Contracts for Web Services

Luca Padovani
http://www.sti.uniurb.it/padovani/

Università degli Studi di Urbino "Carlo Bo"

Outline

Outline

Web services in a nutshell

- distributed processes
- communicating through standard Web protocols (TCP, HTTP, SOAP)
- exchanging data in platform-neutral format (XML)
- dynamically linked
- with machine-understandable self-descriptions

Data contracts Behavioral contracts

Describing data

- XML (eXtensible Markup Language) is the *lingua franca* for inter-platform communication of semi-structured data

```
<order>
  <item>
    <name>PEN</name>
    <quantity>1</quantity>
  </item>
  <item>
    <name>PENCIL</name>
    <quantity>3</quantity>
  </item>
  <address>XYZ</address>
</order>
```

Describing grammars

- XML-Schema (CFGs with restrictions)

```
<element name="order">
  <element name="item"
           minOccurs="0" maxOccurs="unbounded">
    <element name="name" type="string"/>
    <element name="quantity" type="integer"/>
  </element>
  <element name="address" type="string"/>
</element>
```

Behavioral Contracts

Interface descriptions

- `WSDL` 2.0 (W3C recommendation, 2007)

Behavioral descriptions

- `WSCL` 1.0 (W3C note, 2002)
- `WS-BPEL` 2.0 (OASIS standard, 2007)

  "Enabling users to describe business process activities as Web services and define how they can be connected to accomplish specific tasks"

Contracts in WSDL

Focus on the static interface

- Interface = set of operations
- Operation = name + message exchange pattern (MEP)

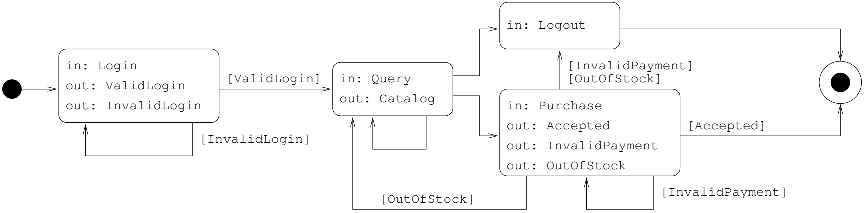
```
<operation name="A" pattern="in-only">
  <input messageLabel="In"/>
</operation>

<operation name="B" pattern="robust-in-only">
  <input messageLabel="In"/>
  <outfault messageLabel="Fault"/>
</operation>
```

Contracts in WSCL

Focus on the dynamic interface

- Conversation = Interactions + Transitions
- Interaction = Types of exchanged messages



Contracts and WS-BPEL

Focus on service structure

- Service = Compositions of Basic Activities

```
<process>
  <sequence>
    <receive operation="Order" variable="Request"/>

    <flow>
      <invoke operation="InStock" inputVariable="Request" outputVariable="InStock"/>
      <invoke operation="Charge" inputVariable="Request" outputVariable="Charge"/>
    </flow>

    <switch>
      <case condition="getVariableData(InStock) == true && getVariableData(Charge) == true)">
        <invoke operation="Ship" inputVariable="Request"/>
        <reply operation="Order" value="OK"/>
      </case>

      <case condition="getVariableData(Charge) == true)">
        <invoke operation="Refund" inputVariable="Request"/>
        <reply operation="Order" value="NO"/>
      </case>

      <otherwise>
        <reply operation="Order" value="NO"/>
      </otherwise>
    </switch>
  </sequence>
</process>
```

The problem of contract equivalence

Web services yellow pages (*registries*)

- `UDDI` 3.0.2 (OASIS standard, 2004)

  "Defining a standard method for enterprises to dynamically discover and invoke Web services"

When are two contracts equivalent?

Outline

Regular expression types + channel schemas

$$T ::= () \mid \mathtt{B} \mid a[T] \mid T, T \mid T + T \mid T^* \mid \mathtt{U} \mid \langle T \rangle^\kappa$$
$$\kappa ::= \mathtt{I} \mid \mathtt{O} \mid \mathtt{IO}$$

Example

```
<element name="order">
  <element name="item"
           minOccurs="0" maxOccurs="unbounded">
    <element name="name" type="string"/>
    <element name="quantity" type="integer"/>
  </element>
  <element name="address" type="string"/>
</element>
```

$\mathtt{order}[\mathtt{item}[\mathtt{name}[\mathtt{string}], \mathtt{quantity}[\mathtt{integer}]]^*, \mathtt{address}[\mathtt{string}]]$

The subschema relation

Intuition

$$S <: T \iff [\![S]\!] \subseteq [\![T]\!]$$

Examples

- $T <: T + S$
- $a[\texttt{integer} + \texttt{string}] <: a[\texttt{integer}] + a[\texttt{string}]$

Channel schemas

- $\langle T \rangle^{\texttt{I}} <: \langle S \rangle^{\texttt{I}} \iff T <: S$
- $\langle T \rangle^{\texttt{O}} <: \langle S \rangle^{\texttt{O}} \iff S <: T$

Complexity matters

Incoming messages are checked against schemas

- checking that a plain XML document (without channel values) x belongs to a schema S can be done in linear time (w.r.t. x's size)
- checking that a channel u belongs to a schema $\langle T \rangle^{\kappa}$ entails computing the subschema relation

How **hard** is it to compute the subschema relation?

The subschema relation is **exponential**

The hard case is the sequence

$$a[S], S' <: \sum_{i \in I} a_i[T_i], T_i'$$

Restriction: label-determinedness

$$i \neq j \Rightarrow a_i \neq a_j \qquad (C_i \cap C_j = \emptyset)$$

Under this restriction, the subschema relation is **polynomial**

References

$$\texttt{PiDuce} = \pi\text{-calculus} + \text{join patterns} + \texttt{XML}$$

- `http://www.cs.unibo.it/PiDuce/`
- Carpineti, Laneve, "*A Basic Contract Language for Web Services*", ESOP 2006.
- Laneve, Padovani, "*Smooth Orchestrators*", FoSSaCS 2006.
- Carpineti, Laneve, Padovani, "*PiDuce – a project for experimenting Web services technologies*", Science of Computer Programming 2009.

Outline

Finding Web services by contract

Compliance = client's satisfaction

$$\rho \dashv \sigma$$

Running a query using *compliance*

$$\mathcal{Q}(\rho) = \{\sigma \in \mathtt{Registry} \mid \rho \dashv \sigma\}$$

Running a query using *duality* $\rho^\perp$ and *subcontract* $\sigma \preceq \tau$

$$\mathcal{Q}(\rho) = \{\sigma \in \mathtt{Registry} \mid \rho^\perp \preceq \sigma\}$$

Finding Web services by contract

Compliance = client's satisfaction

$$\rho \dashv \sigma$$

Running a query using *compliance*

$$\mathcal{Q}(\rho) = \{\sigma \in \texttt{Registry} \mid \rho \dashv \sigma\}$$

Running a query using *duality* $\rho^{\perp}$ and *subcontract* $\sigma \preceq \tau$

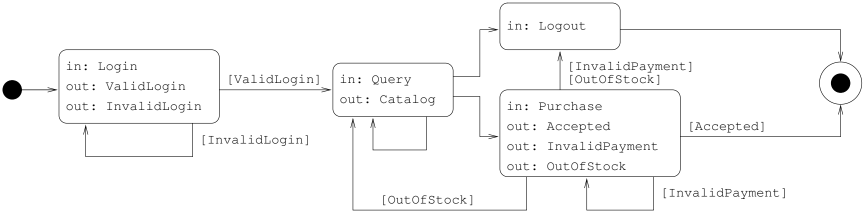
$$\mathcal{Q}(\rho) = \{\sigma \in \texttt{Registry} \mid \rho^{\perp} \preceq \sigma\}$$

Contracts

Syntax

$$\sigma ::= \mathbf{0} \mid a.\sigma \mid \overline{a}.\sigma \mid \sigma + \sigma \mid \sigma \oplus \sigma$$

Example



$$\begin{array}{rcl} \sigma & \stackrel{\text{def}}{=} & \texttt{Login}.(\overline{\texttt{ValidLogin}}.\sigma_1 \oplus \overline{\texttt{InvalidLogin}}.\sigma_2) \\ \sigma_1 & = & \texttt{Query}.\overline{\texttt{Catalog}}.(\sigma_1 + \texttt{Logout}.\mathbf{0} + \texttt{Purchase}\dots) \end{array}$$

Compliance

$$\rho \dashv \sigma \stackrel{\text{def}}{\iff} \rho \parallel \sigma \Longrightarrow \rho' \parallel \sigma' \longrightarrow\!\!\!\!\!\!/\;\; \text{implies } \rho' \xrightarrow{\mathsf{e}}$$

Examples

$$\overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e} \quad \dashv \textbf{?} \quad a + b$$

$$\overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e} \quad \dashv \quad a \oplus b$$

$$\mathbf{0} \quad \dashv \quad \sigma$$

Compliance

$$\rho \dashv \sigma \stackrel{\text{def}}{\iff} \rho \parallel \sigma \Longrightarrow \rho' \parallel \sigma' \not\longrightarrow \text{ implies } \rho' \xrightarrow{\mathsf{e}}$$

Examples

$$\overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e} \quad \dashv \quad a + b \qquad ☺$$

$$\overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e} \quad \dashv \textbf{?} \quad a \oplus b$$

$$\mathbf{0} \quad \dashv \quad \sigma$$

Compliance

$$\rho \dashv \sigma \stackrel{\text{def}}{\iff} \rho \parallel \sigma \Longrightarrow \rho' \parallel \sigma' \not\longrightarrow \text{ implies } \rho' \xrightarrow{\mathsf{e}}$$

Examples

$$\overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e} \quad \dashv \quad a + b \qquad ☺$$

$$\overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e} \quad \dashv \textbf{?} \quad a \oplus b \qquad ☹$$

$$\mathbf{0} \quad \dashv \textbf{?} \quad \sigma$$

Compliance

$$\rho \dashv \sigma \stackrel{\text{def}}{\iff} \rho \parallel \sigma \Longrightarrow \rho' \parallel \sigma' \not\longrightarrow \text{ implies } \rho' \xrightarrow{\mathsf{e}}$$

Examples

$\overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e}$	$\dashv$	$a + b$	☺
$\overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e}$	$\dashv$ **?**	$a \oplus b$	☹
$\mathbf{0}$	$\dashv$ **?**	σ	☹

Subcontract, formally

Set-theoretic interpretation of contracts

$$[\![\sigma]\!]^{\mathsf{s}} \stackrel{\text{def}}{=} \{\rho \mid \rho \dashv \sigma\}$$

Subcontract

$$\sigma \sqsubseteq \tau \stackrel{\text{def}}{\iff} [\![\sigma]\!]^{\mathsf{s}} \subseteq [\![\tau]\!]^{\mathsf{s}}$$

$$\simeq \; = \; \sqsubseteq \cap \sqsupseteq$$

Déjà vu?

- testing framework!

Properties of strong subcontract

Proposition

- ☺ $\sigma \oplus \tau \sqsubseteq \sigma$ (*cf.* *must* preorder)
- ☹ $\sigma \not\sqsubseteq \sigma + \tau$
- ☺ $\sqsubseteq$ is a precongruence

Consequences

- ☺ nice axiomatization
- ☹ cannot be used for *extending* services
- ☺ can be used for safe replacement of parts of services

Not all failures are equal (i.e., there is hope!)

Failure due to client nondeterminism

$$\overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e} \not\dashv a$$

Failure due to service nondeterminism

$$a.\mathsf{e} \not\dashv \overline{a} \oplus \overline{b}$$

Failure due to "system" nondeterminism

$$\overline{a}.\mathsf{e} + b.c.\mathsf{e} \not\dashv a + \overline{b}.\overline{d}$$

Not all failures are equal (i.e., there is hope!)

Failure due to client nondeterminism

$$\overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e} \not\dashv a$$

Failure due to service nondeterminism

$$a.\mathsf{e} \not\dashv \overline{a} \oplus \overline{b}$$

Failure due to “system” nondeterminism

$$\overline{a}.\mathsf{e} + b.c.\mathsf{e} \not\dashv a + \overline{b}.\overline{d}$$

Not all failures are equal (i.e., there is hope!)

Failure due to client nondeterminism

$$\overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e} \not\dashv a$$

Failure due to service nondeterminism

$$a.\mathsf{e} \not\dashv \overline{a} \oplus \overline{b}$$

Failure due to "system" nondeterminism

$$\overline{a}.\mathsf{e} + b.c.\mathsf{e} \not\dashv a + \overline{b}.\overline{d}$$

Idea: orchestrated interaction

client	$\leftrightarrows$	orchestrator	$\leftrightarrows$	service
$\overline{a}$	$\rightarrow$	$\langle a, \overline{a} \rangle$	$\rightarrow$	a
a	$\leftarrow$	$\langle \overline{a}, a \rangle$	$\leftarrow$	$\overline{a}$
$\overline{a}$				a
$\overline{a}$		$\langle a, \overline{a} \rangle$		

Idea: orchestrated interaction

client	$\leftrightarrows$	orchestrator	$\leftrightarrows$	service
$\overline{a}$	$\rightarrow$	$\langle a, \overline{a} \rangle$	$\rightarrow$	a
a	$\leftarrow$	$\langle \overline{a}, a \rangle$	$\leftarrow$	$\overline{a}$
$\overline{a}$				a
$\overline{a}$		$\langle a, \overline{a} \rangle$		

Idea: orchestrated interaction

client	$\leftrightarrows$	orchestrator	$\leftrightarrows$	service
$\overline{a}$	$\rightarrow$	$\langle a, \overline{a} \rangle$	$\rightarrow$	a
a	$\leftarrow$	$\langle \overline{a}, a \rangle$	$\leftarrow$	$\overline{a}$
$\overline{a}$				a
$\overline{a}$		$\langle a, \overline{a} \rangle$		

Idea: orchestrated interaction

client	$\leftrightarrows$	orchestrator	$\leftrightarrows$	service
$\overline{a}$	$\rightarrow$	$\langle a, \overline{a} \rangle$	$\rightarrow$	a
a	$\leftarrow$	$\langle \overline{a}, a \rangle$	$\leftarrow$	$\overline{a}$
$\overline{a}$				a
$\overline{a}$		$\langle a, \overline{a} \rangle$		

Idea: orchestrated interaction

client	$\leftrightarrows$	orchestrator	$\leftrightarrows$	service
$\overline{a}$	$\rightarrow$	$\langle a, \overline{a} \rangle$	$\rightarrow$	a
a	$\leftarrow$	$\langle \overline{a}, a \rangle$	$\leftarrow$	$\overline{a}$
$\overline{a}$				a
$\overline{a}$		$\langle a, \overline{a} \rangle$		

Weak compliance

$$f : \rho \dashv\!\!\!\vdots\; \sigma \stackrel{\text{def}}{\iff} \rho \parallel_f \sigma \Longrightarrow \rho' \parallel_{f'} \sigma' \not\longrightarrow \text{ implies } \rho' \xrightarrow{\mathsf{e}}$$

Examples

$$\langle a, \overline{a} \rangle \vee \langle b, \overline{b} \rangle \quad : \quad \overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e} \quad \dashv\!\!\!\vdots\; ? \quad a + b$$

$$\mathbf{0} \quad : \quad \mathsf{e} + a.b.\mathsf{e} \quad \dashv\!\!\!\vdots \quad \overline{a}$$

$$f \quad : \quad \overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e} \quad \dashv\!\!\!\vdots \quad a \oplus b$$

Weak compliance

$$f : \rho \dashv\!\!\!\models \sigma \stackrel{\text{def}}{\iff} \rho \parallel_f \sigma \Longrightarrow \rho' \parallel_{f'} \sigma' \longrightarrow\!\!\!\!\!\!\!/\;\; \text{implies } \rho' \xrightarrow{\mathsf{e}}$$

Examples

$$\langle a, \overline{a} \rangle \vee \langle b, \overline{b} \rangle \quad : \quad \overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e} \quad \dashv\!\!\!\models \quad a + b \qquad ☺$$

$$\mathbf{0} \quad : \quad \mathsf{e} + a.b.\mathsf{e} \quad \dashv\!\!\!\models \textbf{?} \quad \overline{a}$$

$$f \quad : \quad \overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e} \quad \dashv\!\!\!\models \quad a \oplus b$$

Weak compliance

$$f : \rho \dashv\!\mid \sigma \stackrel{\text{def}}{\iff} \rho \parallel_f \sigma \Longrightarrow \rho' \parallel_{f'} \sigma' \not\longrightarrow \text{ implies } \rho' \xrightarrow{\mathsf{e}}$$

Examples

$\langle a, \overline{a}\rangle \vee \langle b, \overline{b}\rangle$	:	$\overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e}$	$\dashv\!\mid$	$a + b$	☺
$\mathbf{0}$	:	$\mathsf{e} + a.b.\mathsf{e}$	$\dashv\!\mid$	$\overline{a}$	☺
f	:	$\overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e}$	$\dashv\!\mid$ **?**	$a \oplus b$	

Weak compliance

$$f : \rho \dashv\!\!\!\models \sigma \stackrel{\text{def}}{\iff} \rho \parallel_f \sigma \Longrightarrow \rho' \parallel_{f'} \sigma' \not\longrightarrow \text{ implies } \rho' \xrightarrow{\mathsf{e}}$$

Examples

$\langle a, \overline{a} \rangle \vee \langle b, \overline{b} \rangle$	:	$\overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e}$	$\dashv\!\!\!\models$	$a + b$	☺
$\mathbf{0}$	:	$\mathsf{e} + a.b.\mathsf{e}$	$\dashv\!\!\!\models$	$\overline{a}$	☺
f	:	$\overline{a}.\mathsf{e} \oplus \overline{b}.\mathsf{e}$	$\dashv\!\!\!\models$ **?**	$a \oplus b$	☹

Weak subcontract, formally

Set-theoretic interpretation of contracts

$$[\![\sigma]\!]^{\mathsf{w}} \stackrel{\text{def}}{=} \{\rho \mid \exists f : f : \rho \dashv\!\mid \sigma\}$$

Weak subcontract

$$\sigma \preceq \tau \stackrel{\text{def}}{\iff} [\![\sigma]\!]^{\mathsf{s}} \subseteq [\![\tau]\!]^{\mathsf{w}}$$

A few doubts. . .

- is $\preceq$ the subcontract relation we're looking for?
- is $\preceq$ a preorder?

Universal orchestrators

$$\sigma \preceq \tau \iff \forall\rho : (\rho \dashv \sigma \text{ implies } \exists f : f : \rho \dashv\!\!\mid \tau)$$

Universal orchestrator

$$\sigma \preceq \tau \overset{?}{\iff} \exists f : (\forall\rho : \rho \dashv \sigma \text{ implies } f : \rho \dashv\!\!\mid \tau)$$

f is the *universal orchestrator* for $\sigma \preceq \tau$

Proposition (existence of universal orchestrator)

$\sigma \preceq \tau$ if and only if $f : \sigma \preceq \tau$ for some orchestrator f

Universal orchestrators

$$\sigma \preceq \tau \iff \forall \rho : (\rho \dashv \sigma \text{ implies } \exists f : f : \rho \dashv\!\!\mid \tau)$$

Universal orchestrator

$$\sigma \preceq \tau \overset{?}{\iff} \exists f : (\forall \rho : \rho \dashv \sigma \text{ implies } f : \rho \dashv\!\!\mid \tau)$$

f is the *universal orchestrator* for $\sigma \preceq \tau$

Proposition (existence of universal orchestrator)

$\sigma \preceq \tau$ if and only if $f : \sigma \preceq \tau$ for some orchestrator f

Universal orchestrators

$$\sigma \preceq \tau \iff \forall \rho : (\rho \dashv \sigma \text{ implies } \exists f : f : \rho \dashv\!\!\!\!\!\!\; \vdots \; \tau)$$

Universal orchestrator

$$\sigma \preceq \tau \overset{?}{\iff} \exists f : (\forall \rho : \rho \dashv \sigma \text{ implies } f : \rho \dashv\!\!\!\!\!\!\; \vdots \; \tau)$$

f is the *universal orchestrator* for $\sigma \preceq \tau$

Proposition (existence of universal orchestrator)

$\sigma \preceq \tau$ *if and only if* $f : \sigma \preceq \tau$ *for some orchestrator* f

Orchestrators as **morphisms**

Orchestrator application $f(\sigma)$

f	σ	$f(\sigma)$
$\langle a, \overline{a}\rangle \vee \langle b, \overline{b}\rangle$	$a \oplus b$	$a \oplus b$
$\langle a, \overline{a}\rangle$	$a + b$	a
$\langle a, \overline{a}\rangle$	$a \oplus b$	$a \oplus \mathbf{0}$

Theorem

1. $f : \sigma \preceq \tau$ *if and only if* $\sigma \sqsubseteq f(\tau)$
2. $\sigma \sqsubseteq \tau$ *implies* $f(\sigma) \sqsubseteq f(\tau)$

$\preceq$ is a preorder

$\preceq$ is reflexive

Proof.

For every σ there exists I_σ such that $\sigma \simeq I_\sigma(\sigma)$. $\square$

$\preceq$ is transitive

Proof.

For every σ, f, g there exists $f \cdot g$ s.t. $f(g(\sigma)) \simeq (f \cdot g)(\sigma)$. $\square$

Properties of weak subcontract

Proposition

- ☺ $\sqsubseteq \subseteq \preceq$
- ☺ $a \preceq a + b$ (width extension)
- ☺ $\mathbf{0} \preceq \sigma$ (depth extension)
- ☹ $\preceq$ is not a precongruence w.r.t. $+$ and $\oplus$

But...

- ☺ $f : \sigma_1 \preceq \tau_1$ and $f : \sigma_2 \preceq \tau_2$ implies $f : \sigma_1 + \sigma_2 \preceq \tau_1 + \tau_2$ (and similarly for $\oplus$)

Consequences

- ☺ nice proof system
- ☺ algorithm to synthesize "best" orchestrator

Interpretations of orchestrators

As **mediators**

$$\rho \parallel_f \sigma$$

As **morphisms**/**behavioral coercions**

$$f : \sigma \preceq \tau \qquad\qquad f : \tau \rightarrow \sigma$$

As **assumptions** on the environment

$$\langle a, \overline{a} \rangle : a \preceq a + b$$

- it is safe to replace a with $a + b$ if no one ever tries to perform $\overline{b}$

Interpretations of orchestrators

As **mediators**

$$\rho \parallel_f \sigma$$

As **morphisms/behavioral coercions**

$$f : \sigma \preceq \tau \qquad\qquad f : \tau \to \sigma$$

As **assumptions** on the environment

$$\langle a, \overline{a} \rangle : a \preceq a + b$$

- it is safe to replace a with $a + b$ if no one ever tries to perform $\overline{b}$

Interpretations of orchestrators

As **mediators**

$$\rho \parallel_f \sigma$$

As **morphisms/behavioral coercions**

$$f : \sigma \preceq \tau \qquad\qquad f : \tau \to \sigma$$

As **assumptions** on the environment

$$\langle a, \overline{a} \rangle : a \preceq a + b$$

- it is safe to replace a with $a + b$ if no one ever tries to perform $\overline{b}$

Buffered orchestrators

client	$\leftrightarrows$	orchestrator	$\leftrightarrows$	service
$\overline{a}$	$\rightarrow$	$\langle a, \varepsilon \rangle$		σ
		$\vdots$		
ρ		$\langle \varepsilon, \overline{a} \rangle$	$\rightarrow$	a
ρ		$\langle \varepsilon, a \rangle$	$\leftarrow$	$\overline{a}$
		$\vdots$		
a	$\leftarrow$	$\langle \overline{a}, \varepsilon \rangle$		σ

Buffered orchestrators

client	$\leftrightarrows$	orchestrator	$\leftrightarrows$	service
$\overline{a}$	$\rightarrow$	$\langle a, \varepsilon \rangle$		σ
		$\vdots$		
ρ		$\langle \varepsilon, \overline{a} \rangle$	$\rightarrow$	a
ρ		$\langle \varepsilon, a \rangle$	$\leftarrow$	$\overline{a}$
		$\vdots$		
a	$\leftarrow$	$\langle \overline{a}, \varepsilon \rangle$		σ

Buffered orchestrators

client	$\leftrightarrows$	orchestrator	$\leftrightarrows$	service
$\bar{a}$	$\rightarrow$	$\langle a, \varepsilon \rangle$		σ
		$\vdots$		
ρ		$\langle \varepsilon, \bar{a} \rangle$	$\rightarrow$	a
ρ		$\langle \varepsilon, a \rangle$	$\leftarrow$	$\bar{a}$
		$\vdots$		
a	$\leftarrow$	$\langle \bar{a}, \varepsilon \rangle$		σ

$\preceq_k$ is a preorder

$\preceq_k$ is reflexive

Proof.

Same as before. $\square$

$\preceq_k$ is transitive

Proof.

~~For every σ, f, g there exists $f \cdot g$ s.t. $f(g(\sigma)) \simeq (f \cdot g)(\sigma)$.~~ $\square$

Proof.

For every σ, f, g there exists $f \cdot g$ s.t. $f(g(\sigma)) \sqsubseteq (f \cdot g)(\sigma)$. $\square$

$\preceq_k$ is a preorder

$\preceq_k$ is reflexive

Proof.

Same as before. $\square$

$\preceq_k$ is transitive

Proof.

~~For every σ, f, g there exists $f \cdot g$ s.t. $f(g(\sigma)) \simeq (f \cdot g)(\sigma)$.~~ $\square$

Proof.

For every σ, f, g there exists $f \cdot g$ s.t. $f(g(\sigma)) \sqsubseteq (f \cdot g)(\sigma)$. $\square$

Properties of weak k-subcontract

Proposition

- ☺ $\sqsubseteq \subseteq \preceq = \preceq_0 \subseteq \preceq_k$
- ☺ $\overline{a}.\overline{b}.\sigma \preceq_1 \overline{b}.\overline{a}.\sigma$
- ☺ $a.\alpha.\sigma \preceq_1 \alpha.a.\sigma$

Open problems

- ? no complete proof system for $\preceq_k$ is known
- ? is it possible to decide $\preceq_k$ for some k?

But...

- ☺ algorithm to synthesize "best" k-orchestrator

An example (© Wil van der Aalst) (1/3)

$$\sigma \stackrel{\text{def}}{=} \mathit{order}.(\mathit{money} + \overline{\mathit{food}}.\mathit{money})$$

$$\rho_1 \stackrel{\text{def}}{=} \overline{\mathit{order}}.\mathit{food}.\overline{\mathit{money}}.\mathsf{e}$$

$$\rho_1^\perp = \mathit{order}.\overline{\mathit{food}}.\mathit{money}$$

$$f_1 = \langle \mathit{order}, \overline{\mathit{order}} \rangle.\langle \overline{\mathit{food}}, \mathit{food} \rangle.\langle \mathit{money}, \overline{\mathit{money}} \rangle$$

An example (© Wil van der Aalst) (1/3)

$$\sigma \stackrel{\text{def}}{=} \mathit{order}.(\mathit{money} + \overline{\mathit{food}}.\mathit{money})$$

$$\rho_1 \stackrel{\text{def}}{=} \overline{\mathit{order}}.\mathit{food}.\overline{\mathit{money}}.\mathsf{e}$$

$$\rho_1^{\perp} = \mathit{order}.\overline{\mathit{food}}.\mathit{money}$$

$$f_1 = \langle \mathit{order}, \overline{\mathit{order}} \rangle.\langle \overline{\mathit{food}}, \mathit{food} \rangle.\langle \mathit{money}, \overline{\mathit{money}} \rangle$$

An example (© Wil van der Aalst) (1/3)

$$\sigma \stackrel{\text{def}}{=} \mathit{order}.(\mathit{money} + \overline{\mathit{food}}.\mathit{money})$$

$$\rho_1 \stackrel{\text{def}}{=} \overline{\mathit{order}}.\mathit{food}.\overline{\mathit{money}}.\mathsf{e}$$

$$\rho_1^\perp = \mathit{order}.\overline{\mathit{food}}.\mathit{money}$$

$$f_1 = \langle \mathit{order}, \overline{\mathit{order}} \rangle.\langle \overline{\mathit{food}}, \mathit{food} \rangle.\langle \mathit{money}, \overline{\mathit{money}} \rangle$$

An example (2/3)

$$\sigma \stackrel{\text{def}}{=} \mathit{order}.(\mathit{money} + \overline{\mathit{food}}.\mathit{money})$$

$$\rho_2 \stackrel{\text{def}}{=} \overline{\mathit{order}}.(\mathit{food}.\overline{\mathit{money}}.\mathsf{e} + \overline{\mathit{money}}.\mathit{food}.\mathsf{e})$$

$$\rho_2^\perp = \mathit{order}.(\overline{\mathit{food}}.\mathit{money} \oplus \mathit{money}.\overline{\mathit{food}})$$

$$f_2 = \langle \mathit{order}, \overline{\mathit{order}} \rangle.\langle \overline{\mathit{food}}, \mathit{food} \rangle.\langle \mathit{money}, \overline{\mathit{money}} \rangle$$

An example (2/3)

$$
\begin{aligned}
\sigma &\stackrel{\text{def}}{=} \mathit{order}.(\mathit{money} + \overline{\mathit{food}}.\mathit{money}) \\
\rho_2 &\stackrel{\text{def}}{=} \overline{\mathit{order}}.(\mathit{food}.\overline{\mathit{money}}.\mathsf{e} + \overline{\mathit{money}}.\mathit{food}.\mathsf{e}) \\
\rho_2^\perp &= \mathit{order}.(\overline{\mathit{food}}.\mathit{money} \oplus \mathit{money}.\overline{\mathit{food}}) \\
f_2 &= \langle \mathit{order}, \overline{\mathit{order}} \rangle.\langle \overline{\mathit{food}}, \mathit{food} \rangle.\langle \mathit{money}, \overline{\mathit{money}} \rangle
\end{aligned}
$$

An example (2/3)

$$
\begin{array}{rcl}
\sigma & \stackrel{\mathrm{def}}{=} & \mathit{order}.(\mathit{money} + \overline{\mathit{food}}.\mathit{money}) \\
\rho_2 & \stackrel{\mathrm{def}}{=} & \overline{\mathit{order}}.(\mathit{food}.\overline{\mathit{money}}.\mathsf{e} + \overline{\mathit{money}}.\mathit{food}.\mathsf{e}) \\
\rho_2^{\perp} & = & \mathit{order}.(\overline{\mathit{food}}.\mathit{money} \oplus \mathit{money}.\overline{\mathit{food}}) \\
f_2 & = & \langle \mathit{order}, \overline{\mathit{order}} \rangle.\langle \overline{\mathit{food}}, \mathit{food} \rangle.\langle \mathit{money}, \overline{\mathit{money}} \rangle
\end{array}
$$

An example (3/3)

$$\sigma \stackrel{\text{def}}{=} \mathit{order}.(\mathit{money} + \overline{\mathit{food}}.\mathit{money})$$

$$\rho_3 \stackrel{\text{def}}{=} \overline{\mathit{order}}.\overline{\mathit{money}}.\mathit{food}.\mathsf{e}$$

$$\rho_3^\perp = \mathit{order}.\mathit{money}.\overline{\mathit{food}}$$

$$f_3 = \langle \mathit{order}, \overline{\mathit{order}} \rangle.\langle \mathit{money}, \varepsilon \rangle.\langle \overline{\mathit{food}}, \mathit{food} \rangle.\langle \varepsilon, \overline{\mathit{money}} \rangle$$

An example (3/3)

$$\sigma \stackrel{\text{def}}{=} \mathit{order}.(\mathit{money} + \overline{\mathit{food}}.\mathit{money})$$

$$\rho_3 \stackrel{\text{def}}{=} \overline{\mathit{order}}.\overline{\mathit{money}}.\mathit{food}.\mathsf{e}$$

$$\rho_3^\perp = \mathit{order}.\mathit{money}.\overline{\mathit{food}}$$

$$f_3 = \langle \mathit{order}, \overline{\mathit{order}} \rangle.\langle \mathit{money}, \varepsilon \rangle.\langle \overline{\mathit{food}}, \mathit{food} \rangle.\langle \varepsilon, \overline{\mathit{money}} \rangle$$

An example (3/3)

$$\sigma \stackrel{\text{def}}{=} \mathit{order}.(\mathit{money} + \overline{\mathit{food}}.\mathit{money})$$

$$\rho_3 \stackrel{\text{def}}{=} \overline{\mathit{order}}.\overline{\mathit{money}}.\mathit{food}.\mathsf{e}$$

$$\rho_3^{\perp} = \mathit{order}.\mathit{money}.\overline{\mathit{food}}$$

$$f_3 = \langle \mathit{order}, \overline{\mathit{order}} \rangle.\langle \mathit{money}, \varepsilon \rangle.\langle \overline{\mathit{food}}, \mathit{food} \rangle.\langle \varepsilon, \overline{\mathit{money}} \rangle$$

References

- Castagna, Gesbert, Padovani, "*A Theory of Contracts for Web Services*", POPL 2008
- Padovani, "*Contract-directed Synthesis of Simple Orchestrators*", CONCUR 2008.
- Castagna, Gesbert, Padovani, "*A Theory of Contracts for Web Services*", ACM Transactions on Programming Languages and Systems (TOPLAS) 2009.

Outline

Variations on the theme

- Bernardo, Padovani, "*Performance-Oriented Comparison of Web Services via Client-Specific Testing Preorders*", FMOODS 2007.
- Laneve, Padovani, "*The Must Preorder Revisited – An Algebraic Theory for Web Services Contracts*", CONCUR 2007.
- Castagna, Padovani, "*Contracts for Mobile Processes*", CONCUR 2009.

Behavioral Contracts and Session Types

- Laneve, Padovani, "*The Pairing of Contracts and Session Types*", Concurrency, Graphs and Models 2008.
- Padovani, "*Session Types at the Mirror*", ICE 2009.
- Castagna, Dezani-Ciancaglini, Giachino, Padovani, "*Foundations of Session Types*", PPDP 2009.