Contract-Oriented Software Development for Internet Services – *What is it?*

NorduNet3 Project 2006 - 2010

FLACOS’09
Toledo September 2009
Olaf Owe, Cristian Prisacariu, Gerardo Schneider, Oslo University
Gordon Pace, University of Malta
Björn Bjurling, Swedish Institute of Computer Science
Joseph C. Okika, and Anders P. Ravn, Aalborg University

www.ifi.uio.no/cosodis/

Why Internet Services?

Technology allows collaboration across the net.

**CS background:**
- Concurrency Theory 1970 –
- Distributed databases, transactions etc. 1980 -
- Distributed Operating Systems 1990 –
- CORBA 2000 -

+ Internet
Why Contracts?

Collaboration across organizational domains presumes trust, but...

*When trust is insufficient, use contracts*

Software Development?

- Developers need language support to program services that are:
  - Distributed
  - Interoperable
  - Discoverable
  - Contract-aware

*Contract-Oriented Software Development for Internet Services*

*What is it?*
COSoDIS Mission

1. develop novel approaches to implement and reason about contracts in a service oriented architecture.
2. design and give proof of usefulness of system modeling tools and programming language tools
3. to empower SOA developers to deploy highly-dynamic, negotiable and monitorable Internet services.

Key Issues for Contracts

• Definition

\[ C \]

Cristian Prisacariu and Gerardo Schneider,
CL: An Action-based Logic for Reasoning about Contracts, LNCS 5514, June 2009

• Contract checking

\[ C \neq \emptyset, C_1 \leq C_2 \]

Stephen Fenech, G. Pace, and G. Schneider.
Clan: A tool for contract analysis and conflict discovery.
LNCS 5799, October 2009.

• Conformance checking

\[ P \models C \ast \]

• Monitoring

\[ M(P) \parallel I(C) \]

Christian Colombo, G. Pace, and G. Schneider.
Dynamic event-based runtime monitoring of real-time and contextual properties,
LNCS 5996, September 2009
The Marketplace

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Language/Approach</th>
<th>Language/Approach</th>
<th>Language/Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface</td>
<td>WSDL</td>
<td>OWL-S</td>
<td>ebBSI</td>
</tr>
<tr>
<td>Functionality</td>
<td>WS-BPEL, WSOL</td>
<td>OWL-S (IQPE),</td>
<td>ebBPSS</td>
</tr>
<tr>
<td>Protocol</td>
<td>WS-BPEL, WS-CDL</td>
<td>WSMO, OWL-S</td>
<td>ebBPSS</td>
</tr>
<tr>
<td>Security</td>
<td>WS-Security</td>
<td>OWL-S</td>
<td>ebCPA(SecurityPolicy)</td>
</tr>
<tr>
<td>QoS</td>
<td>WS-Policy</td>
<td>OWL-S</td>
<td>ebCPP(XMLDSIG)</td>
</tr>
<tr>
<td></td>
<td>WS-Trust</td>
<td>WSMO</td>
<td>ebCPA</td>
</tr>
<tr>
<td></td>
<td>WSOL</td>
<td>WSML</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WS-LA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Emilia Cambronero, Joseph C. Okika, and Anders P. Ravn, Consistency Checking of Web Service Contracts

Expected Results - 2010

- Applications
  - Logics.
  - Creol, Java, etc
  - WS-BPEL

- Models
  - WS-CDL

- Implementations.
  - CREOL, Java, etc

- Platform
  - Larger Case Study?
  - Monitoring?

- A modal logic for defining high level contracts
  \[ C \models C \neq \emptyset \]
  \[ C_1 \leq C_2 \]

- Model checking tools for checking WS*-style contracts
  \[ P \models C \]

- Monitoring
  \[ M(P) || I(C) \]
SOA is really New!  (Wolfgang Reisig)

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Application Area</th>
<th>Enterprise Computing</th>
<th>Embedded Systems</th>
<th>Service Oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface</td>
<td></td>
<td>Database Schema</td>
<td>RT-profile</td>
<td>Dist. Objects</td>
</tr>
<tr>
<td>Functionality</td>
<td></td>
<td>Queries</td>
<td>Control algorithms</td>
<td>Components</td>
</tr>
<tr>
<td>Protocol</td>
<td></td>
<td>Workflow</td>
<td>Reactive processes</td>
<td>Orchestration</td>
</tr>
<tr>
<td>Dependability</td>
<td></td>
<td>Integrity</td>
<td>Timeliness</td>
<td>Availability</td>
</tr>
<tr>
<td>Fault Tolerance</td>
<td></td>
<td>Transactions</td>
<td>Replicated Processes</td>
<td>Compensation</td>
</tr>
<tr>
<td>QoS</td>
<td></td>
<td>Performance</td>
<td>Firm/Hard R-T</td>
<td>Reliability</td>
</tr>
</tbody>
</table>

A Beautiful Service  (Wolfgang Reisig)

\[
s : I \times S \rightarrow S \times O
\]

or

\[
\begin{align*}
  u & : I \times S \rightarrow O \\
  t & : I \times S \rightarrow S
\end{align*}
\]
And the Ugly Reality \textsuperscript{(Flaviu Cristian*)}

\begin{align*}
u & \subseteq I \times S \times O \\
t & \subseteq I \times S \times S \\
u(i, s) &= \emptyset, \\
u(i, s) &= \{ o, \bot \}, \ldots \\
t(i, s) &= S, \\
t(i, s) &= s, \\
t(i, s) &= \{ s, s' \}, \ldots
\end{align*}

\textsuperscript{*} Flaviu Cristian: Software Fault Tolerance, 1995

The only Real Beauty ? \textsuperscript{(Transactions, Backward recovery)}

\begin{align*}
s & \subseteq I \times S \times S \times O \\
s(i, s) &= \{ ( o, s_o ), ( \bot, s ) \}
\end{align*}

Implement and verify it