Cassiopeia
House of Computer Science

Contract-Oriented
Software Development
for Internet Services

Analyzing Service Contract with Model Checking

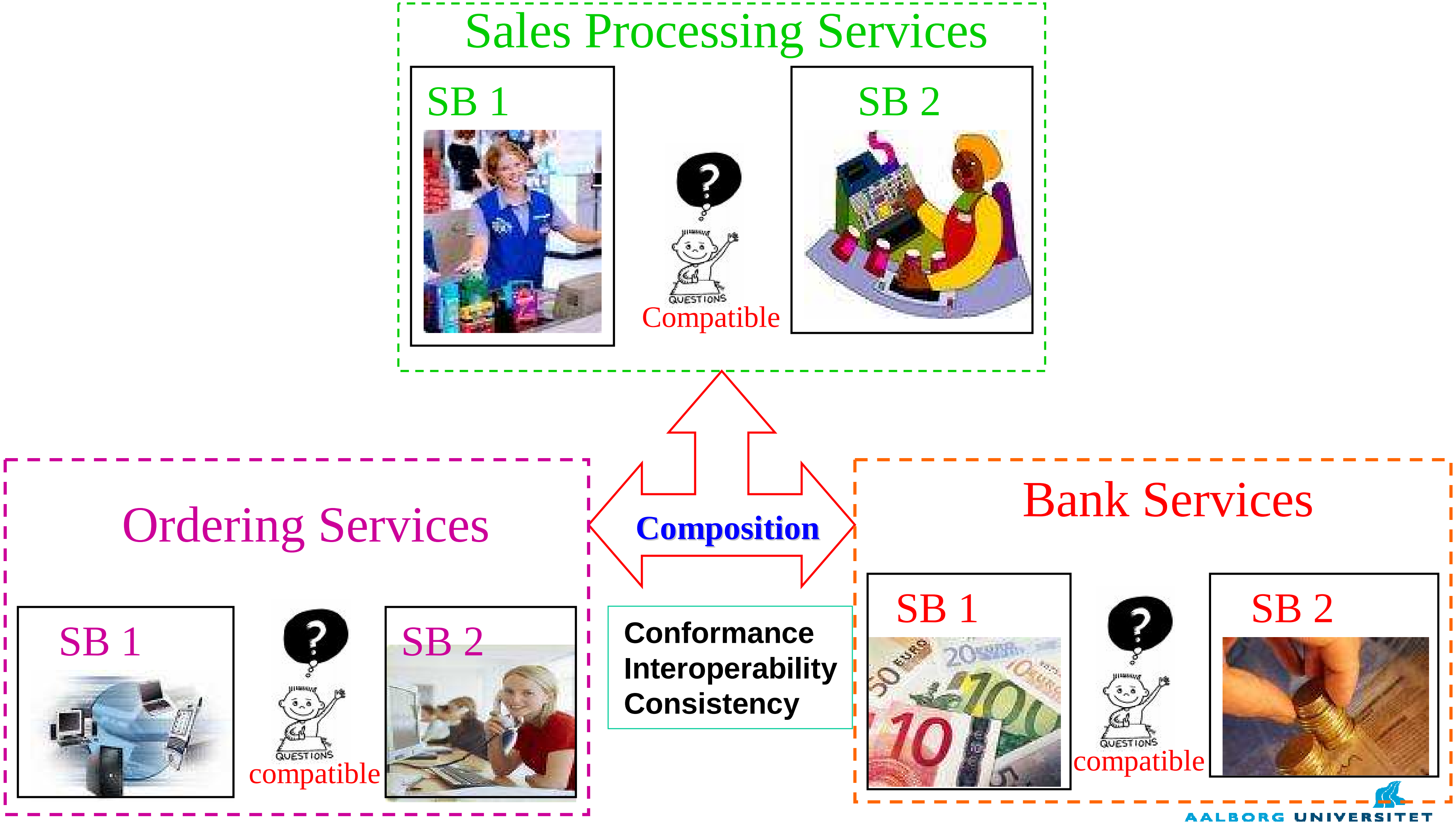
Joseph C. Okika, Anders P. Ravn

Department of Computer Science
Aalborg University, Denmark

FLACOS 2009 - Toledo, Spain, September 24-25, 2009

AALBORG UNIVERSITET

SOA - better Service still a major concern



BPEL – Business Process Execution Language

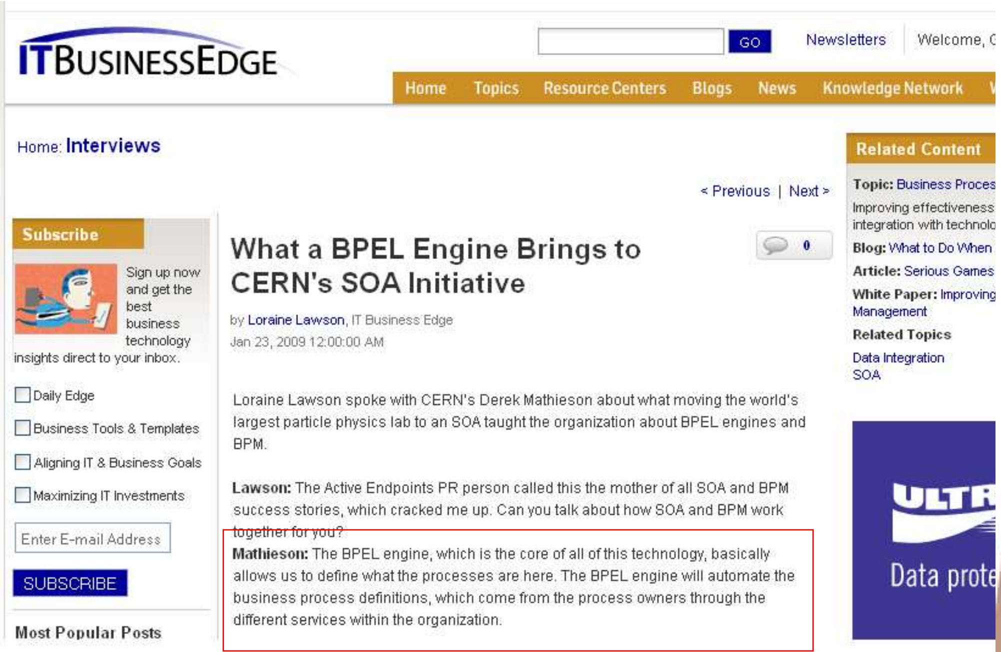
- Emerging Web Service standard
- Specifies behavioral aspects of a service
- Partnerlinks and activities to model the service interaction

```
<?xml version="1.0" encoding="utf-8" ?>
<process name="Travel"
xmlns="http://schemas.xmlsoap.org/ws/2003/03/
business-process/"
...
<partnerLinks>
    <partnerLink name="client"
            partnerLinkType="trv:travelLT"
            myRole="travelService"/> ...
    <partnerLink name="employeeTravelStatus"
            partnerLinkType="emp:employeeLT"
            partnerRole="employeeTravelStatusService"/>
...
```

```
 ...
  <partnerLink name="AmericanAirlines"
       partnerLinkType="aln:flightLT"
       myRole="airlineCustomer"
       partnerRole="airlineService"/>
  <partnerLink name="DeltaAirlines"
       partnerLinkType="aln:flightLT"
       myRole="airlineCustomer"
       partnerRole="airlineService"/>
</partnerLinks>
<!-- Variables are declared here-->
<sequence>
     <receive partnerLink="client"
          portType="trv:TravelApprovalPT"
          operation="TravelApproval"
          variable="TravelRequest"
          createInstance="yes" />
```

Why BPEL?

- ✓ Real world examples.
  - ➢ European Council for Nuclear Research
  - ➢ CJIB under Dutch Ministry of Justice



ITBUSINESSEDGE

GO Newsletters Welcome,

Home Topics Resource Centers Blogs News Knowledge Network

Home: **Interviews**

< Previous | Next >

Subscribe

Sign up now and get the best business technology insights direct to your inbox.

- Daily Edge
- Business Tools & Templates
- Aligning IT & Business Goals
- Maximizing IT Investments

Enter E-mail Address

SUBSCRIBE

Most Popular Posts

What a BPEL Engine Brings to CERN's SOA Initiative

by Loraine Lawson, IT Business Edge
Jan 23, 2009 12:00:00 AM

Loraine Lawson spoke with CERN's Derek Mathieson about what moving the world's largest particle physics lab to an SOA taught the organization about BPEL engines and BPM.

Lawson: The Active Endpoints PR person called this the mother of all SOA and BPM success stories, which cracked me up. Can you talk about how SOA and BPM work together for you?

Mathieson: The BPEL engine, which is the core of all of this technology, basically allows us to define what the processes are here. The BPEL engine will automate the business process definitions, which come from the process owners through the different services within the organization.

Related Content

Topic: Business Proces
Improving effectiveness integration with technolo
Blog: What to Do When
Article: Serious Games
White Paper: Improving Management

Related Topics

Data Integration
SOA

Capgemini ORACLE in collaboration with Centraal Justitieel Incasso Bureau

Service-Oriented Architecture Positions the CJIB for the Future

Capgemini's Integrated Architecture Framework used to deploy Oracle E-Business Suite accelerated by BPEL Process Manager

The Situation

The Central Justitieel Incasso Bureau (CJIB) is an independent implementation authority operating under the Dutch Ministry of Justice responsible for administering, collecting and coordinating fines and sanctions levied by the Dutch judicial system. In an environment of changing laws and the CJIB's ever-expanding range of responsibilities, the CJIB needs a reliable, efficient and, most importantly, flexible IT system that can effectively manage the execution of fines and sanctions.

The Solution

The CJIB has to handle a complex combination of tasks and collection methods. It therefore chose to adopt a service-oriented architecture (SOA) based on the Oracle E-business Suite (eBS).
BPEL

Process Manager, also by Oracle, is part of this solution. This architecture allows the reuse of standard components within the executions of different fines and sanctions by the CJIB. Designing and changing processes will be dramatically simplified and require less programming.

The Result

Capgemini supported the CJIB in developing a blueprint of the Program NoorderWint to implement the future SOA architecture. As a method, Capgemini used its own Integrated Architecture Framework (IAF). After completing the implementation, the CJIB will have a reliable, flexible and future-proof system landscape.

"Thanks to the knowledge and experience of Capgemini, in combination with its IAF methodology, we jointly shaped our SOA ideas. Of absolute crucial importance for the project's feasibility is that Capgemini's architects have the skills and courage to be pragmatic!"

Jan van Dijk
CIO
Centraal Justitieel Incasso Bureau

- ✓ Real world examples…
  - ➢ European Commission
    - ➢ EMCS

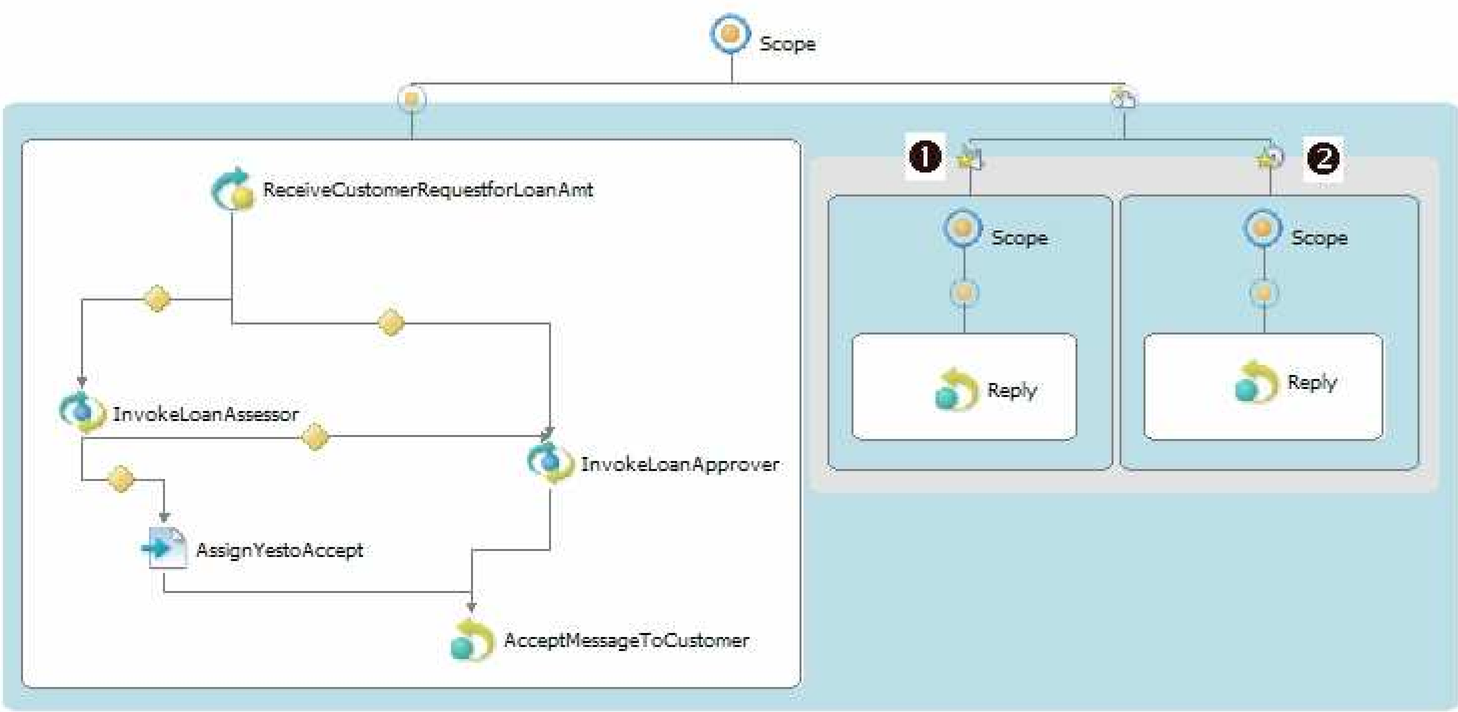
Business Process Management Suite (BPMS) for creating sophisticated business processes (including long running, asynchronous processes). The modelling is achieved using BPMN as the standard graphical notation, and XPDL as the XML-based process definition language or BPEL as the serialized XML programming language for the specification of executable business processes (applied primarily to the orchestration of Web services). Those languages improve communication and portability of process models.
They facilitate designing, defining, implementing, and deploying composite applications and services from a number of distributed and autonomous software components, offering a flexible way to achieve the required business collaborations.

- ✓ BPEL has some unique features

Event Handler

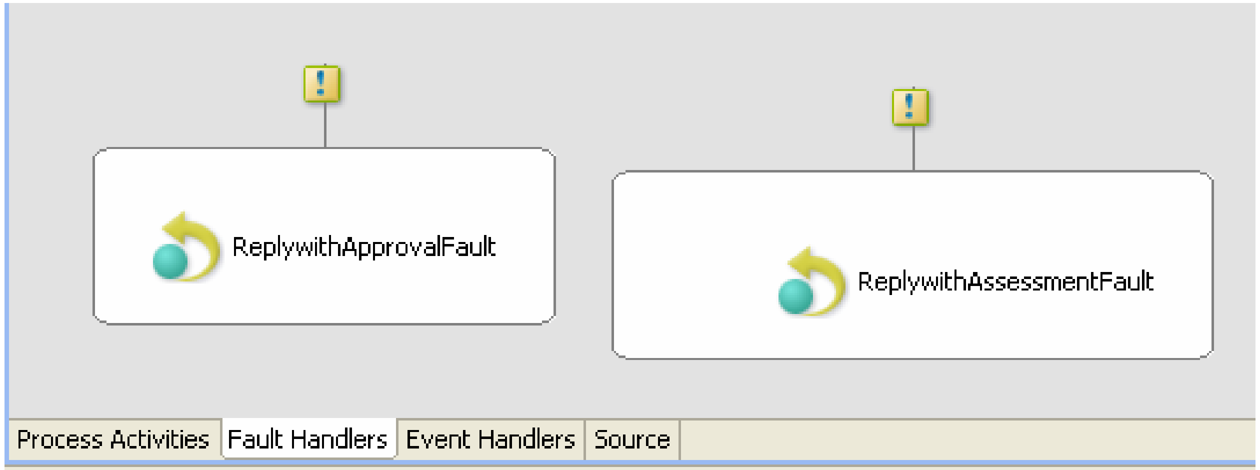
Two Constructs

1. OnEvent – message event
2. OnAlarm – alarm event



Fault Handler

- Reverse work – Undo a partial/unsuccessful work of a scope that a fault has occurred in.
- Not same as compensation handler.
- Eg. Internal process error, platform specific fault, a web service operation cannot complete successfully, a throw activity

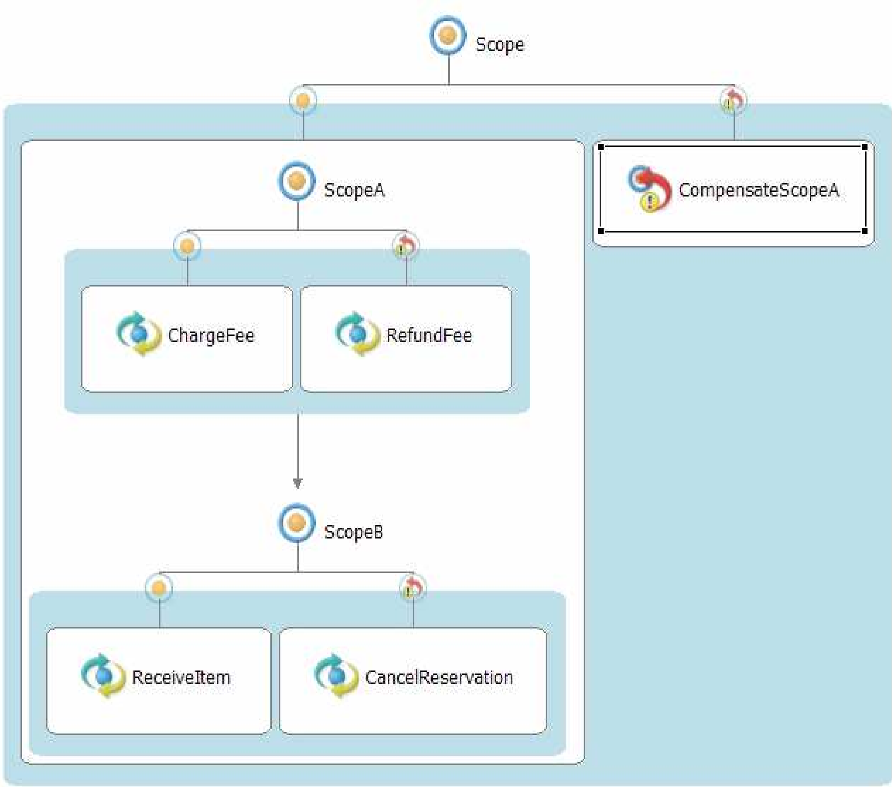


Compensation Handler

- Execution serves to reverse some previously executed application logic

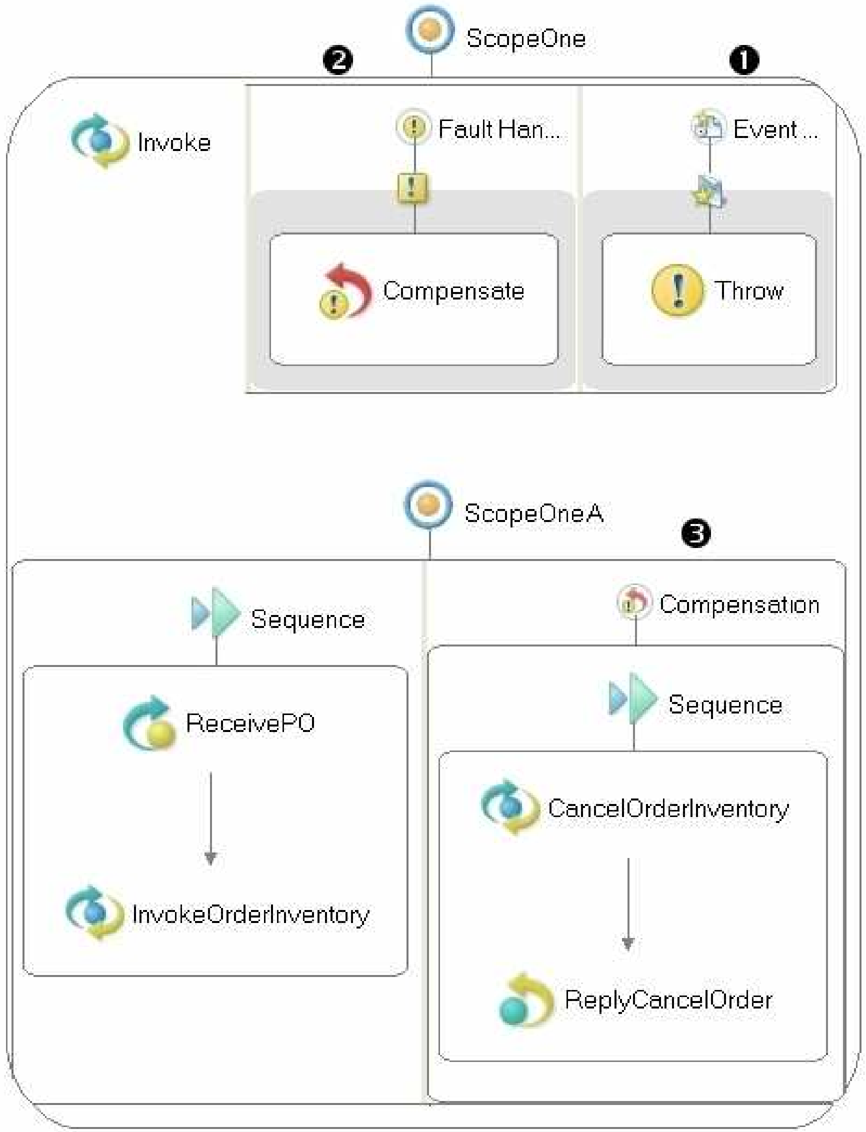
  ie. When the scope is completed
- No automatic restoration of data during compensation
- Up to the application to define its own compensation behavior

- Eg. Cancelling reservation, putting an order on hold, removing a charge on a credit card.



Putting them together

1) Event handler receives a cancellation message and throws a fault to the Fault Handler
2) Fault Handler executes a compensate for the previously completed (linked) scope
3) Compensation Handler for the completed scope rolls back the work of InvokeOrderInventory

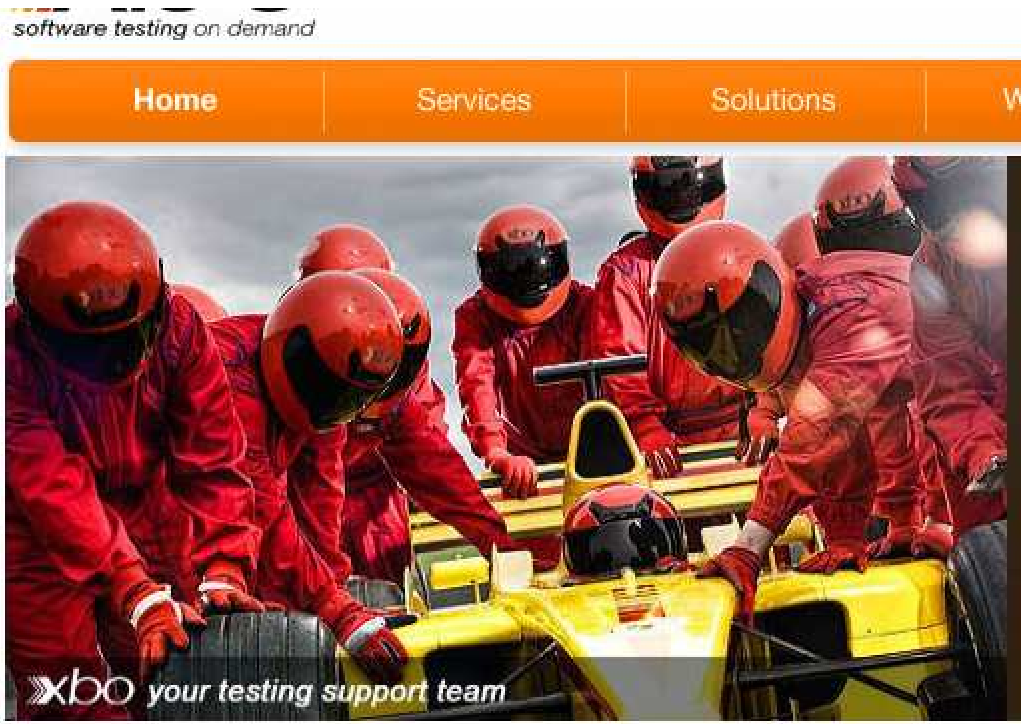


Does it really compensate
the charge on a credit card?

Does the Service Work?

We can do … Software Testing

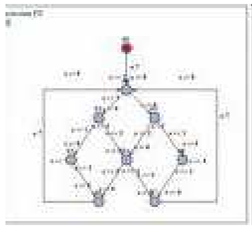
Black box, White box testing
Unit testing
Incremental Integration testing
Integration testing
Functional testing
System testing
End-to-end testing
Sanity testing
Regression testing,
Acceptance testing
Load testing
Stress testing
Performance testing
Usability testing
Security testing
Compatibility testing, etc.

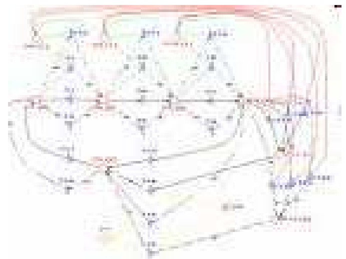


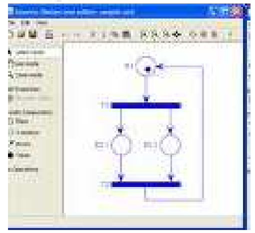
http://www.xbosoft.com/images/index/index800_07.jpg

We can do … Analysis

- based on:
  - Automata
  - Petri nets
  - Abstract State Machines
  - Process Algebra
  - etc.







$$Proc ::= \alpha.Proc \mid \lfloor Proc \rfloor (Proc) \mid Proc + Proc \mid Proc \mid Proc \mid recX.Proc \mid X \mid \Omega \mid Proc \setminus a \mid 0$$

✓ Automata/Model Checking

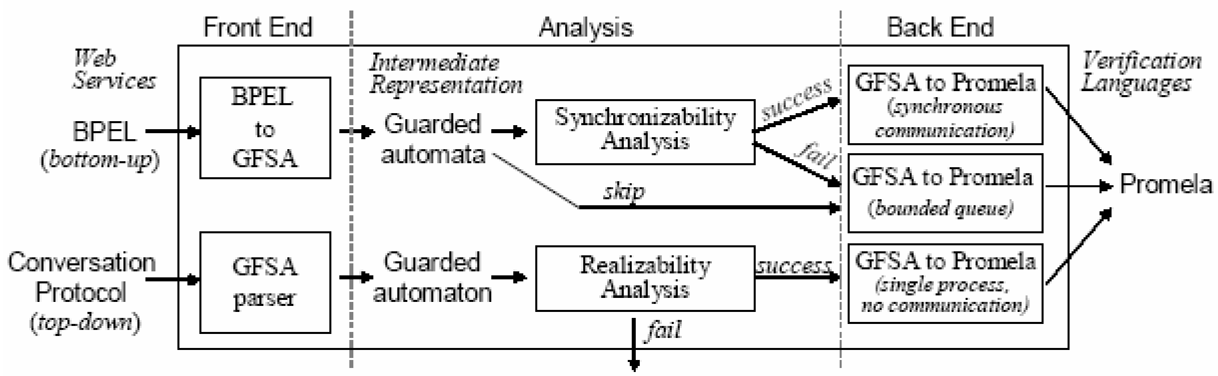
Model Checking has made some progress

Model Checking tools are reaching maturity

Tools: SPIN, NuSMV, UppAal, CWB

Related Work I

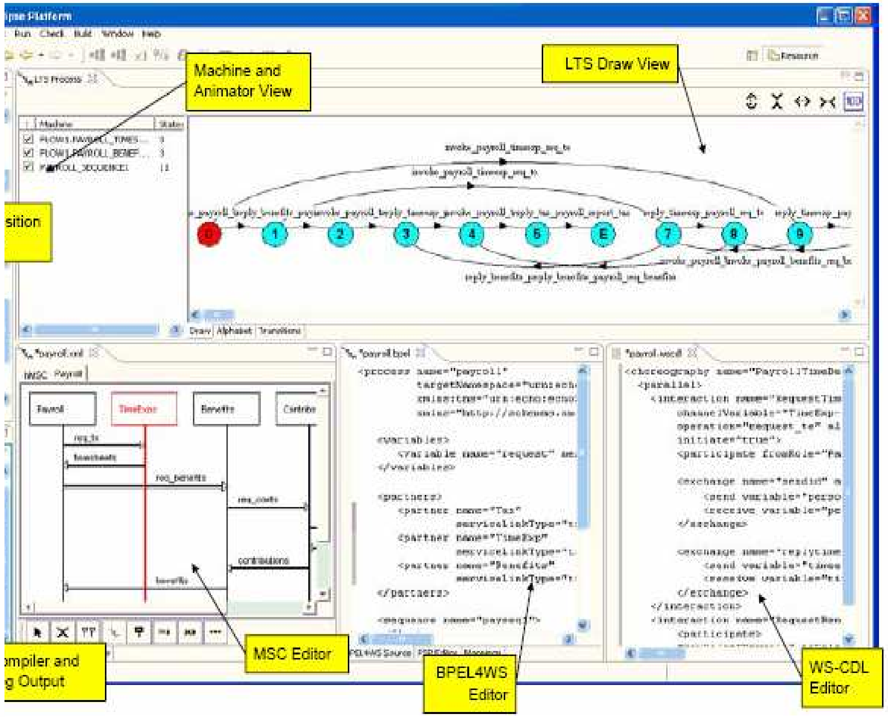
- Intermediate Representation: automata with XPath guards (called GFSA) as an intermediate representation for web services.
- A translator from BPEL4WS to GFSA is developed,
- Model checker SPIN used as the back-end of WSAT to check LTL properties.



WSAT: A Tool for Formal Analysis of Web Services: Xiang Fu, Tevfik Bultan, and Jianwen Su
Computer Aided Verification, 16th International Conference,
CAV 2004, Boston, MA, USA, July 13-17, 2004

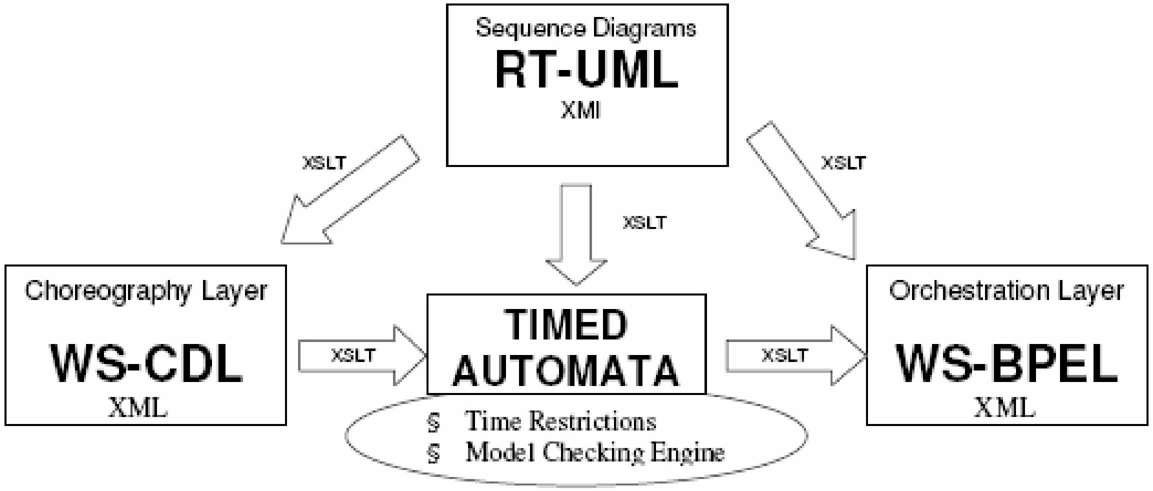
Related Work II

BPEL Construct Example	and FSP Representation
<sequence> <receive operation="1" partner> </receive> <receive operation="2" partner> </receive> </sequence>	ACT1 = (receive_p1_p2_op1 -> END). ACT2 = (receive_p1_p2_op2 -> END). SEQUENCE = ACT1;ACT2;END.
<switch name = "MPS"> <case condition= "cond1" = "true"> <receive…> <otherwise> <reply…> </switch>	SWITCH = if cond1-true then ACT1;END else if cond2-true then ACT2;END else END.
<while **condition =** "cond1" = "true"> <sequence> <receive…> </while>	WHILE = If condition-true then ACT1;WHILE else END.
<pick name ="pick1"> <onMessage> <invoke operation="1" > <onAlarm> <invoke operation="2"> </pick>	PICK1 = (event1 -> ACT1; END | event2 -> ACT2; END).
<flow name="flow1"> <receive operation="1">… <receive operation="2">… </flow>	||FLOW1 = (ACT1 || ACT2).



H. Foster, S. Uchitel, J. Magee, and J. Kramer, "**Tool Support for Model-Based Engineering of Web Service Compositions**," presented at 3rd IEEE International Conference on Web Services (ICWS2005), Orlando, FL, 2005

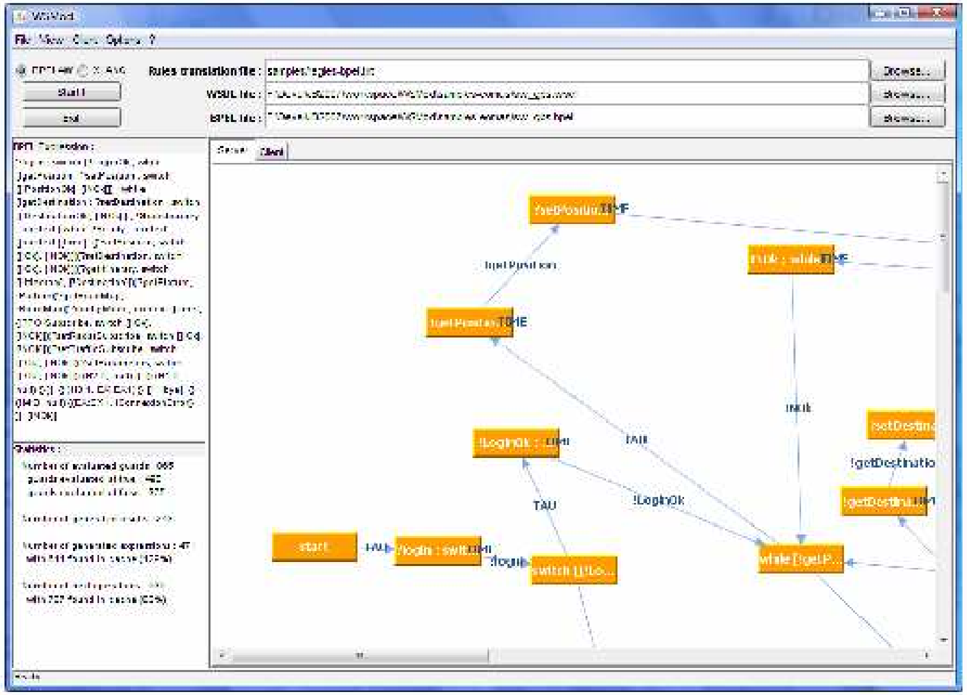
Related Work III



Using RT-UML for modelling web services
María-Emilia Cambronero, Juan José Pardo, Gregorio Diaz, Valentin Valero, SAC 2007:643-648

Related Work IV

- Exhaustive simulation based on a formalisation of BPEL semantics using the Algebra of Timed Processes (ATP).
- Models analysed by model checking value-based temporal logic properties using the CADP toolbox.



Formal Modeling and Discrete-Time Analysis of BPEL Web Services.
Radu Mateescu and Sylvain Rampacek.
International Journal of Simulation and Process Modelling 2008 - Vol. 4, No.3/4 pp. 183 – 194

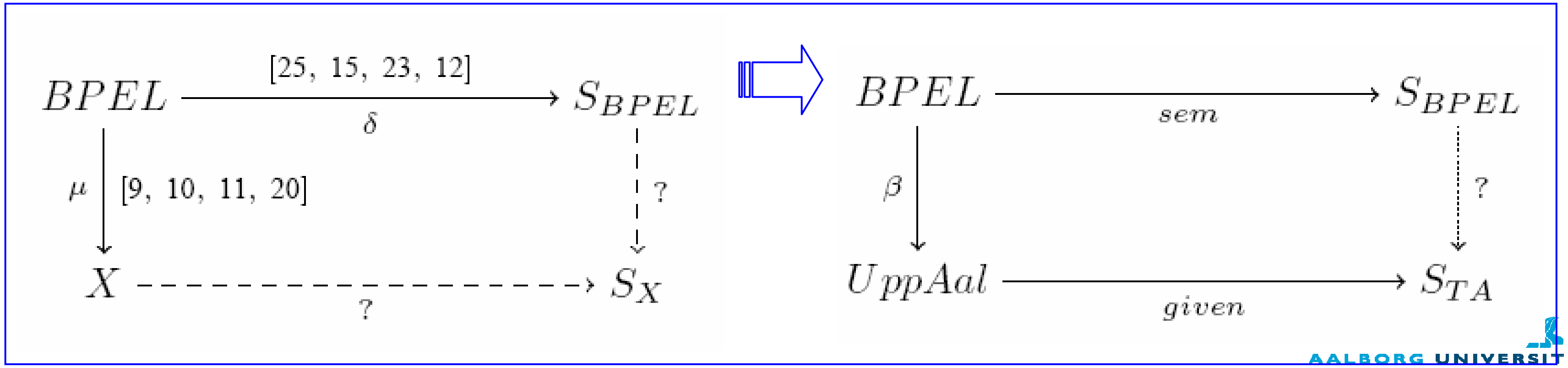
Approach

Many theoretical results, tools proposals

- semantic definition/mapping to target language/ applicability.
- Mostly fragments of BPEL without the intricate features

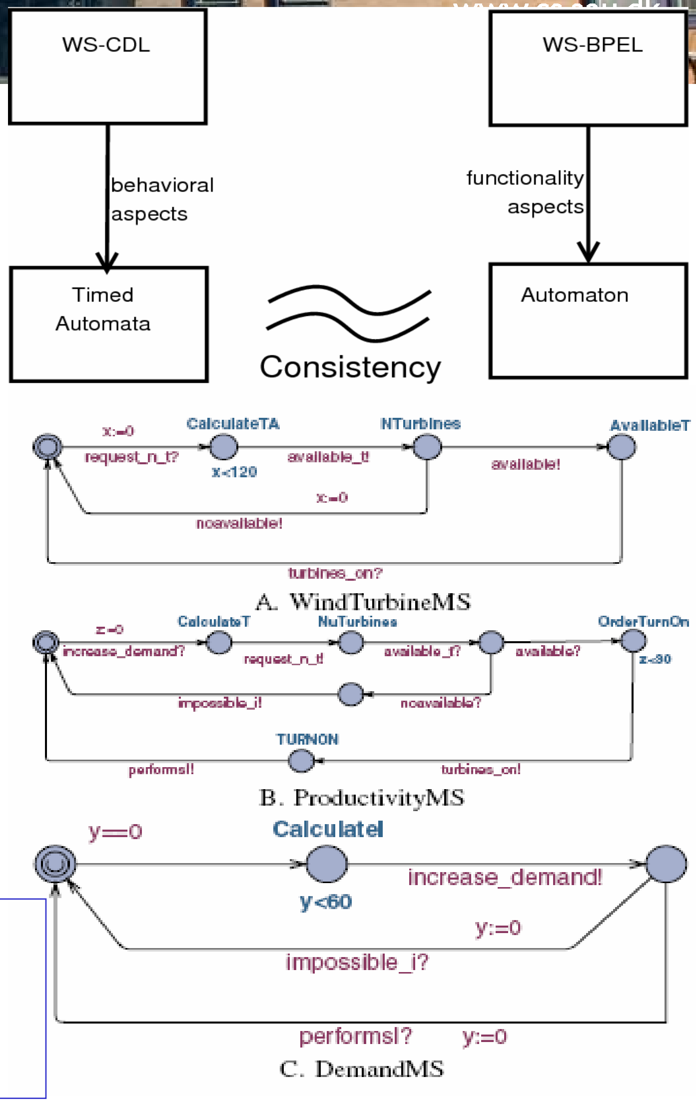
Our approach: - make a formal model of a service

- WS-BPEL
- Analysis of behaviour
  - Orchestration
  - Choreography
- UppAal



Previous Efforts

- ❖ Derive semantic models in the form of (timed) automata
- ❖ Several specifications – different aspects.
- ❖ Simulation relation



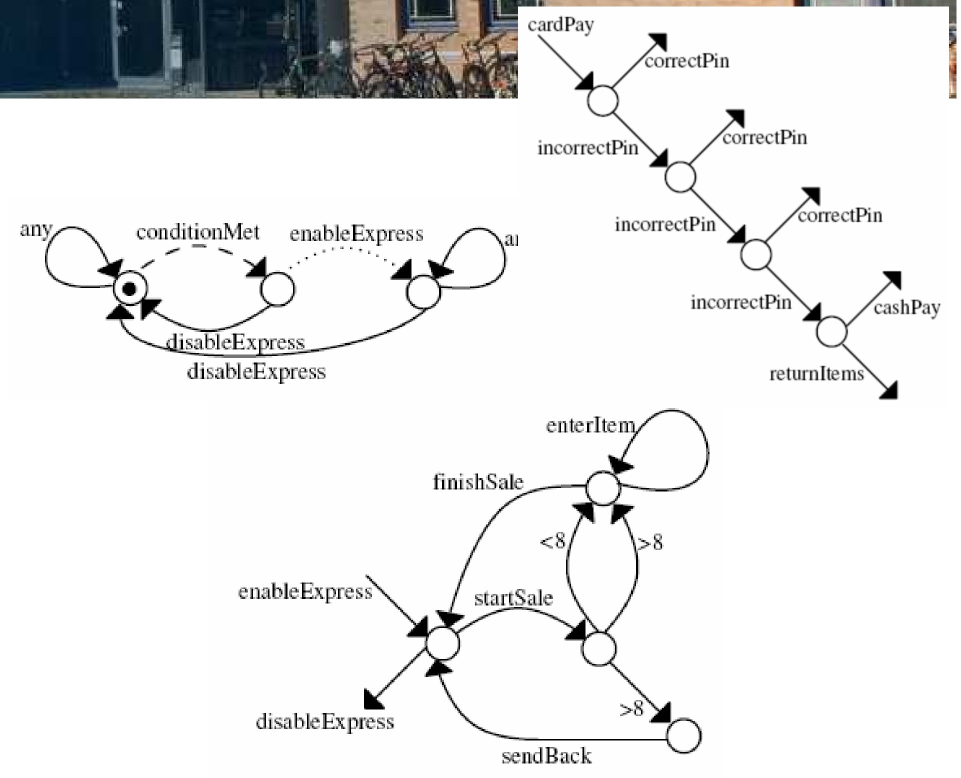
Consistency Checking of Web Service Contracts
International Journal On Advances in Systems and Measurements, issn 1942-261x vol. 1, no. 1, year 2008, http://www.iariajournals.org/systems_and_measurements/"

Analyzing Web Service Contracts : an aspect oriented approach. Cambronero, M.-Emilia ; Okika, Joseph C. ; Ravn, Anders Peter. Proceedings of the International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies : UBICOMM'2007. IEEE Computer Society, 2007. 149-154

Cassiopeia
House of Computer Science

Previous Efforts …

- Specification of the CoCoME case study
  - temporal logics, operational, deontic specification
- Comparison between contract specifications
  - card pay, express mode, sales process
- Discussion on how easy it is to analyze the specifications



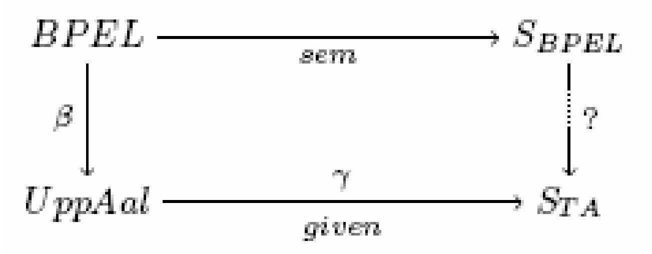
	LTL	CTL	CSP	$\mathcal{CL}$
F1	✓	✓	✓	✓
F2	–	–	–	✓
F3	–	✓	(✓)	✓

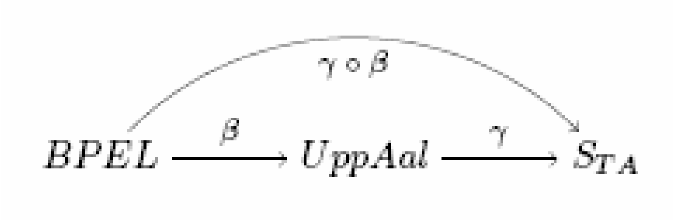
On the specification of full contracts. Stephen Fenech, Joseph Okika, Gordon J. Pace, Anders P. Ravn, and Gerardo Schneider. In 6th International Workshop on Formal Engineering approaches to Software Components and Architectures *(FESCA'09)*, ENTCS, (York, UK), March 2009

AALBORG UNIVERSITET

Current effort

- Full behavior of BPEL
- Timed Automata for the model with rendering to UppAal
- Semantics based on UppAal
  - Semantic preserving extraction/translation





- Semantics based on Rewriting Logic
  - Executable operational semantics

SOS for full BPEL (I)

- Service Interaction

Activity	Semantic Rules
receive	$(receive\ p,\ s, (\rho, \sigma)) \xrightarrow{?p} (\epsilon, s, (\rho, \sigma'))$ $(receive\ p,\ s, (\rho, \sigma)) \xrightarrow{\chi} (receive\ p,\ s, (\rho, \sigma))$
reply	$(reply\ p,\ s, (\rho, \sigma)) \xrightarrow{!p} (\epsilon, s, (\rho, \sigma))$ $(reply\ p,\ s, (\rho, \sigma)) \xrightarrow{\chi} (reply\ p,\ s, (\rho, \sigma))$
invoke	$(invoke\ p,\ s, (\rho, \sigma)) \xrightarrow{!p} (\epsilon, s, (\rho, \sigma))$ $(invoke\ p,\ s, (\rho, \sigma)) \xrightarrow{\chi} (invoke\ p,\ s, (\rho, \sigma))$ $\dfrac{(invoke\ p_1,\ s, (\rho, \sigma)) \xrightarrow{!p_1} (\epsilon, s, (\rho, \sigma')) \qquad (invoke\ p_1,\ s, (\rho, \sigma)) \xrightarrow{?p_2} (\epsilon, s, (\rho, \sigma))}{(invoke\ p_1\ p_2,\ s, (\rho, \sigma)) \xrightarrow{!p_1, ?p_2} (\epsilon,\ s, (\rho, \sigma'))}$

SOS for full BPEL (II)

- Scope + handlers

$$\textbf{scope 1:}\quad (scope\ s_0\ \mathcal{A}\ \mathcal{F}\ \mathcal{E}\ \mathfrak{C}\ \mathfrak{T}, s, (\rho,\sigma)) \xrightarrow{\tau} (sequence\ \mathcal{A}\ endscope, s_0, (\rho',\sigma'))$$

$$\textbf{where}\quad \rho' = \rho + [s_0 \mapsto (s,\rho,l_{s_{new}})] + [f \mapsto \mathcal{A} | (f,\mathcal{A}) \in \mathcal{F}] + [e \mapsto \mathcal{A} | (e,\mathcal{A}) \in \mathcal{E}]$$

$$\sigma' = \sigma + [l_{s_{new}} \mapsto (\mathfrak{C}, \mathfrak{T}, \rho')]$$

$$\textbf{scope 2:}\quad (endscope\ s, (\rho,\sigma)) \xrightarrow{\tau} (\epsilon, s', (\rho',\sigma))\ \textbf{where}\ \ s' = \rho[s]_1,\ \rho' = \rho[s]_2 + [s \mapsto \rho[s]_3]$$

$$\textbf{scope 3:}\quad (exit\ s, (\rho,\sigma)) \xrightarrow{\tau} (\epsilon, s', (\rho',\sigma))\ \textbf{where}\ \ s' = \rho[s]_1,\ \rho' = \rho[s]_2$$

$$\textbf{scope 4a:}\quad \frac{\rho[f] = \mathcal{A}}{(throw\ f,\ s, (\rho,\sigma)) \xrightarrow{\tau} (\mathcal{A}', s, (\rho,\sigma))\ \ \text{where } \mathcal{A}' = sequence\ \rho[f]\ endscope}\ (f, \mathcal{A}_f)$$

$$\textbf{scope 4b:}$$

$$\frac{\rho[f] = \mathcal{A}}{(throw\ f,\ s, (\rho,\sigma)) \xrightarrow{\tau} (\mathcal{A}', s, (\rho,\sigma))\ \ \text{where } \mathcal{A}' = sequence\ \rho[f]\ endscope}\ (f, \mathcal{A}_{catchAll})$$

$$\textbf{scope 4c:}$$

$$\frac{\rho[f] = \mathcal{A}}{(throw\ f,\ s, (\rho,\sigma)) \xrightarrow{\tau} (\mathcal{A}', s, (\rho,\sigma))\ \ \text{where } \mathcal{A}' = sequence\ \rho[f]\ endscope}\ (f, rethrow\ f)$$

$$\textbf{scope 5:}\quad (rethrow\ f,\ s, (\rho,\sigma)) \xrightarrow{\tau} (sequence\ exit\ throw\ f, s, (\rho',\sigma))$$

AALBORG UNIVERSITET

Lessons learned so far …

- Better Service is still a major concern
- Analyzing service orchestration: important for SOA and Cloud Computing
- Few semantics for orchestration analysis
- Intricate features difficult to formalize
- Difficult to create an UppAal model directly from the standard
- But automated analysis will
  - improve quality
  - can reduce cost; example testing cost

Some Issues

Suppose two concurrent isolated scopes, S1 and S2, access a common set of variables and partner links (external to them) for read or write operations. The semantics of isolated scopes ensure that the results would be no different if all conflicting activities (read/write and write/write activities) on all shared variables and partner links were conceptually reordered so that either all such activities within S1 are completed before any in S2 or vice versa.

- Concurrent scope with compensation
- Non-deterministic behaviour due to cascading compensation
- Non-termination
- Data (infiniteness) handling
- Properties (common to every service)
- …