ON-TIME National Workshop Sweden, 16 October 2014
Innovations in Timetable planning and Traffic control

[Optimal Networks for Train Integration Management across Europe]
Collaborative Project
7th Framework Programme

ON-TIME Real-Time Traffic Management of Minor Perturbations
Rob M.P. Goverde\textsuperscript{1} & Thomas Albrecht
\textsuperscript{1}Delft University of Technology, The Netherlands

r.m.p.goverde@tudelft.nl
What does WP4 do?

- Real-time perturbation management in case of small delays
- Control measures
  - Re-order trains
  - Re-route trains
  - Re-time trains
  - Cancel or add non-commercial stops (operational stops)

- *No interaction* with RU necessary
- Mainly *automatic* decisions
Questions from practice

• These algorithms cover just particular aspects of optimisation.

• These algorithms can’t work in real time.

• Are these algorithms really efficient in practice? (Are they able to provide additional capacity, less delays etc.?)

• These algorithms require data which is not available.

• What will our dispatchers say if they are confronted with these algorithms?
Questions from practice

- These algorithms cover just particular aspects of optimisation. → Modular system design (1)

- These algorithms can’t work in real time. → Real-time environment for data exchange (2)

- Are these algorithms really efficient in practice? (Are they able to provide additional capacity, less delays etc.?) → Evaluation in simulations (3)

- These algorithms require data which is not available. → Standardized data model (4)

- What will our dispatchers say if they are confronted with these algorithms? → Concept for human machine interaction (5)
Modular design

- Different modules from different universities
- Different aspects of railway traffic control

- Traffic State Monitoring
- Route Setting (Automatic Execution)
- Traffic State Prediction
- ROMA Core
Questions from practice

- These algorithms cover just particular aspects of optimisation. → **Modular system design (1)**
- These algorithms can’t work in real time. → **Real-time environment for data exchange (2)**
- Are these algorithms really efficient in practice? (Are they able to provide additional capacity, less delays etc.?) → **Evaluation in simulations (3)**
- These algorithms require data which is not available. → **Standardized data model (4)**
- What will our dispatchers say if they are confronted with these algorithms? → **Concept for human machine interaction (5)**
Data exchange architecture

- Service-Oriented-Architecture
- Publish/Subscribe Services

![Diagram showing data exchange architecture](image)
Interaction WP4

Architecture  Monitoring  Prediction  Conflict Resolution

- TDSectionOccupation
- CurrentTrafficState

FP7 - ON-TIME Collaborative Project

ON-TIME National Workshop Sweden, Borlänge, 16 October 2014  Pag. 8
Interaction WP4

Architecture  Monitoring  Prediction  Conflict Resolution

TDSectionOccupation

CurrentTrafficState

FutureTrafficState

FutureTrafficState
CDR

- Different models, classes of algorithms, and programming languages
- Implemented CDR approaches
  - **ROMA**: Alternative Graph Approach, B&B Solution (TU Delft)
  - **RECIFE**: MILP (IFSTTAR)
  - **DEJRM**: Evolutionary Algorithm (UoB)
- All algorithms extended and adapted to work with the control loop in real-time using the interfaces defined
- Result: **Real-Time Traffic Plan**
Real-time traffic plan

- Describes microscopically how the traffic shall be executed
- Routing, timing and stopping information

**Routing:**
- which routes will the trains take
- in which order will trains pass over sections

**Timing:**
- when will a train occupy a certain section

**Stopping:**
- where and when will trains stop
Routing part

Train view

<table>
<thead>
<tr>
<th>Train 1001</th>
<th>Train 801</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route</td>
<td>Section</td>
</tr>
<tr>
<td>O-A</td>
<td>TS1</td>
</tr>
<tr>
<td>A-C</td>
<td>TS2</td>
</tr>
<tr>
<td>TS4</td>
<td></td>
</tr>
<tr>
<td>C-D</td>
<td>TS5</td>
</tr>
<tr>
<td>D-E</td>
<td>TS6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Route</th>
<th>Sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>O-A</td>
<td>TS1</td>
</tr>
<tr>
<td>A-B</td>
<td>TS2</td>
</tr>
<tr>
<td>B-D</td>
<td>TS5</td>
</tr>
<tr>
<td>D-E</td>
<td>TS6</td>
</tr>
</tbody>
</table>

RTTP = Result of conflict detection and resolution function

Infrastructure view

<table>
<thead>
<tr>
<th></th>
<th>TS1</th>
<th>TS2</th>
<th>TS3</th>
<th>TS4</th>
<th>TS5</th>
<th>TS6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>1001</td>
<td>801</td>
<td>1001</td>
<td>801</td>
<td>801</td>
<td></td>
</tr>
<tr>
<td>801</td>
<td>801</td>
<td></td>
<td>1001</td>
<td>1001</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

OR/RTTP = Result of conflict detection and resolution function
Interaction WP4

Architecture  Execution  Prediction  Conflict Resolution  Envelope

HMI

RTTP
Questions from practice

- These algorithms cover just particular aspects of optimisation. → **Modular system design (1)**
- These algorithms can’t work in real time. → **Real-time environment for data exchange (2)**
- Are these algorithms really efficient in practice? (Are they able to provide additional capacity, less delays etc.?) → **Evaluation in simulations (3)**
- These algorithms require data which is not available. → **Standardized data model (4)**
- What will our dispatchers say if they are confronted with these algorithms? → **Concept for human machine interaction (5)**
Simulator as railway system

- HERMES Simulator
- DataProvider
  - Traffic State Monitoring
  - Route Setting (Automatic Execution)
  - Traffic State Prediction
  - ROMA Core
Results Iron-Ore Line

• Scenario 1: Entrance delay train 9904 of 40 min

**Without ON-TIME**

- Deviation area $[\text{h}^2]$ 4:15
- Maximum delay [h:mm] 1:45
- Settling time [h:mm] 3:50

**With ON-TIME (ROMA)**

- Deviation area $[\text{h}^2]$ 3:45
- Maximum delay [h:mm] 1:15
- Settling time [h:mm] 3:45
Results Iron-Ore Line

- Scenario 2: Speed restriction 20 km/h Rensjön-Bergfors

Without ON-TIME

- Deviation area [h²]: 11:43
- Maximum delay [h:mm]: 4:17
- Settling time [h:mm]: 6:19

With ON-TIME (ROMA)

- Deviation area [h²]: 9:42
- Maximum delay [h:mm]: 2:50
- Settling time [h:mm]: 6:25
Questions from practice

- These algorithms cover just particular aspects of optimisation. → **Modular system design (1)**
- These algorithms can't work in real time. → **Real-time environment for data exchange (2)**
- Are these algorithms really efficient in practice? (Are they able to provide additional capacity, less delays etc.?) → **Evaluation in simulations (3)**
- These algorithms require data which is not available. → **Standardized data model (4)**
- What will our dispatchers say if they are confronted with these algorithms? → **Concept for human machine interaction (5)**
Standardized data

- **railML 2.2**
  - Timetable
  - Infrastructure
  - Rolling stock
  - Interlocking (**NEW!**)  

- Microscopic data
- Example for Sweden
Questions from practice

• These algorithms cover just particular aspects of optimisation. → Modular system design (1)
• These algorithms can’t work in real time. → Real-time environment for data exchange (2)
• Are these algorithms really efficient in practice? (Are they able to provide additional capacity, less delays etc.?) → Evaluation in simulations (3)
• These algorithms require data which is not available. → Standardized data model (4)
• What will our dispatchers say if they are confronted with these algorithms? → Concept for human machine interaction (5)
Human Machine Integration

- Ansaldo HMI
Train describer + Train graph
Example: shift overtaking location

Human constraints to optimization

- Graph showing time on the x-axis and position on the y-axis.
- Various lines indicating different scenarios.
- Dashed lines indicate possible shift overtaking locations.
- Text: Human constraints to optimization.
Open issues

- Including interaction with human
- Including interaction with driving optimization
- More stochasticity in the simulation
- Comparisons with reality

- More detailed data models
  (to see how much is still to gain)
Conclusions/ Lessons learnt

• A modular automatic real-time traffic management of small perturbations is feasible

• Modules, tools and experience are available for next steps of test and integration with real rail system

• Testing requires automation, but automation requires „100% solutions“

• Handling with simulators can be as difficult as with real-world railway systems

• Data modelling/consistency and system integration tests were the most underestimated and time-consuming aspects of the project