An Operational Semantics for Model Checking Long Running Transactions

Hengbiao Yu, Zhenbang Chen, Ji Wang
zbchen@nudt.edu.cn

School of Computer, National University of Defense Technology, Changsha, China
National Laboratory for Parallel and Distributed Processing, Changsha, China

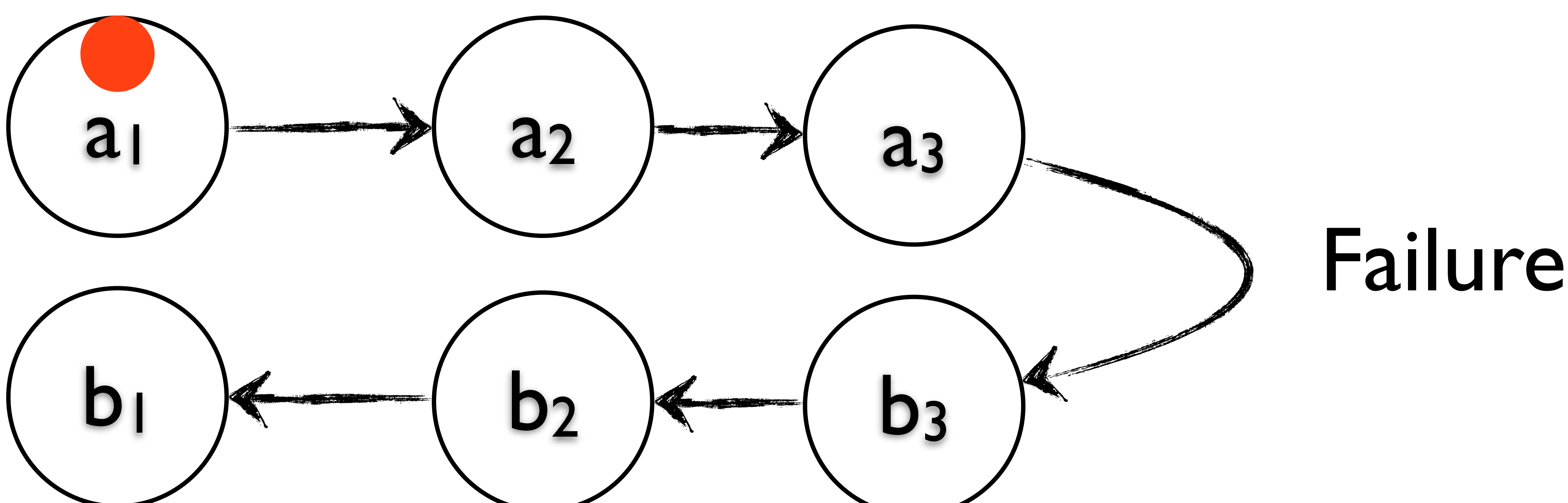
Long Running Transactions

- Database
  - Long-lived transactions
  - ACID transactions
- SAGAS
  - 1987, SIGMOD
- Compensation

Compensation

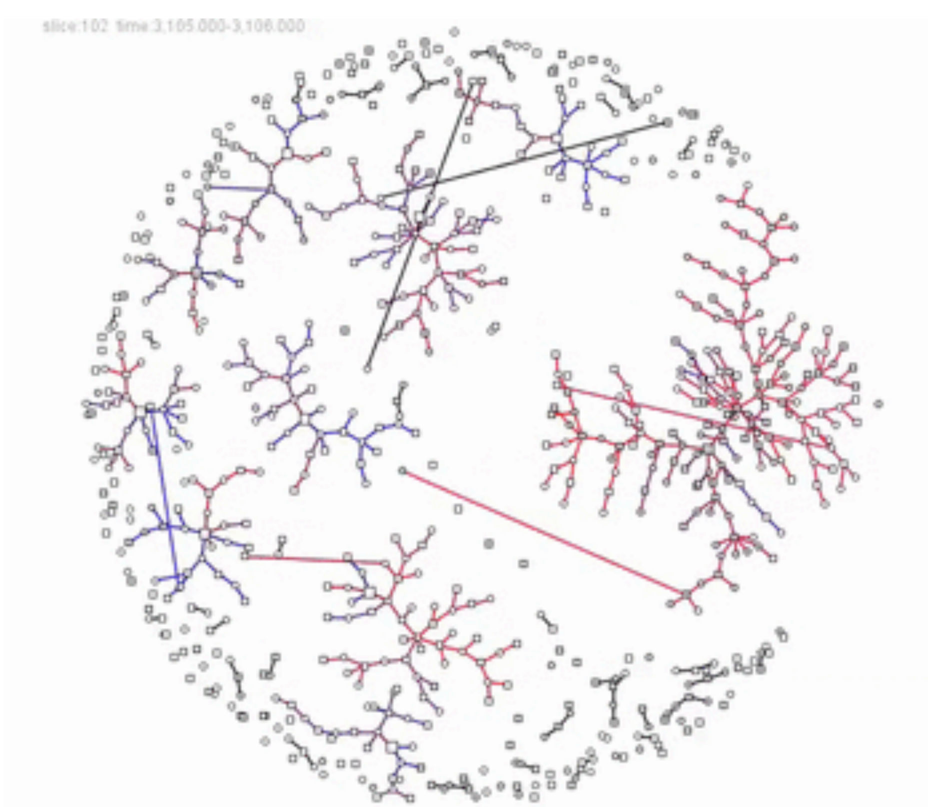
Compensation-based LRTs

- An activity has its compensation activity
- Use compensations in case of failures
- A relaxed atomicity and consistency



Service-Oriented Computing

- Services are world wide distributed
- Coordinate to accomplish a task
  - Highly dynamic, not stable

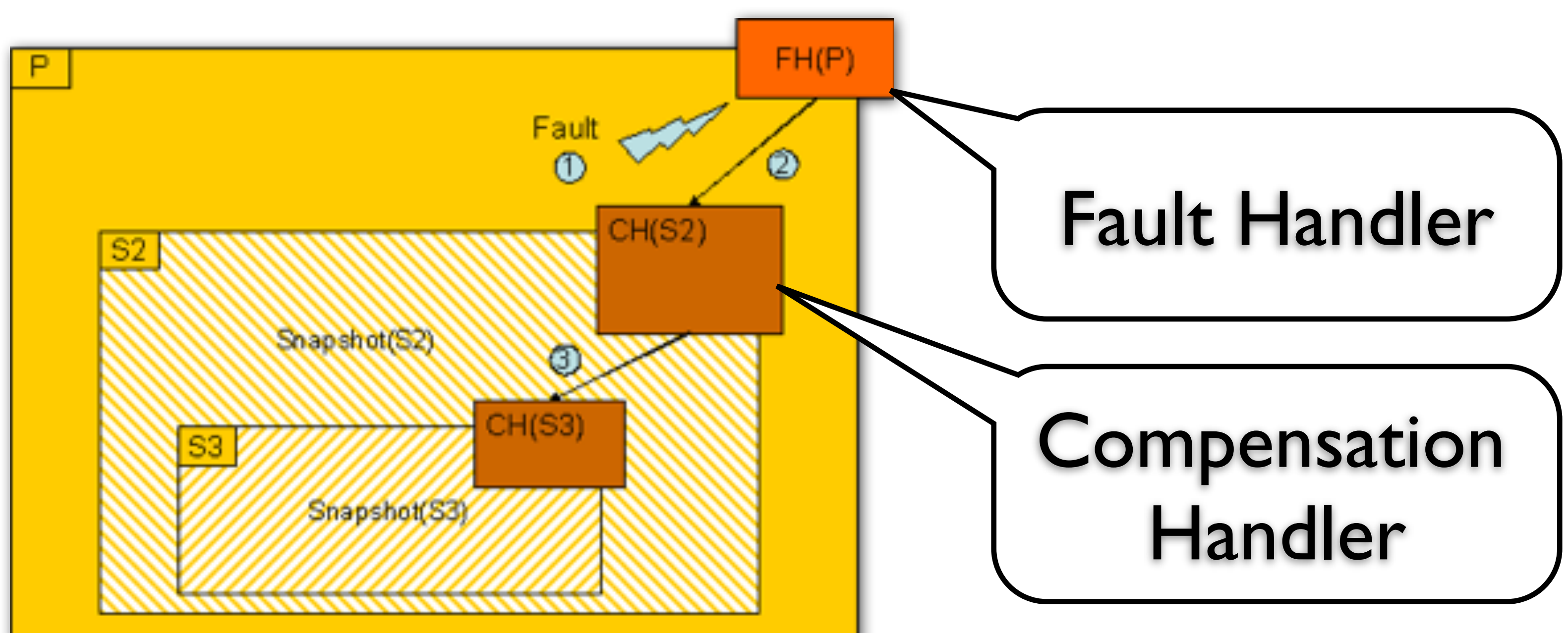


How to ensure consistency
in case of a failure?

Long running transactions

Compensation-based Programming in SOC

- Industrial service orchestration languages
  - Compensation based fault handling
  - Very flexible recovery mechanisms



WS-BPEL 2.0, OASIS Standard, 11 April 2007

Compensation-based Programming in SOC

- Industrial service orchestration languages
  - Compensation based fault handling
  - Very flexible recovery mechanisms
- Formal Languages
  - cCSP, StAC, SAGAs Calculi, *etc.*
  - Theoretical foundations

Compensating CSP

- A CSP variant for modeling LRTs
  - Proposed by Butler *et al.*, 2005
- Two types of processes
  - Standard & Compensable
- Composition operators
  - Choice, sequence, interleaving, *etc*.

Extended cCSP

$$
\begin{aligned}
P \quad &::= \quad a \mid P;P \mid \boxed{P \sqcap P \mid P \Box P \mid P \underset{X}{\parallel} P \mid P \setminus X \mid P[\![R]\!]} \mid P \rhd P \mid [PP] \mid \\
&\qquad \mathbf{skip} \mid \mathbf{throw} \mid \mathbf{yield} \mid \boxed{\mu\, p.F(p)} \\
PP \quad &::= \quad P \div P \mid \boxed{PP;PP \mid PP \sqcap PP \mid PP \Box PP \mid PP \underset{X}{\parallel} PP} \mid PP \boxtimes PP \mid \\
&\qquad \boxed{PP \setminus X \mid PP[\![R]\!]} \mid \mathbf{skipp} \mid \mathbf{throww} \mid \mathbf{yieldd} \mid \boxed{\mu\, pp.FF(pp)}
\end{aligned}
$$

- Enabling the modeling of non-determinism, deadlock and livelock
    - Stable-failures semantics

Zhenbang Chen and Zhiming Liu. An Extended cCSP with Stable Failures Semantics. **7th International Colloquium on Theoretical Aspects of Computing (ICTAC'10)**, LNCS 6255, 2010.

Extended cCSP

$$
\begin{aligned}
P \quad ::= \quad & a \mid P;P \mid \boxed{P \sqcap P \mid P \Box P \mid P \underset{X}{\parallel} P \mid P \setminus X \mid P[\![R]\!]} \mid P \rhd P \mid [PP] \mid \\
& \mathbf{skip} \mid \mathbf{throw} \mid \mathbf{yield} \mid \boxed{\mu\, p.F(p)} \\
PP \quad ::= \quad & P \div P \mid \boxed{PP;PP \mid PP \sqcap PP \mid PP \Box PP \mid PP \underset{X}{\parallel} PP} \mid PP \boxtimes PP \mid \\
& \boxed{PP \setminus X \mid PP[\![R]\!]} \mid \mathbf{skipp} \mid \mathbf{throww} \mid \mathbf{yieldd} \mid \boxed{\mu\, pp.FF(pp)}
\end{aligned}
$$

- Failure-divergence semantics, refinement

Zhenbang Chen, Zhiming Liu and Ji Wang. Failure-Divergence Refinement of Compensating Communicating Processes. **17th International Symposium on Formal Methods (FM'11)**, LNCS 6664, 2011.

Zhenbang Chen, Zhiming Liu and Ji Wang. Failure-Divergence Semantics and Refinement of Long Running Transactions. ***Theoretical Computer Science* (TCS)**, Vol 455, pp:31-65, 2012

Current Issue

- No operational semantics for extended cCSP
    - There is one for cCSP
- No tool support for modeling and verification
    - Animating
    - Model checking
    - ...

What we have done in this paper

- An operational semantics for the extended cCSP
- Model checking problem w.r.t. regular properties
- A prototype tool built on PAT
  - Modeling, animating and model checking

Operational Semantics

$$P \quad ::= \quad a \mid P;P \mid P \sqcap P \mid P \Box P \mid P \underset{X}{\parallel} P \mid P \setminus X \mid P[\![R]\!] \mid P \rhd P \mid [PP] \mid$$
$$\mathbf{skip} \mid \mathbf{throw} \mid \mathbf{yield} \mid \mu\, p.F(p)$$

$$PP \quad ::= \quad P \div P \mid PP;PP \mid PP \sqcap PP \mid PP \Box PP \mid PP \underset{X}{\parallel} PP \mid PP \boxtimes PP \mid$$
$$PP \setminus X \mid PP[\![R]\!] \mid \mathbf{skipp} \mid \mathbf{throww} \mid \mathbf{yieldd} \mid \mu\, pp.FF(pp)$$

Compensation Pair

$$\frac{P \xrightarrow{e} P'}{P \div Q \xrightarrow{e} P' \div Q}\ (e \in \Sigma^{\tau}) \qquad \frac{P \xrightarrow{\checkmark} 0}{P \div Q \xrightarrow{\checkmark} Q} \qquad \frac{P \xrightarrow{\omega} 0}{P \div Q \xrightarrow{\omega} \mathbf{skip}}\ (\omega \in \{!, ?\})$$

Example

$$a_1 \div b_1 \xrightarrow{a_1} \text{skip} \div b_1 \xrightarrow{\surd} b_1$$

Operational Semantics

$$
\begin{array}{lll}
P & ::= & a \mid P;P \mid P \sqcap P \mid P \Box P \mid P \underset{X}{\parallel} P \mid P \setminus X \mid P[\![R]\!] \mid P \rhd P \mid [PP] \mid \\
 & & \mathbf{skip} \mid \mathbf{throw} \mid \mathbf{yield} \mid \mu\, p.F(p) \\
PP & ::= & P \div P \mid PP;PP \mid PP \sqcap PP \mid PP \Box PP \mid PP \underset{X}{\parallel} PP \mid PP \boxtimes PP \mid \\
 & & PP \setminus X \mid PP[\![R]\!] \mid \mathbf{skipp} \mid \mathbf{throww} \mid \mathbf{yieldd} \mid \mu\, pp.FF(pp)
\end{array}
$$

Sequential Composition

$$\frac{PP \xrightarrow{\checkmark} P \wedge QQ \xrightarrow{e} QQ'}{PP;QQ \xrightarrow{e} \langle QQ', P\rangle} \quad (e \in \Sigma^{\tau})$$

$$\frac{PP \xrightarrow{\checkmark} P \wedge QQ \xrightarrow{\omega} Q}{PP;QQ \xrightarrow{\omega} Q;P} \quad (\omega \in \Omega)$$

Example

$$a_1 \div b_1 \; ; a_2 \div b_2$$

$$\Big\downarrow a_1$$

$$\text{skip} \div b_1 \; ; a_2 \div b_2$$

$$\Big\downarrow a_2$$

$$\langle \text{skip} \div b_2 \; , b_1 \rangle$$

Operational Semantics

$$P \quad ::= \quad a \mid P;P \mid P \sqcap P \mid P \Box P \mid P \underset{X}{\parallel} P \mid P \setminus X \mid P[\![R]\!] \mid P \rhd P \mid [PP] \mid \mathbf{skip} \mid \mathbf{throw} \mid \mathbf{yield} \mid \mu\, p.F(p)$$

$$PP \quad ::= \quad P \div P \mid PP;PP \mid PP \sqcap PP \mid PP \Box PP \mid PP \underset{X}{\parallel} PP \mid PP \boxtimes PP \mid PP \setminus X \mid PP[\![R]\!] \mid \mathbf{skipp} \mid \mathbf{throww} \mid \mathbf{yieldd} \mid \mu\, pp.FF(pp)$$

Nested Configuration

$$\frac{QQ \xrightarrow{e} QQ'}{\langle QQ, P\rangle \xrightarrow{e} \langle QQ', P\rangle} \quad (e \in \Sigma^\tau) \qquad \frac{QQ \xrightarrow{\omega} Q}{\langle QQ, P\rangle \xrightarrow{\omega} Q; P} \quad (\omega \in \Omega)$$

Example

$$a_1 \div b_1 \;;\, a_2 \div b_2 \xrightarrow{a_1} \text{skip} \div b_1 \;;\, a_2 \div b_2 \xrightarrow{a_2} \langle \text{skip} \div b_2 \,,\, b_1 \rangle \xrightarrow{\surd} b_2 \;;\, b_1$$

Operational Semantics

$$
\begin{aligned}
P \quad ::= \quad & a \mid P;P \mid P \sqcap P \mid P \Box P \mid P \parallel_X P \mid P \setminus X \mid P[\![R]\!] \mid P \rhd P \mid [PP] \mid \\
& \mathbf{skip} \mid \mathbf{throw} \mid \mathbf{yield} \mid \mu\, p.F(p) \\
PP \quad ::= \quad & P \div P \mid PP;PP \mid PP \sqcap PP \mid PP \Box PP \mid PP \parallel_X PP \mid PP \boxtimes PP \mid \\
& PP \setminus X \mid PP[\![R]\!] \mid \mathbf{skipp} \mid \mathbf{throww} \mid \mathbf{yieldd} \mid \mu\, pp.FF(pp)
\end{aligned}
$$

Configuration Discussion

$$\frac{PP \xrightarrow{\checkmark} P \wedge QQ \xrightarrow{e} QQ'}{PP;QQ \xrightarrow{e} \langle QQ', P\rangle} \quad (e \in \Sigma^{\tau})$$

$a_1 \div b_1 \;;\; (\text{skip} \div b_2 \;;\; a_3 \div b_3) \xrightarrow{a_1} \text{skip} \div b_1 \;;\; (\text{skip} \div b_2 \;;\; a_3 \div b_3)$

$\text{skip} \div b_2 \;;\; a_3 \div b_3 \xrightarrow{a_3} \langle \text{skip} \div b_3 \,,\, b_2 \rangle$ No rule exists in the before semantics

$\text{skip} \div b_1 \;;\; (\text{skip} \div b_2 \;;\; a_3 \div b_3) \xrightarrow{a_3} \langle\langle \text{skip} \div b_3 \,,\, b_2\rangle, b_1\rangle \xrightarrow{\surd} b_3 \;;\; b_2 \;;\; b_1$

Operational Semantics

$$P \quad ::= \quad a \mid P;P \mid P \sqcap P \mid P \Box P \mid P \underset{X}{\parallel} P \mid P \setminus X \mid P[\![R]\!] \mid P \rhd P \mid [PP] \mid$$
$$\mathbf{skip} \mid \mathbf{throw} \mid \mathbf{yield} \mid \mu\, p.F(p)$$

$$PP \quad ::= \quad P \div P \mid PP;PP \mid PP \sqcap PP \mid PP \Box PP \mid PP \underset{X}{\parallel} PP \mid PP \boxtimes PP \mid$$
$$PP \setminus X \mid PP[\![R]\!] \mid \mathbf{skipp} \mid \mathbf{throww} \mid \mathbf{yieldd} \mid \mu\, pp.FF(pp)$$

Parallel Composition

$$\frac{PP \xrightarrow{e} PP' \wedge QQ \xrightarrow{e} QQ'}{PP \underset{X}{\parallel} QQ \xrightarrow{e} PP' \underset{X}{\parallel} QQ'}\ (e \in X) \qquad \frac{PP \xrightarrow{\omega_1} P \wedge QQ \xrightarrow{\omega_2} Q}{PP \underset{X}{\parallel} QQ \xrightarrow{\omega_1 \| \omega_2} P \underset{X}{\parallel} Q}\ (\omega_1, \omega_2 \in \Omega)$$

Example

$$(a_1 \div b_1 \underset{\{a_1,a_2\}}{\|} a_2 \div b_2) \qquad \text{deadlock}$$

Operational Semantics

$$P \quad ::= \quad a \mid P;P \mid P \sqcap P \mid P \Box P \mid P \underset{X}{\parallel} P \mid P \setminus X \mid P[\![R]\!] \mid P \rhd P \mid [PP] \mid \mathbf{skip} \mid \mathbf{throw} \mid \mathbf{yield} \mid \mu\, p.F(p)$$

$$PP \quad ::= \quad P \div P \mid PP;PP \mid PP \sqcap PP \mid PP \Box PP \mid PP \underset{X}{\parallel} PP \mid PP \boxtimes PP \mid PP \setminus X \mid PP[\![R]\!] \mid \mathbf{skipp} \mid \mathbf{throww} \mid \mathbf{yieldd} \mid \mu\, pp.FF(pp)$$

Parallel Composition

$$\frac{PP \xrightarrow{e} PP' \wedge QQ \xrightarrow{e} QQ'}{PP \underset{X}{\parallel} QQ \xrightarrow{e} PP' \underset{X}{\parallel} QQ'}\ (e \in X) \qquad \frac{PP \xrightarrow{\omega_1} P \wedge QQ \xrightarrow{\omega_2} Q}{PP \underset{X}{\parallel} QQ \xrightarrow{\omega_1 \| \omega_2} P \underset{X}{\parallel} Q}\ (\omega_1, \omega_2 \in \Omega)$$

Example

$$(a_1 \div b_1 \underset{\{a_1,a_2\}}{||} a_1 \div b_2) \xrightarrow{a_1} (\mathrm{skip} \div b_1 \underset{\{a_1,a_2\}}{||} \mathrm{skip} \div b_2) \xrightarrow{\surd} b_1 \underset{\{a_1,a_2\}}{||} b_2$$

Correspondence with FD Semantics

- A method for deriving FD semantics from operational semantics
  - Inspired by the method for CSP

Theorem 1
For a standard process *P*, the derived model according to the operational semantics is equal to the FD model of P.

Basic idea of proof: induce the structures of processes

Correspondence with FD Semantics

- A method for deriving FD semantics from operational semantics
  - Inspired by the method for CSP

Theorem 2

For a compensable process *PP*, the derived model according to the operational semantics is equal to the FD model of *PP*.

Model Checking Problem

- General model checking problem with respect to regular properties

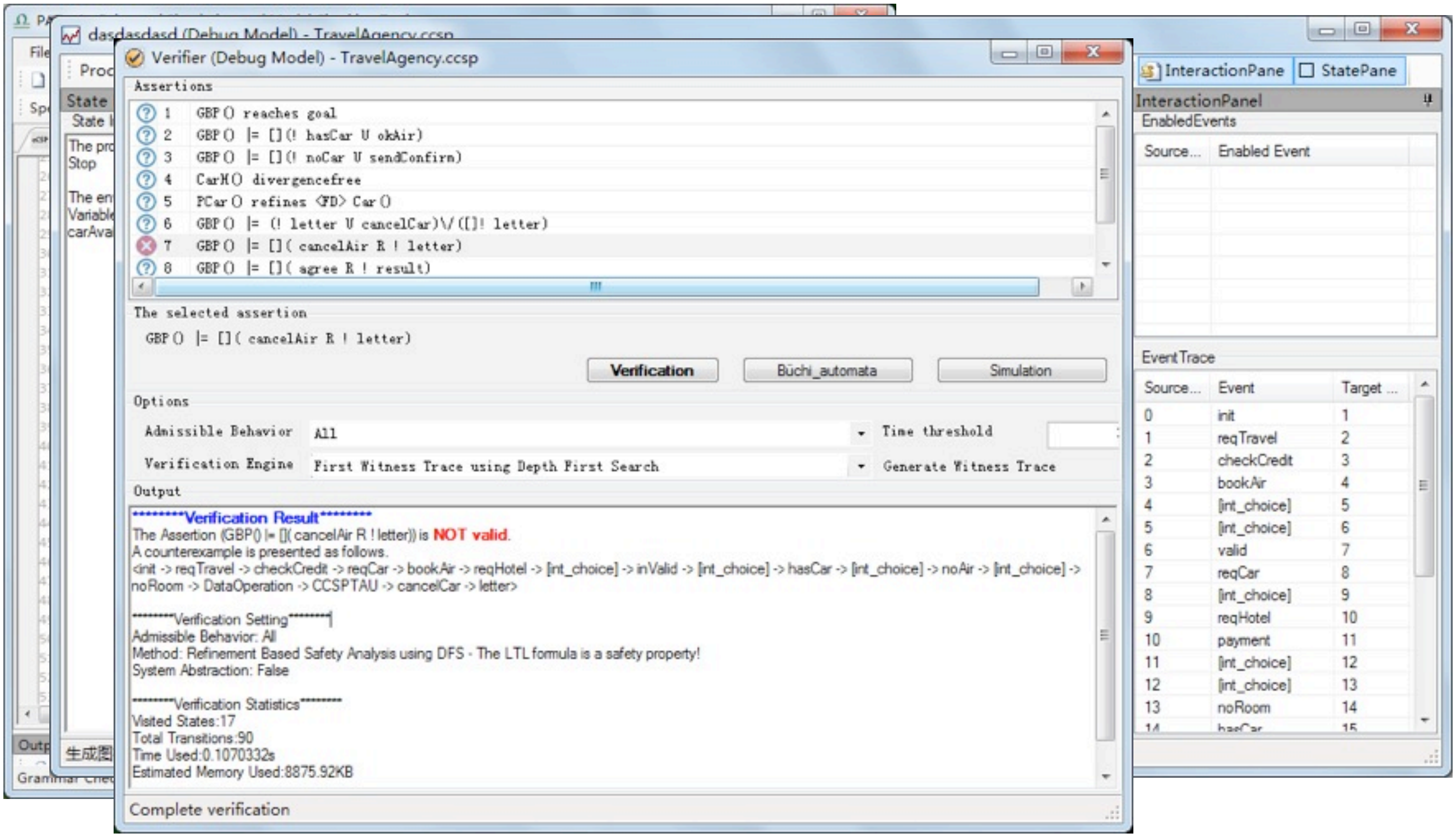
Theorem 3
Given a standard process P of the extended cCSP and an FSM R, the language inclusion problem $L(T(P)) \subseteq L(R)$ is undecidable.

Basic idea of proof: reduction from the halting problem of Minsky 2-counter machine

Tool Implementation

- Built on Process Analysis Toolkit (PAT)
  - Extended cCSP parser
  - Operational semantics Implementation
- Features
  - Modeling environment, animating
  - LTL model checking
  - Refinement checking (only standard process)

Tool Screenshots



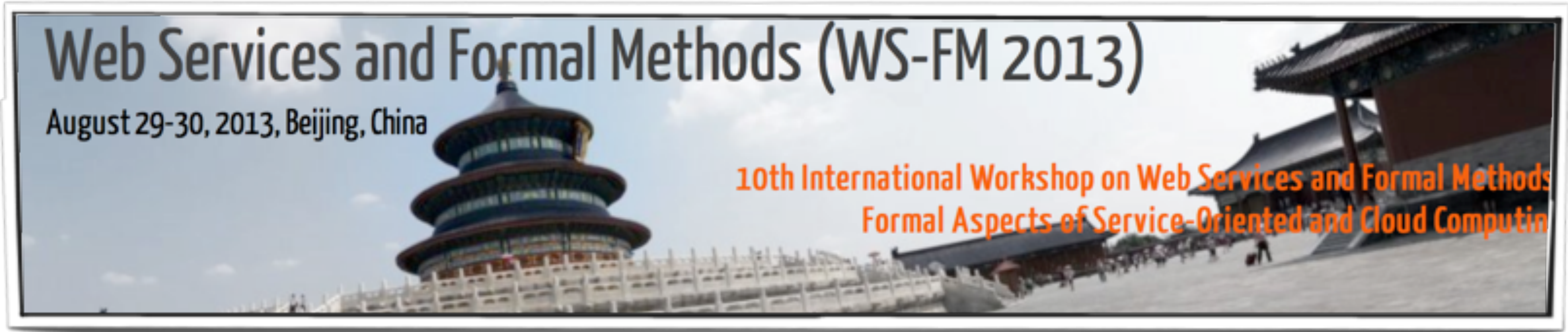
http://rcos.iist.unu.edu/~zbchen/LRT.html

Conclusion

- An operational semantics for extended cCSP
    - Correspondence with FD semantics
- Model checking problem discussion
- A tool for modeling and verifying LRTs
    - Animating, LTL model checking
    - Refinement checking

Next Step

- Refinement checking of compensable processes
- Model checking Algorithms
- Applications
  - Business process verification: BPMN
  - Service orchestration verification: WS-BPEL
  - Recovery-oriented Computing



End

Thank you!