Timed CSP in Railway Domain

Markus Roggenbach (Swansea, Wales)

with contributions by
Marc Dragon (Swansea, Wales), Yoshinao Isobe (AIST, Japan)

Vino, July 2011
Overview

What it is all about – The SafeCap project
Safety – CSP
Towards capacity – Timed CSP
What it is all about?
UK Government White Paper ’07

- “Rail’s biggest contribution to tackling global warming comes from increasing its capacity.”
- “Reliability and capacity are amongst top passengers’ concerns.”
- “Increasing capacity is the most urgent investment need.”
“The public accounts committee stated that trains in England and Wales will be more crowded and that the attempts to increase capacity have failed ...”
SafeCap

Overcoming railway capacity challenges without undermining rail network safety.

M.Roggenbach: Timed CSP in Railway Domain; Vino, July 2011
SafeCap

Overcoming railway capacity challenges without undermining rail network safety.

**Funded by:** Railway Safety and Standards Board / EPSRC.

**Started:** 2/2011.

**Partners:** Swansea, Newcastle, Invensys, AIST (Japan).

**Aim:** To identify and validate design patterns for improving capacity by altering route design, track layout, signalling principles and driving rules.
A concrete example

Old control table

<table>
<thead>
<tr>
<th>S14</th>
<th>Clear</th>
<th>Lit</th>
<th>Point</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proceed</td>
<td>AI, AJ, AK, AL</td>
<td>S16</td>
<td>P921R</td>
<td>AH 28sec</td>
</tr>
</tbody>
</table>

New control table

<table>
<thead>
<tr>
<th>S14</th>
<th>Clear</th>
<th>Lit</th>
<th>Point</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proceed</td>
<td>AI, AJ, AK</td>
<td>S16</td>
<td>P921R</td>
<td>n/a</td>
</tr>
</tbody>
</table>

M. Roggenbach: Timed CSP in Railway Domain; Vino, July 2011
Safety – CSP
CSP verification approach

Model:
1. Safety as a CSP process $Safe$.
2. Railway as a CSP process $System$.

Prove:

$Safe$ trace-refines to $System$.

$(\text{traces}(System) \subseteq \text{traces}(Safe))$
CSP verification approach

Model:
1. Safety as a CSP process $Safe$.
2. Railway as a CSP process $System$.

Prove:

$Safe$ trace-refines to $System$.

$(\text{traces}(System) \subseteq \text{traces}(Safe))$

Winter '02: CSP modelling for safety in railway domain.
A toy problem

Signal 1 – protects Track 2.
Signal 2 – protects Track 3.
Signal 3 – protects Track 4 and Track 1.
All moves in \textit{Safe}
All moves in \textit{System}
M. Roggenbach: Timed CSP in Railway Domain; Vino, July 2011
CSP-Prover (Swansea/Tsukuba)

```
(*----------------------------------*
  InitialSafeMove <=T TrainAndSignals
(*----------------------------------*)

lemma "InitialSafeMove <=T TrainAndSignals"
apply (simp add: InitialSafeMove_def TrainAndSignals_def)
apply (rule cspT_fp_induct_left[of _ "Spc_to_Imp"])
apply (simp_all)
apply (simp)

ISO8-----XEmacs: Train.thy (Isar script XS:isar/s Font Size)

proof (prove): step 0

goal (1 subgoal):
  1. InitialSafeMove <=T TrainAndSignals
```
Towards capacity – Timed CSP
Timed CSP

Time: Real number $\geq 0$.
Newtonian time concept – single conceptual global clock.

Conservatively extends CSP by three primitives:
- $a@u \rightarrow P(u)$ – Time of an action.
- $P \triangleright^d Q$ – Timed timeout.
- $P \triangle_e Q$ – Timed interrupt.

Syntactic sugar, e.g.
- $Wait d = Stop \triangleright^d Skip$. 

M.Roggenbach: Timed CSP in Railway Domain; Vino, July 2011
Theorem: Timed CSP is closed under rational numbers.
**Theorem:** Timed CSP is closed under rational numbers.

**Proof (Sketch):**

\[
\begin{align*}
P \sim_{d'} P' \\
(P \triangleright^d Q) \sim_{d'} (P' \triangleright^{d-d'} Q) \\
0 < d' \leq d.
\end{align*}
\]
Theorem: Timed CSP is closed under rational numbers.

Proof (Sketch):

\[
\frac{P \sim^{d'} P'}{(P \triangleright^d Q) \sim^{d'} (P' \triangleright^{d-d'} Q)} \quad 0 < d' \leq d.
\]

Assume:

(1) \((P \triangleright^d Q)\) has rational times only; (2) \(d'\) is rational.
Theorem: Timed CSP is closed under rational numbers.

Proof (Sketch):

\[
\begin{align*}
P & \xrightarrow{d'} P' \\
(P \triangleright^d Q) & \xrightarrow{d'} (P' \triangleright^{d-d'} Q) \\
0 < d' & \leq d.
\end{align*}
\]

Assume:

(1) \((P \triangleright^d Q)\) has rational times only; (2) \(d'\) is rational.

Then:

(3) \(d - d'\) is rational (rational arithmetic).

(4) \(P'\) has rational times only (induction hypothesis).

Thus: \((P' \triangleright^{d-d'} Q)\) has rational times only.
Applying the above signalling pattern

“With Overlap”
S1 Clear
Proceed \( t_1, t_2, t_3 \)

changes to

“Without Overlap”
S1 Clear
Proceed \( t_1, t_2 \)
Time assumptions (artificial!)

It takes a train at least

10 time units

• From entering a track – until being fully on the track.

<table>
<thead>
<tr>
<th>Actions</th>
<th>Track AA</th>
<th>Track AB</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ev.10, rr.TrackAA.TrackAB</td>
<td>■</td>
<td>■</td>
<td>10</td>
</tr>
</tbody>
</table>
Time assumptions (artificial!)

It takes a train at least

10 time units

- From entering a track – until being fully on the track.

<table>
<thead>
<tr>
<th>Actions</th>
<th>Track AA</th>
<th>Track AB</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Ev.10, rr.TrackAA.TrackAB</td>
<td></td>
<td></td>
<td>10</td>
</tr>
</tbody>
</table>

50 time units

- From being fully on a track – until entering the next track.

<table>
<thead>
<tr>
<th>Actions</th>
<th>Track AB</th>
<th>Track AC</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Ev.50, ff.TrackAB.TrackAC</td>
<td></td>
<td></td>
<td>50</td>
</tr>
</tbody>
</table>
Time assumptions (artificial!)

It takes a train at least

10 time units

- From entering a track – until being fully on the track.

<table>
<thead>
<tr>
<th>Actions</th>
<th>Track AA</th>
<th>Track AB</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ev.10, rr. TrackAA. TrackAB</td>
<td></td>
<td></td>
<td>10</td>
</tr>
</tbody>
</table>

50 time units

- From being fully on a track – until entering the next track.

<table>
<thead>
<tr>
<th>Actions</th>
<th>Track AB</th>
<th>Track AC</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ev.50, ff. TrackAB. TrackAC</td>
<td></td>
<td></td>
<td>50</td>
</tr>
</tbody>
</table>

Signal changes take no time.

M. Roggenbach: Timed CSP in Railway Domain; Vino, July 2011
SingleTrain (Train1) =
    WAIT 50; Moveff.entry.t1.Train1
    → WAIT 10; Moverr.entry.t1.Train1
    → Gone.1
    → WAIT 50; BehaveTrain(t1.t1.Train1)
TrainAndSignalsOptimised =
   (TwoTrains \ {{ | Gone, Moveff.exit.exit, Moverr.exit.exit | } })
   [ | {{ | Moveff.entry.t1,
       Moveff.t2.t3,
       Moverr.t2.t3,
       Moverr.t4.exit
   | } } | ]

invariant_ok
CSP: (WAIT 10 ; Q(dotTuplet[agent_call(src_span[23]

Global Time: 110

Evolution Transition 10
Evolution Transition [0..10]

Moveff.t1.t2.Train1
tau(timeout)
Evolution Transition 50
tau(hide(Gone,1))
Moverr.entry.t1.Train1
tau(timeout)
Evolution Transition 10
Moveff.entry.t1.Train1
tau(timeout)
Evolution Transition 50
start_cspm_MAIN
Simulation “With overlap”
for shortest time possible

<table>
<thead>
<tr>
<th>Actions</th>
<th>entry</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
<th>exit</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ev.50, ff.entry.t1.Train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Ev.10, rr.entry.t1.Train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>Ev.50, ff.t1.t2.Train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>Ev.10, rr.t1.t2.Train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>110</td>
</tr>
<tr>
<td>Ev.50, ff.t2.t3.Train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>120</td>
</tr>
<tr>
<td>Ev.10, rr.t2.t3.Train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>170</td>
</tr>
<tr>
<td>Ev.50, ff.t3.t4.Train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>180</td>
</tr>
<tr>
<td>Ev.10, rr.t3.t4.Train1, ff.entry.t1.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>230</td>
</tr>
<tr>
<td>Ev.10, rr.entry.t1.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>240</td>
</tr>
<tr>
<td>Ev.40, ff.t4.exit.Train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>250</td>
</tr>
<tr>
<td>Ev.10, rr.t4.exit.Train1, ff.t1.t2.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>290</td>
</tr>
<tr>
<td>Ev.10, rr.t1.t2.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>300</td>
</tr>
<tr>
<td>Ev.50, ff.t2.t3.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>310</td>
</tr>
<tr>
<td>Ev.10, rr.t2.t3.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>360</td>
</tr>
<tr>
<td>Ev.50, ff.t3.t4.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>370</td>
</tr>
<tr>
<td>Ev.10, rr.t3.t4.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>420</td>
</tr>
<tr>
<td>Ev.50, ff.t4.exit.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>430</td>
</tr>
<tr>
<td>Ev.50, ff.t4.exit.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>480</td>
</tr>
<tr>
<td>Ev.50, ff.t4.exit.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>490</td>
</tr>
</tbody>
</table>
Simulation “Without overlap” for shortest time possible

<table>
<thead>
<tr>
<th>Actions</th>
<th>entry</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
<th>exit</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ev.50, ff.entry:t1.T_train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Ev.10, rr.entry:t1.T_train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>Ev.50, ff.t1.t2.T_train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>Ev.10, rr.t1.t2.T_train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>110</td>
</tr>
<tr>
<td>Ev.50, ff.t2.t3.T_train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>120</td>
</tr>
<tr>
<td>Ev.10, rr.t2.t3.T_train1, ff.entry:t1.T_train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>170</td>
</tr>
<tr>
<td>Ev.10, rr.t3.t4.T_train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>180</td>
</tr>
<tr>
<td>Ev.10, rr.t1.t2.T_train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>190</td>
</tr>
<tr>
<td>Ev.40, ff.t3.t4.T_train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>230</td>
</tr>
<tr>
<td>Ev.10, rr.t3.t4.T_train1, ff.t1.t2.T_train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>240</td>
</tr>
<tr>
<td>Ev.10, rr.t1.t2.T_train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>250</td>
</tr>
<tr>
<td>Ev.40, ff.t4.exit.T_train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>290</td>
</tr>
<tr>
<td>Ev.10, rr.t4.exit.T_train1, ff.t2.t3.T_train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>300</td>
</tr>
<tr>
<td>Ev.10, rr.t2.t3.T_train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>310</td>
</tr>
<tr>
<td>Ev.50, ff.t3.t4.T_train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>360</td>
</tr>
<tr>
<td>Ev.10, rr.t3.t4.T_train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>370</td>
</tr>
<tr>
<td>Ev.50, ff.t4.exit.T_train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>420</td>
</tr>
<tr>
<td>Ev.10, rr.t4.exit.T_train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>430</td>
</tr>
</tbody>
</table>
Simulation “Without overlap”
for shortest time possible

<table>
<thead>
<tr>
<th>Actions</th>
<th>entry</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
<th>exit</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ev.50, ff.entry.t1.Train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ev.10, rr.entry.t1.Train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ev.50, ff.t1.t2.Train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ev.10, rr.t1.t2.Train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ev.50, ff.t2.t3.Train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ev.10, rr.t2.t3.Train1, ff.entry.t1.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ev.10, rr.entry.t1.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ev.40, ff.t3.t4.Train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ev.10, rr.t3.t4.Train1, ff.t1.t2.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ev.10, rr.t1.t2.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ev.40, ff.t4.exit.Train1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ev.10, rr.t4.exit.Train1, ff.t2.t3.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ev.10, rr.t2.t3.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ev.50, ff.t3.t4.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ev.10, rr.t3.t4.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ev.50, ff.t4.exit.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ev.10, rr.t4.exit.Train2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Reduction:** 60 time units (minimal passing time for 2 trains).

M.Roggenbach: Timed CSP in Railway Domain; Vino, July 2011
How about safety?

We know from Schneider (2000):

Safe refines to “System_with_time_removed”
implies
Safe refines to System_with_time.
(i.e. System_with_time is safe)
How about safety?

We know from Schneider (2000):

\[
\text{Safe refines to } \text{"System\_with\_time\_removed"} \implies \text{Safe refines to } \text{System\_with\_time}.
\]

(i.e. \text{System\_with\_time} is safe)

FDR shows: Both variants are safe.
Questions on capacity

1. System provides fast service
   i.e., System has a timed trace trace.
   Tools: Simulation; prove “System refinesTo
   timed-Trace-process”.

2. System-1 is at least as good as System-2
   System-1 refinesTo System-2 w.r.t. timed traces
   Tool: refinement proof.
3. System does not provide fast service
   Proof: not (System refinesTo timed-Trace-process) or
   (System || timed-Trace-Process) = STOP.
   Tool: refinement proof.
Conclusion
CSP: (Safety)

- FDR & CSP-Prover prove safety.
Summary

**CSP:** (Safety)
- FDR & CSP-Prover prove safety.

**Timed-CSP:** (Capacity)
- Timed CSP-Simulator (new) allows to explore capacity.
- Safety analysis carries over to timed models.
Future Work

CSP: (Safety)
• “Automatize” CSP-Prover for railway safety proofs.
Future Work

CSP: (Safety)
• “Automatize” CSP-Prover for railway safety proofs.

Timed-CSP: (Capacity)
• Complete Timed CSP-Simulator and Timed CSP-Prover.
Future Work

**CSP**: (Safety)
- “Automatize” CSP-Prover for railway safety proofs.

**Timed-CSP**: (Capacity)
- Complete Timed CSP-Simulator and Timed CSP-Prover.

**Patterns**:
- Explore more patterns in larger settings.

M.Roggenbach: Timed CSP in Railway Domain; Vino, July 2011
Future Work

CSP: (Safety)
- “Automatize” CSP-Prover for railway safety proofs.

Timed-CSP: (Capacity)
- Complete Timed CSP-Simulator and Timed CSP-Prover.

Patterns:
- Explore more patterns in larger settings.

Modelling with Timed CSP and B:
- Explore integrated modelling & verification approaches.

M.Roggenbach: Timed CSP in Railway Domain; Vino, July 2011