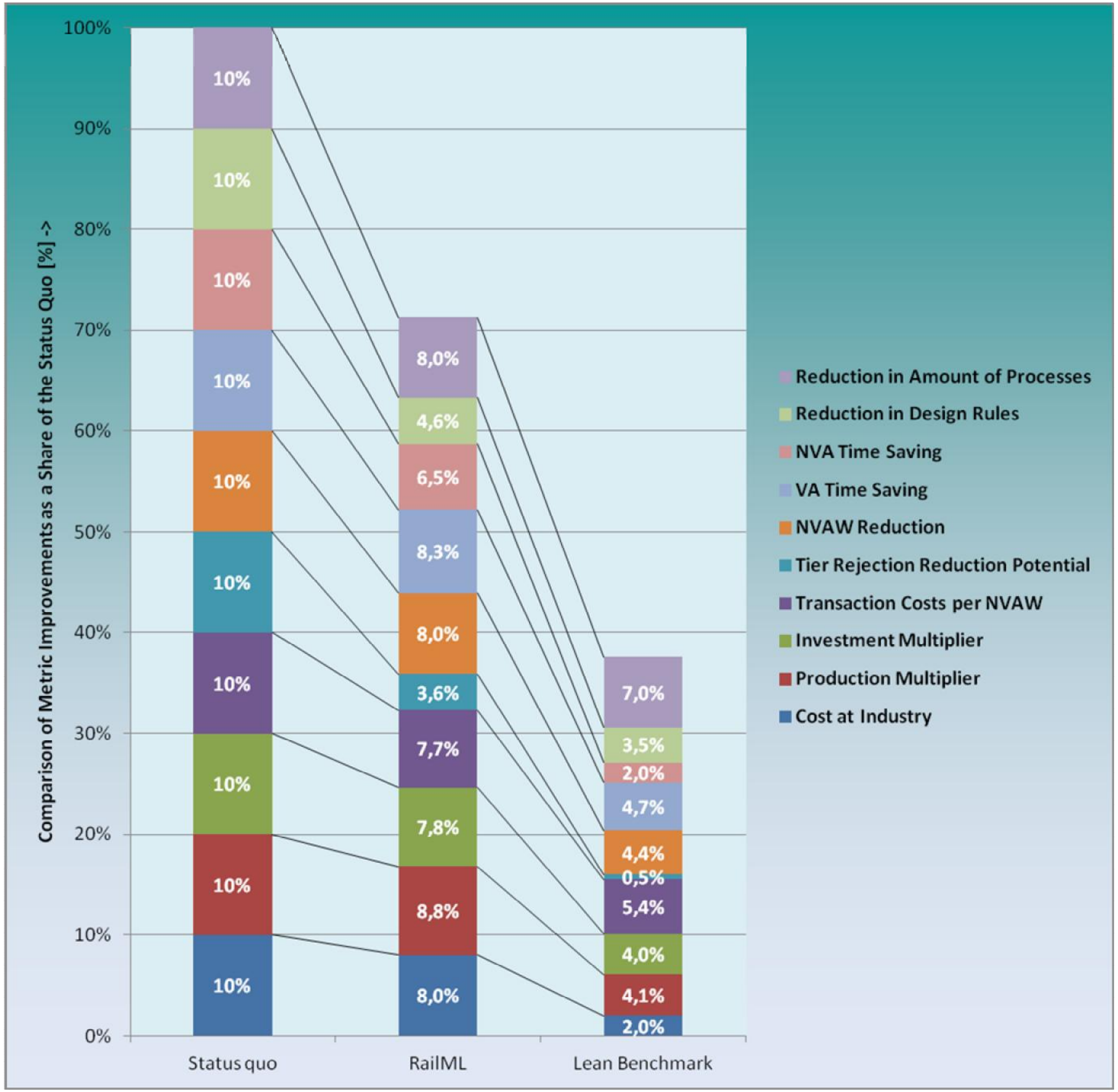
Performance Measurement: Lean



- Benchmark needed
- Womack and Jones (1996) introduce Lean Production
- Lean Engineering Design unexplored
- Transformation strategies plausible


Transformation of the Interlocking Engineering Design Chain
A
B
C
D
E
F
G
Structural Lean Improvement
Lean Policy Improvement
Waste Elimination
Value Added Activities
Standardization
railML®
Open Source Tools
Information Sharing / Collaborative Working
Modular Design
Flow
Pull
Waste Priorization
Perfectionism

Performance Improvement



Conclusions for railML in Interlocking Engineering Design (2)

- railML especially successful to reduce:
  - Complexity
  - Non-value added time
  - Validation cycles

- railML lacks improvement potential on:
  - Cost
  - Productivity
  - Risk
  - Non-value added work

Success factors

- Single database
- Non black box - visualization
- Safety case
- Integration with other signaling systems and engineering parties
- railML v2.2 progress
- Best practice
- Transformation strategy