Verifying Business Process Compatibility

Peter Wong, University of Oxford, UK
(Joint work with Jeremy Gibbons)

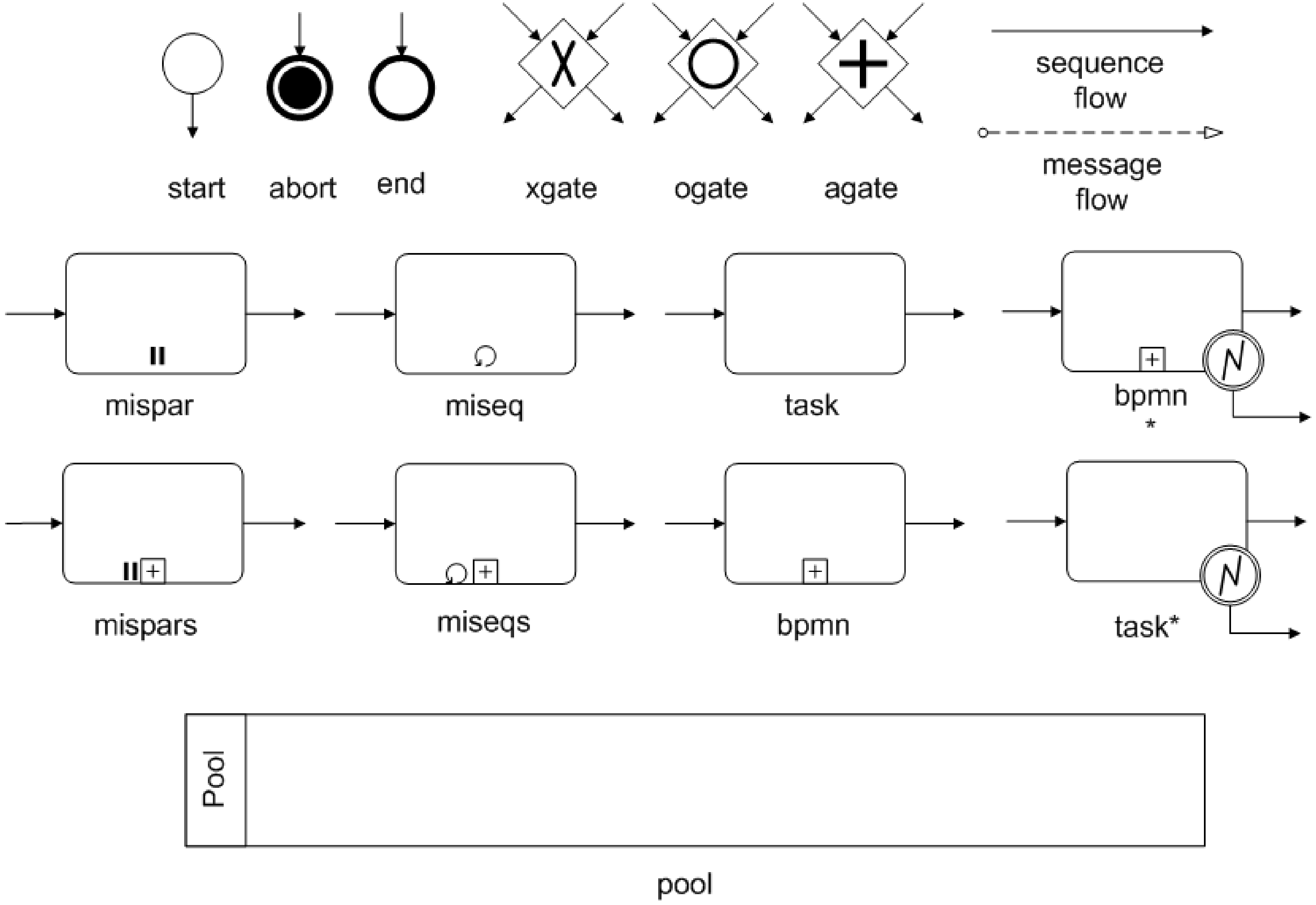
June 2007

Introduction

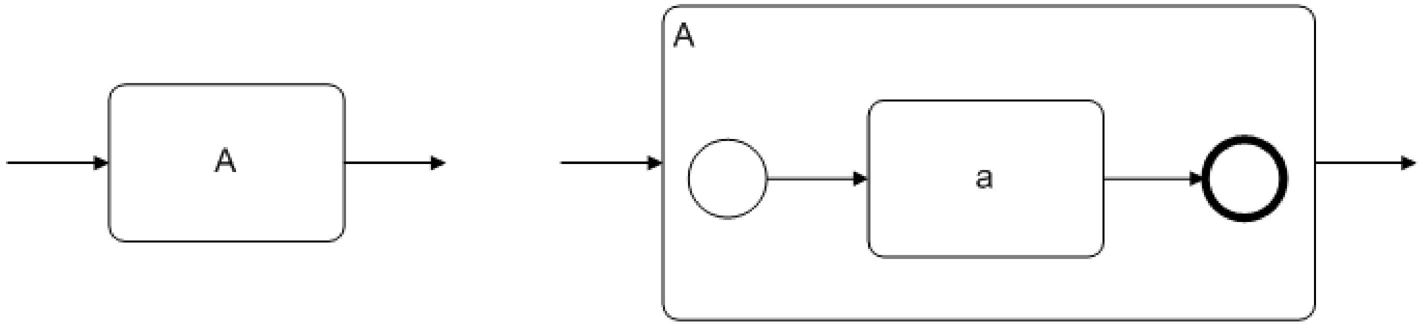
- Business processes may be described as a Business Process Modelling Notation (BPMN) diagram;
- *Local* business process is the composition of service/task components within a local domain;
- *Globally* these business processes may *collaborate via messaging*;
- We formally describe these diagrams semantically using the language of the process algebra Communicating Sequential Processes (CSP);
- Our approach facilitates specification and verification (consistency, compatibility etc.);
- This work is prerequisite to a BPM-based support for clinical trial specification.

see: Example

Business Process Modelling Notation

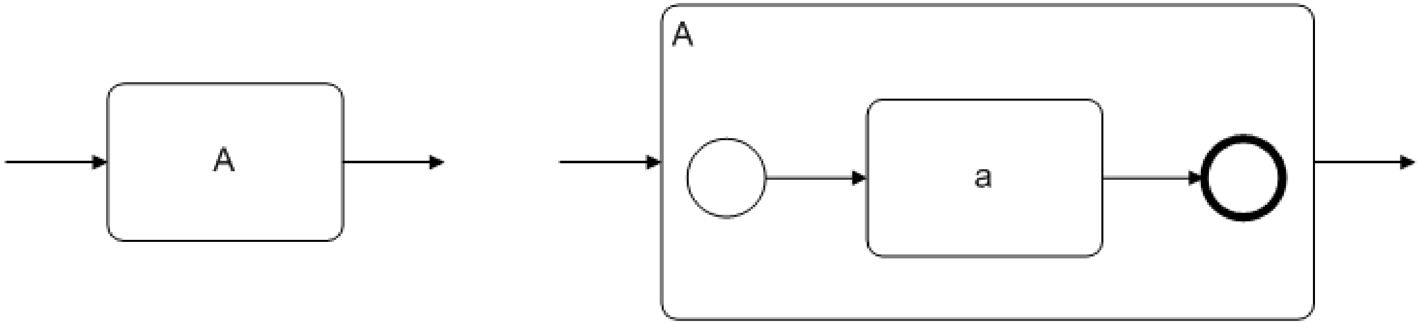


Business Process Modelling Notation (cont.)

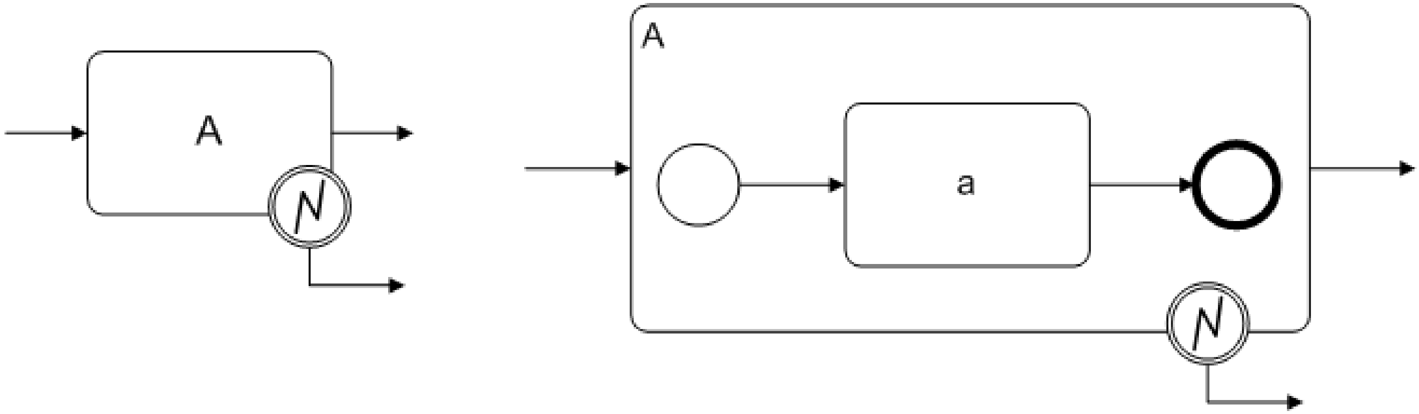


Tasks, subprocesses ...

Business Process Modelling Notation (cont.)

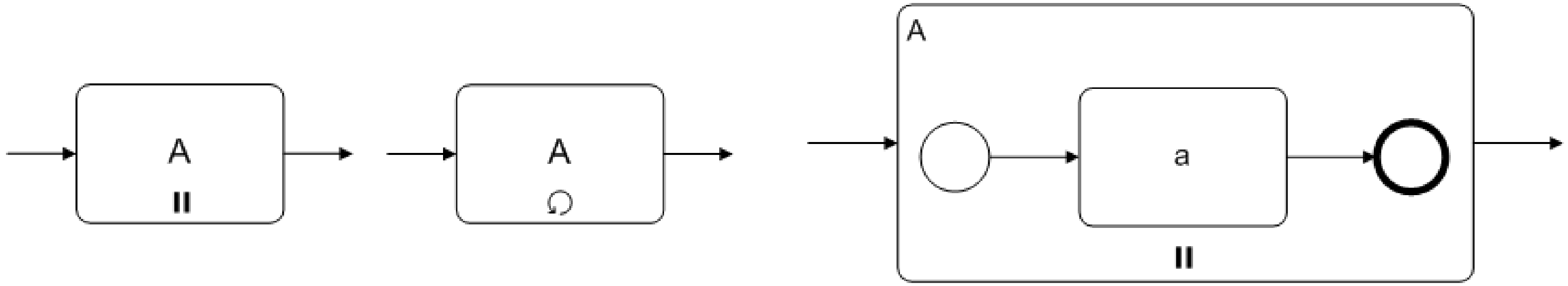


Tasks, subprocesses ...



With exception flows ...

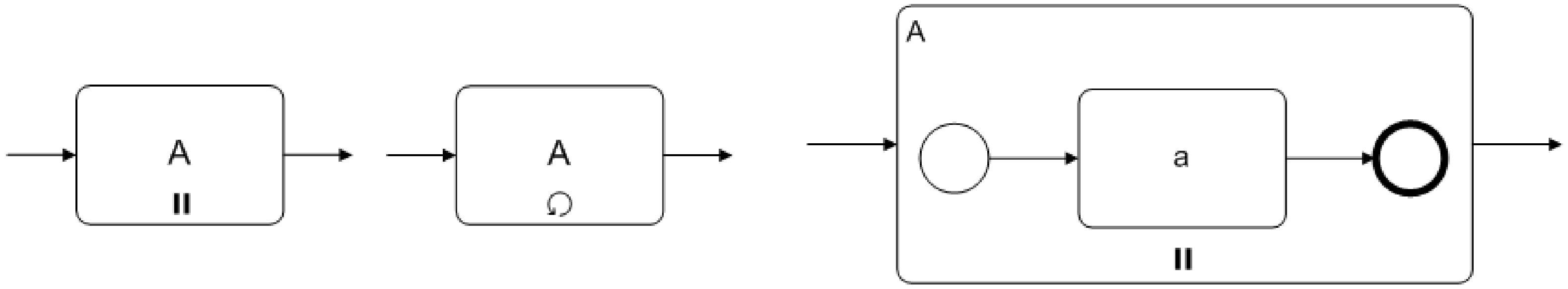
Business Process Modelling Notation (cont.)



Sequential and *parallel* multiple instances of tasks and subprocesses

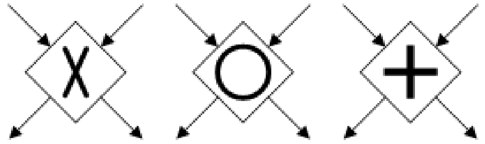
…

Business Process Modelling Notation (cont.)



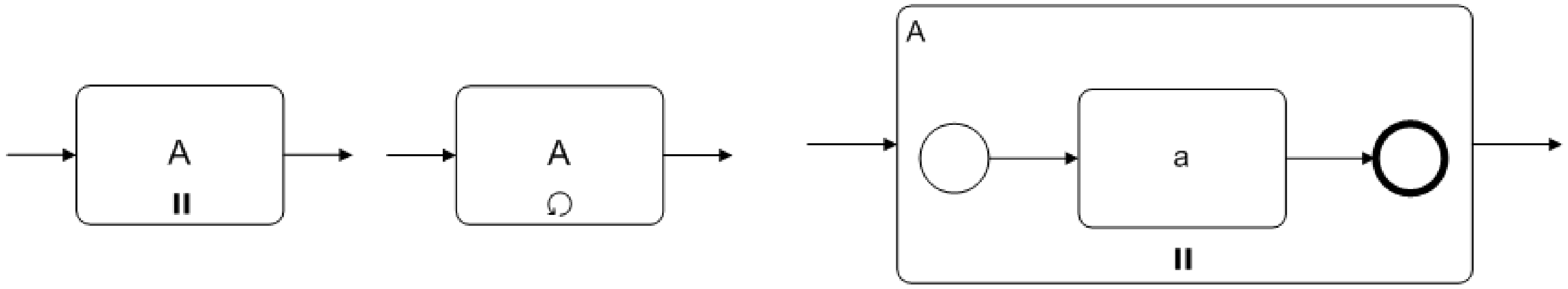
Sequential and *parallel* multiple instances of tasks and subprocesses

...

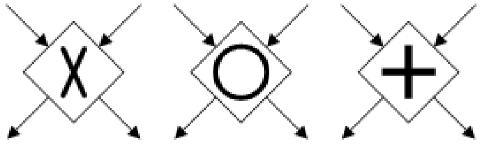


Decision gateways ...

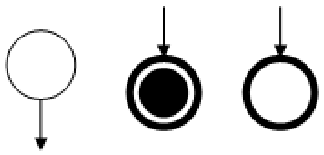
Business Process Modelling Notation (cont.)



Sequential and *parallel* multiple instances of tasks and subprocesses ...

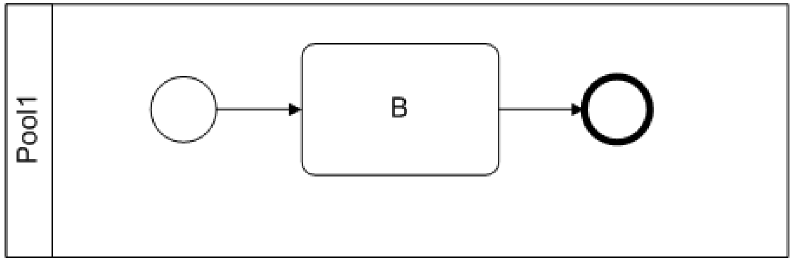


Decision gateways ...



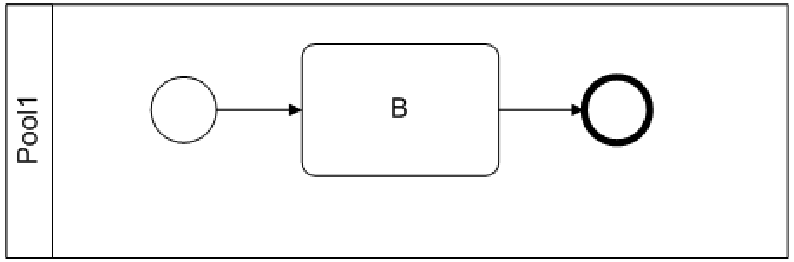
Events ...

Business Process Modelling Notation (cont.)

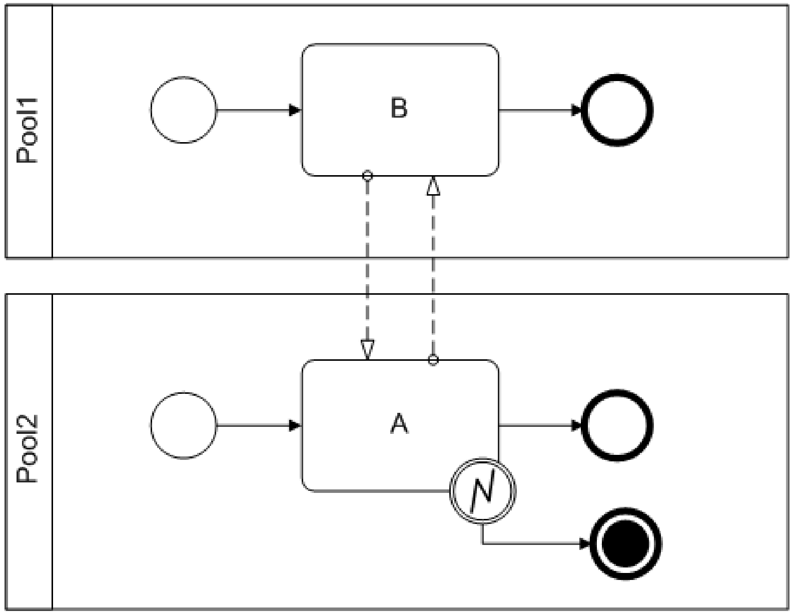


Local composition ...

Business Process Modelling Notation (cont.)



Local composition ...



Global collaboration ...

On formalising BPMN

- Use Z as a syntactic vehicle (schema language, typed set theory);
- Define each BPMN state with the schema type State;

$$\begin{aligned}\mathit{State} \mathrel{\widehat{=}} [&\mathit{in}, \mathit{out}, \mathit{error} : \mathbb{P}\, \mathit{Transition};\ \mathit{type} : \mathit{Type};\\ &\mathit{rec}, \mathit{snd}, \mathit{acc}, \mathit{rep}, \mathit{brk} : \mathbb{P}\, \mathit{Messageflow}]\end{aligned}$$

- A BPMN diagram is a non-empty finite set of *well-formed* states $\mathit{WCF} : \mathbb{P}(\mathbb{P}\, \mathit{State})$;

$$\begin{aligned}&\mathit{BPD} ::= \mathit{states}\langle\!\langle \mathit{WCF} \rangle\!\rangle\\ &\mathit{Env} == \mathit{BName} \mathrel{+\!\!\!\rightarrow} \mathit{BPD}\end{aligned}$$

- A process semantics for BPMN in CSP.

$$[\![.]\!] : \mathit{BName} \mathrel{+\!\!\!\rightarrow} \mathit{Env} \mathrel{+\!\!\!\rightarrow} \mathit{Process}$$

CSP

The grammar of CSP (subset).

$$P, Q ::= P \interleave Q \mid P \,[|\, A \,|]\, Q \mid P \parallel Q \mid P \setminus A \mid P \bigtriangleup Q \mid \\ P \,\Box\, Q \mid P \fatsemi Q \mid e \rightarrow P \mid \mathit{Skip} \mid \mathit{Stop}$$

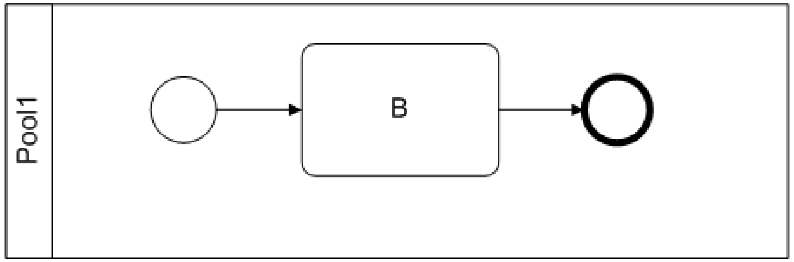
- We write $\Box\, i : \{1 \mathinner{..} n\} \bullet P(i)$ to denote $P(1) \,\Box\, \mathinner{..} \,\Box\, P(n)$, similarly for operators $\interleave$ and $\parallel$;
- Three standard behavioural models (Traces $\mathcal{T}$, Stable Failures $\mathcal{F}$, Failures-Divergences $\mathcal{N}$);
- Formal verification via refinement checks;
- FDR - automated CSP model checker.

see: CSP

A Workflow Activity

In CSP the following process description $P1$ defines a basic activity. αP is the alphabet of process P.

$$\begin{aligned}
P1 = \text{let } X &= \Box\, i : (\alpha Y \setminus \{\, \mathit{fin}.1 \,\}) \bullet (i \rightarrow X \,\Box\, \mathit{fin}.1 \rightarrow \mathit{Skip}) \\
Y &= (S \parallel B \parallel E) \\
S &= \mathit{int}.b \rightarrow \mathit{fin}.1 \rightarrow \mathit{Skip} \\
B &= (\mathit{int}.b \rightarrow \mathit{st}.b \rightarrow \mathit{int}.e \rightarrow B) \,\Box\, \mathit{fin}.1 \rightarrow \mathit{Skip} \\
E &= \mathit{int}.e \rightarrow \mathit{fin}.1 \rightarrow \mathit{Skip} \\
\text{within } & (Y \,[\![\, \alpha Y \,]\!]\, X) \setminus \{\!|\, \mathit{int} \,|\!\}
\end{aligned}$$



Case Study

We present two simple examples based on a ticket reservation system adopted from Web Services Choreography Interface specification document.

- 1st example: single business process (orchestration) to show **consistency** check.
- 2nd example: business collaboration (choreography) to show **compatibility** check.

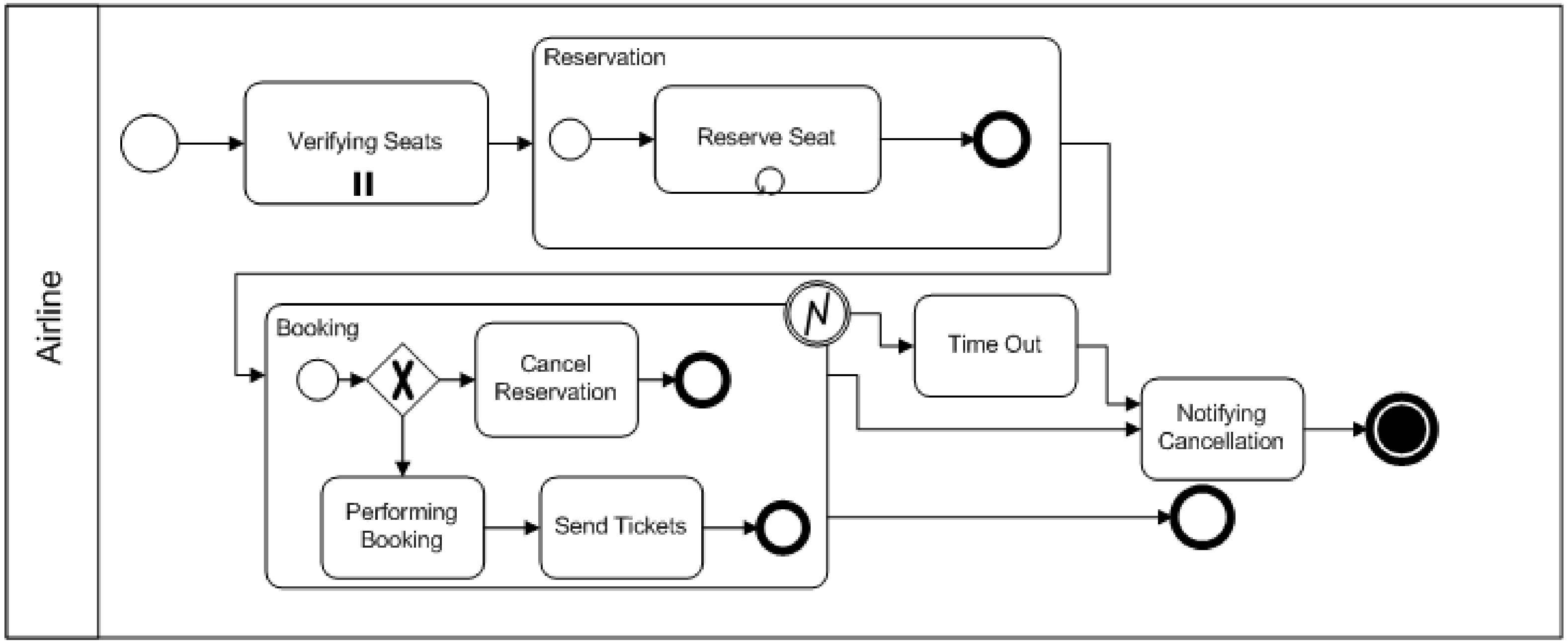
Orchestration

- Service composition;
- Local domain;
- Single participant viewpoint;
- Executable (BPEL) or Abstract (WSCI interface);
- Individual BPMN Pool

Choreography

- Collaboration protocol;
- Global domain;
- Multi-participant viewpoint;
- Abstract (not executable) (WSCDL or WSCI Global Model);
- Message flows between BPMN Pools

A Single Business Process



see: Booking subprocess

see: Process in CSP

see: Consistency Verification

A Single Business Process (cont.)

Let J be the index set:

$$\begin{aligned} \mathit{Airline} = \text{let } & X = \Box\, i : (\alpha Y \setminus \{ \mathit{fin}.1, \mathit{abt}.1 \}) \bullet \\ & \qquad (i \rightarrow X \,\Box\, \mathit{abt}.1 \rightarrow \mathit{Stop} \,\Box\, \mathit{fin}.1 \rightarrow \mathit{Skip}) \\ & Y = (\parallel j : J \bullet \alpha P(j) \circ P(j)) \\ \text{within } & (Y \; [\![\alpha Y]\!] \; X) \setminus \{\!| \mathit{init} |\!\} \end{aligned}$$

$$\begin{aligned} P(\mathit{timeout}) &= (\mathit{int.timeout} \rightarrow \mathit{st.timeout} \rightarrow \mathit{int.notify2} \rightarrow P(\mathit{notify})) \\ &\qquad \Box\, (\mathit{fin}.1 \rightarrow \mathit{Skip}) \\ P(\mathit{notify}) &= ((\mathit{int.notify1} \rightarrow \mathit{Skip} \,\Box\, \mathit{int.notify2} \rightarrow \mathit{Skip}) \,\mathbin{\S}\, \\ &\qquad \mathit{st.notify} \rightarrow \mathit{int.abt} \rightarrow P(\mathit{notify})) \,\Box\, (\mathit{fin}.1 \rightarrow \mathit{Skip}) \end{aligned}$$

see: BPMN diagram

A Single Business Process (cont.)

Booking subprocess, let J'' be the index set:

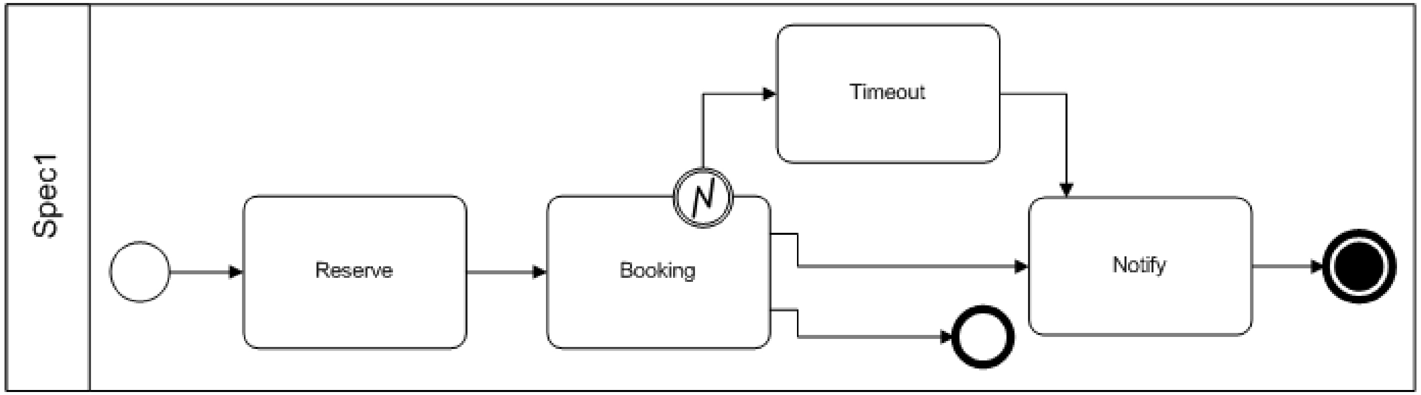
$$\begin{aligned}
\mathit{Booking} = \mathbf{let}\ X &= \Box\, i : (\alpha Y \setminus \{\mathit{fin}.3, \mathit{fin}.4\}) \bullet \\
&\quad (i \rightarrow X \mathrel{\Box} (\mathit{fin}.3 \rightarrow \mathit{Skip} \mathrel{\Box} \mathit{fin}.4 \rightarrow \mathit{Skip})) \\
Y &= (\|\, j : J'' \bullet \alpha P(j) \circ P(j)) \\
\mathbf{within}\ & (Y \mathbin{[|}\, \alpha Y \,\mathbin{|]} X) \setminus \{|\mathit{int}|\}
\end{aligned}$$

$$\begin{aligned}
P(\mathit{start2}) &= (\mathit{int}.\mathit{xs3} \rightarrow P(\mathit{start2})) \mathrel{\Box} (\mathit{fin}.3 \rightarrow \mathit{Skip} \mathrel{\Box} \mathit{fin}.4 \rightarrow \mathit{Skip}) \\
P(\mathit{xs3}) &= ((\mathit{int}.\mathit{xs3} \rightarrow (\mathit{int}.\mathit{pbooking} \rightarrow \mathit{Skip} \mathrel{\Box} \mathit{init}.\mathit{cancel} \rightarrow \mathit{Skip})) \fatsemi \\
&\qquad P(\mathit{xs3})) \mathrel{\Box} (\mathit{fin}.3 \rightarrow \mathit{Skip} \mathrel{\Box} \mathit{fin}.4 \rightarrow \mathit{Skip})
\end{aligned}$$

see: BPMN diagram

Consistency Verification

- CSP's process refinement allows us to design and construct specifications using BPMN;



- We ask FDR to verify the following refinement assertion;

$$Spec1 \sqsubseteq_{\mathcal{F}} (Airline \setminus (\alpha Airline \setminus \alpha Spec1))$$

- This refinement check demonstrates semantic **consistency** between different levels of abstraction.

see: BPMN diagram

Collaboration - Global Model

- A collaboration of business processes hence is the parallel composition of processes each corresponding to a local participant.

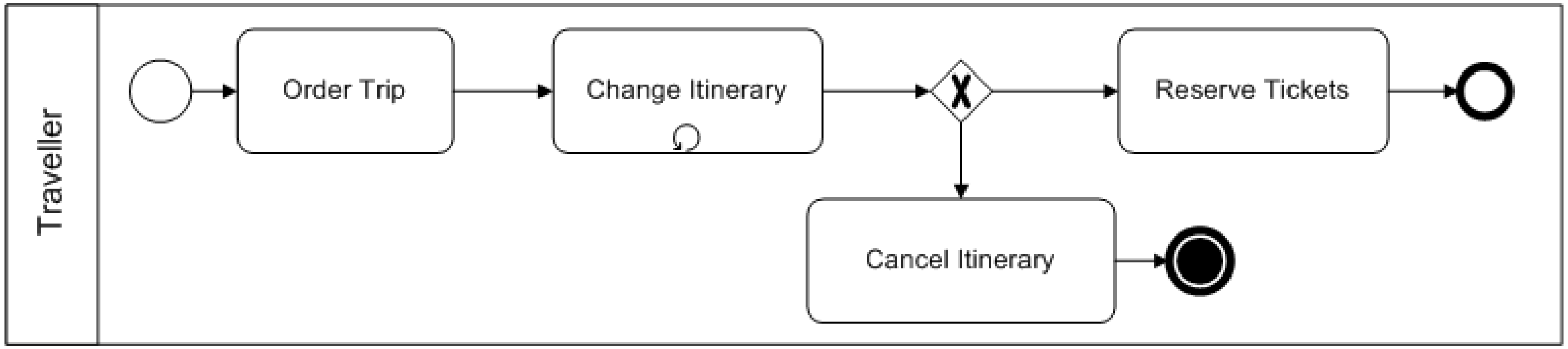
$$\mathit{Collab} = (\mathit{Trm} \parallel \mathit{Ag}) \setminus \{\!\!\{ \mathit{msg} \}\!\!\}$$

- Trm is the model of the traveller participant;
- Ag is the model of the travel agent participant

see: Collaboration in BPMN

see: Compatibility Verification

Compatibility Verification



- We ask FDR to verify the following refinement assertion;

$$Tr \sqsubseteq_{\mathcal{F}} (Collab \setminus (\alpha Collab \setminus \alpha Tr))$$

- This refinement check tells us whether the collaboration behaves as specified by the traveller participant;
- This requires the travel agent to be **compatible** with the traveller participant.

see: Collaboration in CSP

Compatibility Verification (cont.)

- The refinement assertion **does not hold** and a deadlock has occurred;
- Participants in the collaboration are **incompatible**;
- The following counterexample is given by FDR.

$$(\langle st.tr.order, st.tr.cancel\rangle, \Sigma)$$

Compatibility Verification (cont.)

Detailed analysis of the failures of Trm and Ag may be carried out:

$$(\langle st.tr.order, msg.order.in, msg.order.out, \\ msg.change.end, starts.tr.cancel\rangle, ref1)$$

$$(\langle msg.order.in, st.ag.order, msg.order.out, \\ msg.change.end\rangle, ref2)$$

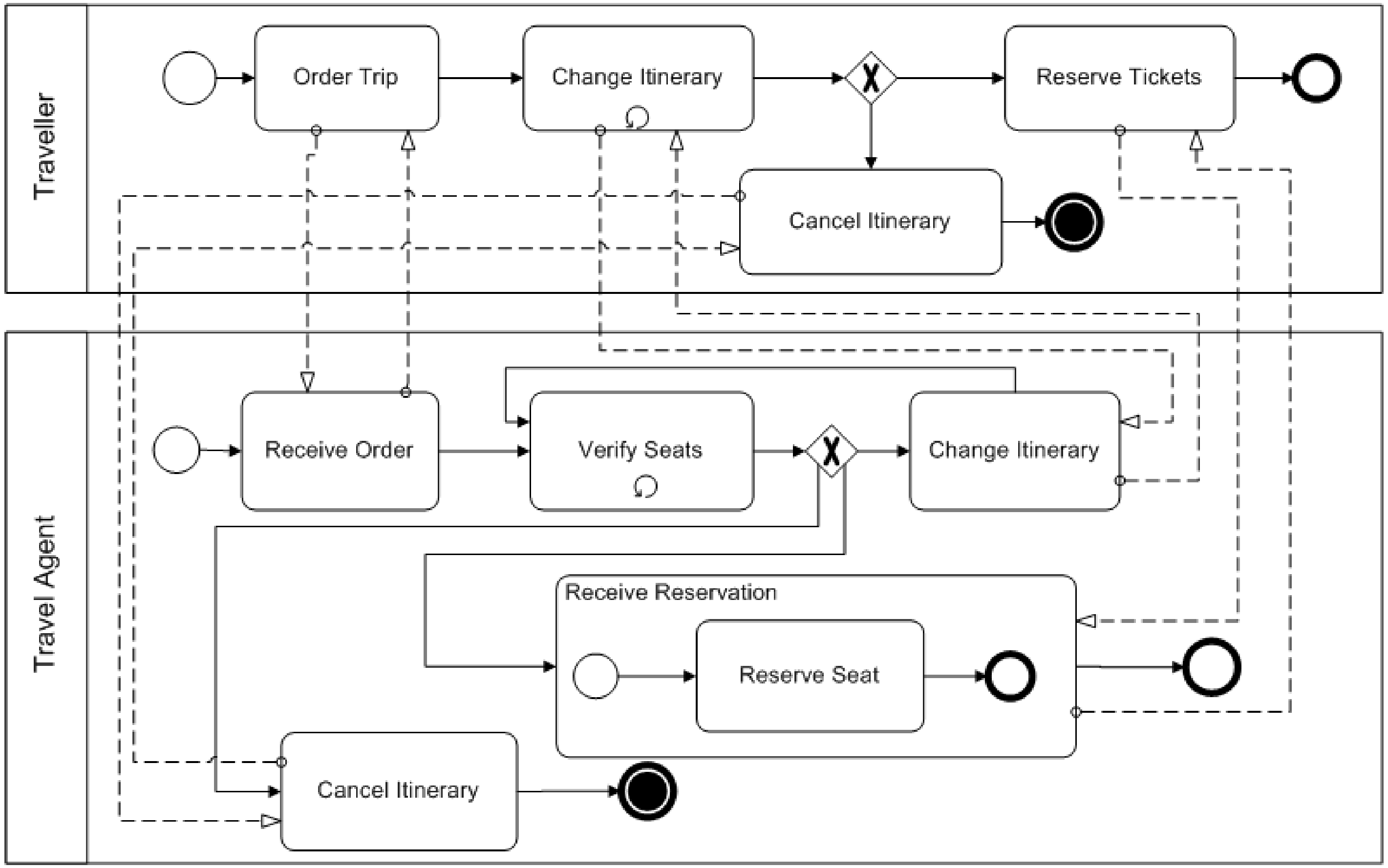
where $msg.cancel.in \notin ref1$ and $msg.cancel.in \in ref2$.

Compatibility Verification (cont.)

- The traveller may cancel her itinerary before deciding to reserve her ticket, and send a message to the travel agent about the cancellation;
- The travel agent may only carry out her cancellation after entering the reservation phase, and hence **may not receive** the message from the traveller.

see: Collaboration in BPMN

Compatibility Verification (cont.)



see: Collaboration in BPMN

see: Refinement Checks

Compatibility Verification (cont.)

We ask FDR to verify the following refinement assertion:

$$Tr \sqsubseteq_{\mathcal{F}} (Collab2 \setminus (\alpha Collab2 \setminus \alpha Tr))$$

where $Collab2 = (Trm \parallel Ag2) \setminus \{\!|msg|\!\}$

see: Model Correction

Future Work

CancerGrid :

- standardise trial model and CONSORT compliance;
- provide a SOA framework for trial software generation;
- Toward a BPM-based support for clinical trial specification

Ongoing Work:

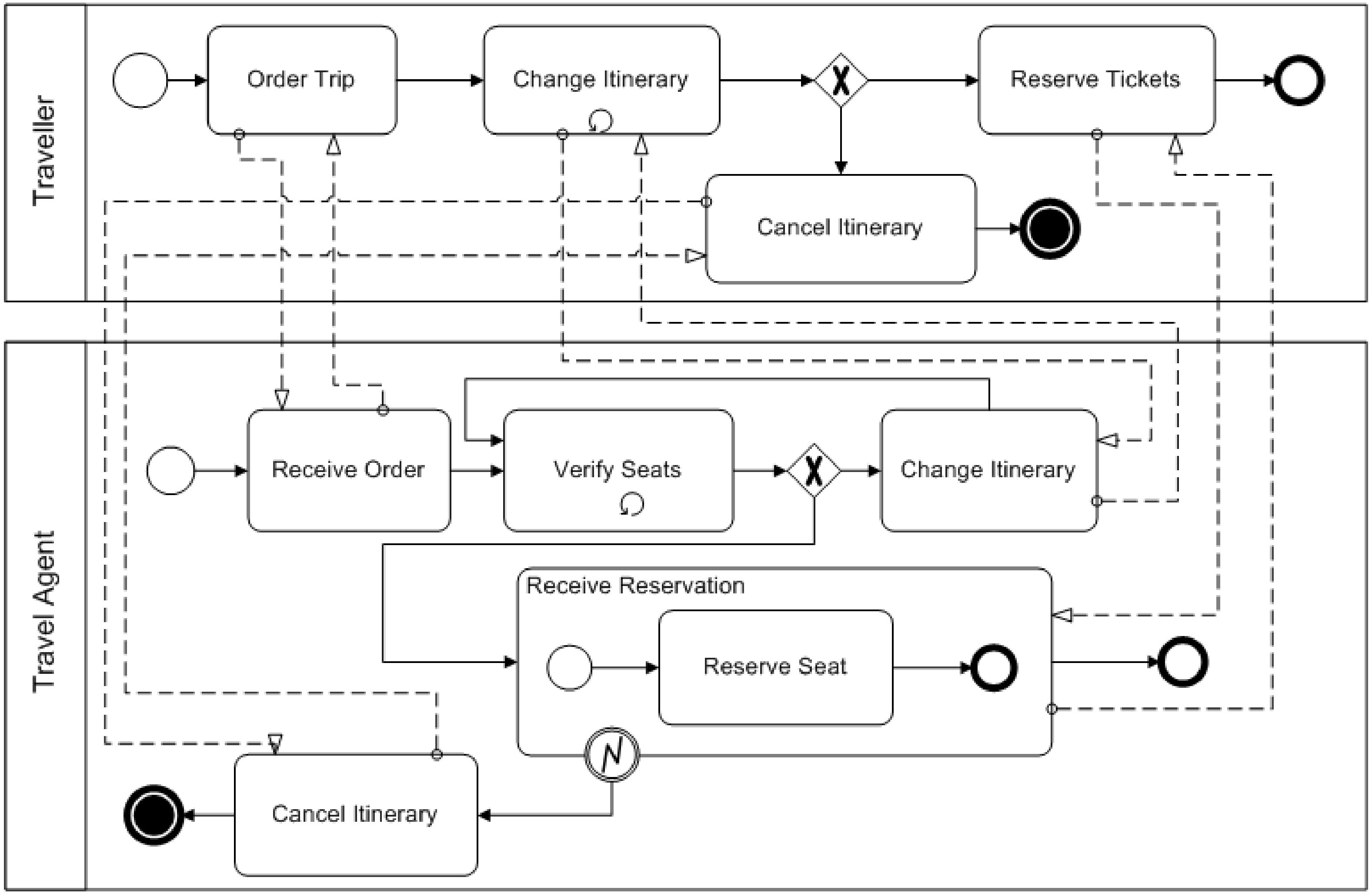
- Extend BPMN for capturing medical information;
- Compensation and Association (Dataflow);
- Automate our translation using an existing BPMN graphical editor

Thank You

Web site: `http://www.comlab.ox.ac.uk/peter.wong/`
Email: `peter.wong@comlab.ox.ac.uk`

Index

Ticket Reservation Collaboration



see: Introduction

CSP

The grammar of CSP (subset).

$$P, Q ::= P \parallel\!| Q \mid P [| A |] Q \mid P \parallel Q \mid P \setminus A \mid P \bigtriangleup Q \mid$$
$$P \Box Q \mid P \mathbin{;} Q \mid e \rightarrow P \mid \mathit{Skip} \mid \mathit{Stop}$$

- Skip, Stop - termination;.
- $e \rightarrow P$ - prefixing;
- $P \mathbin{;} Q$ - sequential composition.

see: CSP

CSP

The grammar of CSP (subset).

$$P, Q ::= P \mid\mid\mid Q \mid P [| A |] Q \mid P \parallel Q \mid P \setminus A \mid P \bigtriangleup Q \mid$$
$$P \Box Q \mid P \fatsemi Q \mid e \rightarrow P \mid \mathit{Skip} \mid \mathit{Stop}$$

- $P \mid\mid\mid Q$ - interleaving;
- $P [| A |] Q$ - partial interleaving;
- $P \parallel Q$ - parallel composition.

see: CSP

CSP

The grammar of CSP (subset).

$$P, Q ::= P \mathbin{|||} Q \mid P \mathbin{[|\, A \,|]} Q \mid P \parallel Q \mid P \setminus A \mid P \mathbin{\triangle} Q \mid$$
$$P \mathbin{\Box} Q \mid P \mathbin{;} Q \mid e \rightarrow P \mid \mathit{Skip} \mid \mathit{Stop}$$

- $P \setminus A$ - hiding;
- $P \mathbin{\triangle} Q$ - interrupt;
- $P \mathbin{\Box} Q$ - external choice.

see: CSP

CSP

The grammar of CSP (subset).

$$\begin{aligned} P, Q ::= {} & P \mathbin{|||} Q \mid P \mathbin{[|\, A \,|]} Q \mid P \parallel Q \mid P \setminus A \mid P \bigtriangleup Q \mid \\ & P \mathbin{\Box} Q \mid P \mathbin{⨾} Q \mid e \rightarrow P \mid \mathit{Skip} \mid \mathit{Stop} \end{aligned}$$

- Traces refinement $(\mathit{traces} : \mathit{CSP} \rightarrow \mathbb{P}(\mathrm{seq}\, \Sigma))$

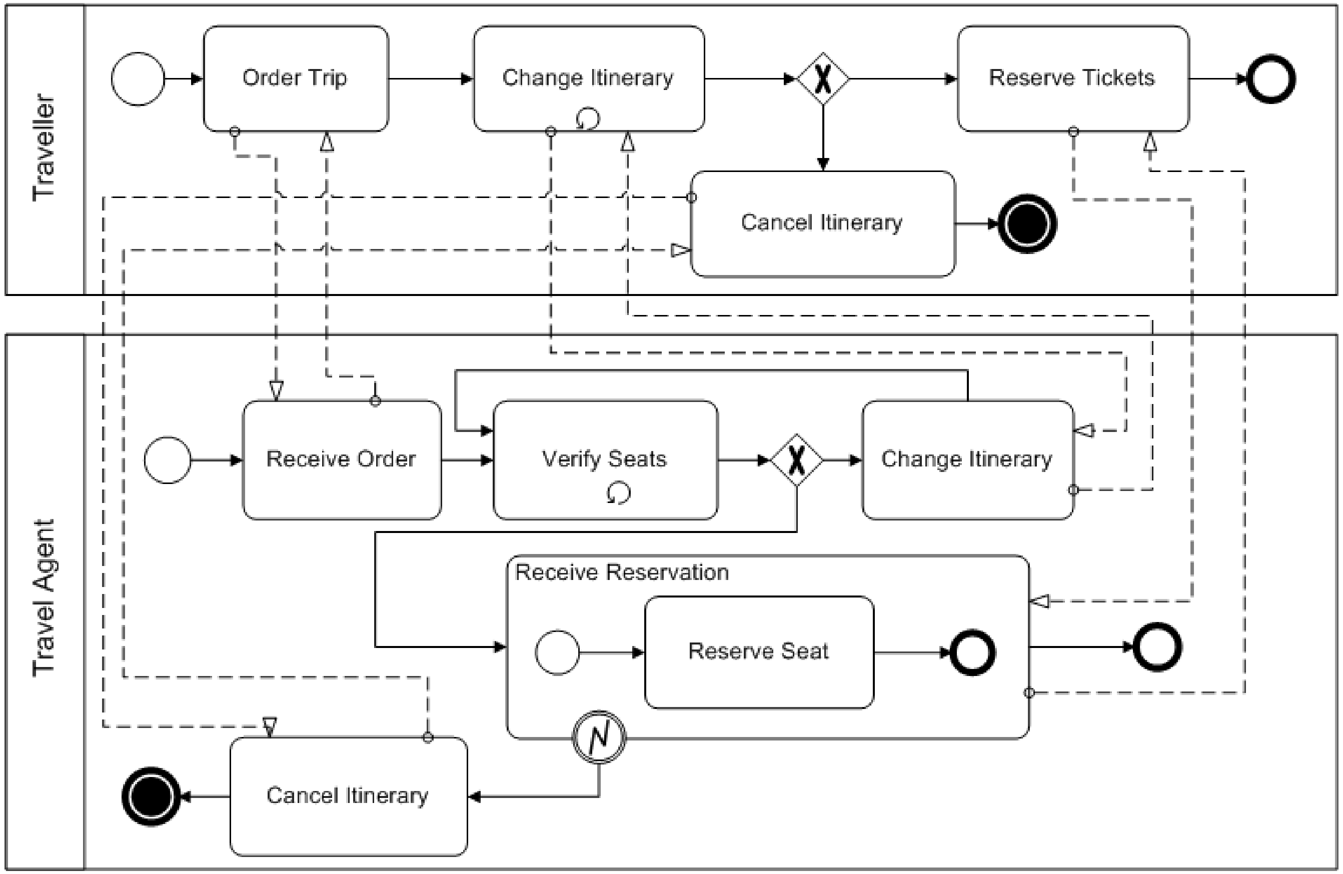
$$\begin{array}{|l} \_ \sqsubseteq_{\mathcal{T}} \_ : \mathit{CSP} \leftrightarrow \mathit{CSP} \\ \hline \forall\, P, Q : \mathit{CSP} \bullet P \sqsubseteq_{\mathcal{T}} Q \Leftrightarrow \mathit{traces}(P) \supseteq \mathit{traces}(Q) \end{array}$$

- Failures refinement $(\mathit{failures} : \mathit{CSP} \rightarrow \mathbb{P}(\mathrm{seq}\, \Sigma \times \mathbb{P}\, \Sigma))$

$$\begin{array}{|l} \_ \sqsubseteq_{\mathcal{F}} \_ : \mathit{CSP} \leftrightarrow \mathit{CSP} \\ \hline \forall\, P, Q : \mathit{CSP} \bullet \\ \quad P \sqsubseteq_{\mathcal{F}} Q \Leftrightarrow \mathit{traces}(P) \supseteq \mathit{traces}(Q) \wedge \mathit{failures}(P) \supseteq \mathit{failures}(Q) \end{array}$$

see: CSP

Ticket Reservation Collaboration



see: Introduction

see: Collaboration in CSP

see: Model Correction and Error