Contract-directed synthesis of simple orchestrators

Luca Padovani

University of Urbino

CONCUR 2008

Web services in a nutshell

- distributed processes
- communicating through standard Web protocols (tcp, http, soap)
- exchanging data in platform-neutral format (xml)
- **self-describing** (behavioral contracts)

Web services yellow pages (*registries*)

- UDDI (OASIS standard, 2004)

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

Finding Web services by contract

Compliance = client's satisfaction

$$\rho \dashv \sigma$$

Running a query with *compliance*

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

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

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

The quest for $\preceq$

Desired properties of $\preceq$

- reduction of nondeterminism ($a \oplus b \preceq a$)
- extension of functionalities ($a \preceq a + b$)
- some permutation of messages ($a.c \preceq c.a$)

The problem

- *reduction* alone is too strict
- *extension* is unsafe
- *extension;reduction* is not transitive
- *permutation* is not allowed

Idea

- use (simple) orchestrators

Summary

1. contracts
2. *simple orchestrators*
3. subcontract with orchestration
4. orchestrator synthesis

A language for contracts – CCS without τ's

Syntax

$$\sigma \quad ::= \quad \mathtt{0} \quad | \quad \alpha.\sigma \quad | \quad \sigma + \sigma \quad | \quad \sigma \oplus \sigma$$

Examples

- $\mathtt{Number}.\mathtt{Number}.(\mathtt{Add}.\overline{\mathtt{Number}} + \mathtt{Divide}.\overline{\mathtt{Number}})$
- $\mathtt{Login}.(\overline{\mathtt{OK}} \oplus \overline{\mathtt{Invalid}})$

Semantics

$$\alpha.\sigma \xrightarrow{\alpha} \sigma \qquad \sigma \oplus \tau \longrightarrow \sigma \qquad \frac{\sigma \xrightarrow{\alpha} \sigma'}{\sigma + \tau \xrightarrow{\alpha} \sigma'} \qquad \frac{\sigma \longrightarrow \sigma'}{\sigma + \tau \longrightarrow \sigma' + \tau}$$

Same transition relation as CCS without τ's

$$a + (b \oplus c) \longrightarrow a + b$$

Compliance = graceful termination

Client/service interaction

$$\frac{\rho \longrightarrow \rho'}{\rho \parallel \sigma \longrightarrow \rho' \parallel \sigma} \qquad \frac{\sigma \longrightarrow \sigma'}{\rho \parallel \sigma \longrightarrow \rho \parallel \sigma'} \qquad \frac{\rho \xrightarrow{\alpha} \rho' \quad \sigma \xrightarrow{\overline{\alpha}} \sigma'}{\rho \parallel \sigma \longrightarrow \rho' \parallel \sigma'}$$

Compliance

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

Examples

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

Subcontract relation

$$\sigma \sqsubseteq \tau \quad \stackrel{\text{def}}{\iff} \quad \rho \dashv \sigma \text{ implies } \rho \dashv \tau$$

	$a \oplus b \sqsubseteq a$		
$\overline{a}.\mathsf{e} + \overline{c}.\mathsf{e} + \overline{b}$	$a \oplus c \not\sqsubseteq (a \oplus c) + b$		$\langle a, \overline{a} \rangle \vee \langle c, \overline{c} \rangle$
$\overline{a}.(\mathsf{e} + \overline{b})$	$a \not\sqsubseteq a.b$		$\langle a, \overline{a} \rangle$
$\overline{a}.\overline{c}.b.\mathsf{e}$	$a.c.\overline{b} \not\sqsubseteq c.a.\overline{b}$		$\langle a, \varepsilon \rangle.\langle c, \varepsilon \rangle.\langle \varepsilon, \overline{c} \rangle.\langle \varepsilon, \overline{a} \rangle.\langle b, \overline{b} \rangle$

Simple orchestrators

Orchestration actions

$$\mu \quad ::= \quad \langle \alpha, \varepsilon \rangle \quad | \quad \langle \varepsilon, \alpha \rangle \quad | \quad \langle \alpha, \overline{\alpha} \rangle$$

Syntax

$$f \quad ::= \quad 0 \quad | \quad \mu.f \quad | \quad f \vee f$$

Semantics

$$\begin{array}{rcl} [\![0]\!] & = & \{\varepsilon\} \\ [\![\mu.f]\!] & = & \{\varepsilon\} \cup \{\mu s \mid s \in [\![f]\!]\} \\ [\![f \vee g]\!] & = & [\![f]\!] \cup [\![g]\!] \end{array}$$

$$f \stackrel{\mu}{\longmapsto} g \quad \stackrel{\text{def}}{\iff} \quad \{s \mid \mu s \in [\![f]\!]\} = [\![g]\!]$$

Simple orchestrators: validity constraints

$\langle \overline{a}, \varepsilon \rangle$	NO	absurd
$\langle a, \varepsilon \rangle . \langle a, \varepsilon \rangle . \langle a, \varepsilon \rangle \dots$	NO	not bounded
$\langle a, \varepsilon \rangle . \langle \overline{a}, \varepsilon \rangle$	NO	not directional
$\langle a, \overline{a} \rangle$	OK	
$\langle a, \varepsilon \rangle . \langle \varepsilon, \overline{a} \rangle$	OK	

Fact

*Valid orchestrators are **fair** and **finite-state***

Weak compliance = **assisted** graceful termination

Assisted client/service interaction

$$\frac{\rho \longrightarrow \rho'}{\rho \parallel_f \sigma \longrightarrow \rho' \parallel_f \sigma} \qquad \frac{\sigma \longrightarrow \sigma'}{\rho \parallel_f \sigma \longrightarrow \rho \parallel_f \sigma'} \qquad \frac{\rho \xrightarrow{\overline{\alpha}} \rho' \quad f \xmapsto{\langle \alpha, \overline{\alpha} \rangle} f' \quad \sigma \xrightarrow{\alpha} \sigma'}{\rho \parallel_f \sigma \longrightarrow \rho' \parallel_{f'} \sigma'}$$

$$\frac{\rho \xrightarrow{\overline{\alpha}} \rho' \quad f \xmapsto{\langle \alpha, \varepsilon \rangle} f'}{\rho \parallel_f \sigma \longrightarrow \rho' \parallel_{f'} \sigma} \qquad \frac{f \xmapsto{\langle \varepsilon, \overline{\alpha} \rangle} f' \quad \sigma \xrightarrow{\alpha} \sigma'}{\rho \parallel_f \sigma \longrightarrow \rho \parallel_{f'} \sigma'}$$

Weak compliance

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

Examples

- $\langle a, \overline{a} \rangle \vee \langle c, \overline{c} \rangle : \overline{a}.\mathsf{e} + \overline{c}.\mathsf{e} + \overline{b} \dashv\!\!\| (a \oplus c) + b$
- $\langle a, \overline{a} \rangle : \overline{a}.\mathsf{e} \not\dashv\!\!\| a \oplus c$

Weak subcontract relation

$$\sigma \preceq \tau \quad \stackrel{\text{def}}{\iff} \quad \rho \dashv \sigma \text{ implies } f : \rho \dashv\!\!\!\mid \tau \text{ for some } f$$

Universal orchestrator

$$f : \sigma \preceq \tau \quad \stackrel{\text{def}}{\iff} \quad \rho \dashv \sigma \text{ implies } f : \rho \dashv\!\!\!\mid \tau$$

Proposition

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

$$\left.\begin{array}{r} \rho \dashv \sigma \\ \rho' \dashv \sigma \end{array}\right\} \Rightarrow \rho \oplus \rho' \dashv \sigma \Rightarrow f : \rho \oplus \rho' \dashv\!\!\!\mid \tau \Rightarrow \left\{\begin{array}{l} f : \rho \dashv\!\!\!\mid \tau \\ f : \rho' \dashv\!\!\!\mid \tau \end{array}\right.$$

Consequences

- f can be cached in the registry
- orchestrators as morphisms: $f : \tau \rightarrow \sigma$

Orchestrators as morphisms

$$\langle a, \overline{a}\rangle \vee \langle c, \overline{c}\rangle \quad : \quad a \oplus c \preceq (a \oplus c) + b$$

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

$$\langle a, \varepsilon\rangle.\langle c, \varepsilon\rangle.\langle \varepsilon, \overline{c}\rangle.\langle \varepsilon, \overline{a}\rangle.\langle b, \overline{b}\rangle \quad : \quad a.c.\overline{b} \preceq c.a.\overline{b}$$

$$f : \rho \dashv\!\!\| \, \sigma \qquad \Rightarrow \qquad \sigma \xrightarrow{f} f(\sigma) \qquad \rho \dashv f(\sigma)$$

Theorem

$f : \sigma \preceq \tau$ *if and only if* $\sigma \sqsubseteq f(\tau)$

Is $\preceq$ transitive?

$$f : \sigma \preceq \tau \quad \stackrel{\text{def}}{\iff} \quad \rho \dashv \sigma \text{ implies } f : \rho \dashv\!\!\| \tau$$

$$\left.\begin{array}{r} f : \sigma \preceq \tau \\ g : \tau \preceq \sigma' \end{array}\right\} \Rightarrow \left.\begin{array}{c} \sigma \sqsubseteq f(\tau) \\ \tau \sqsubseteq g(\sigma') \end{array}\right\} \Rightarrow \sigma \sqsubseteq f(\tau) \sqsubseteq f(g(\sigma'))$$

$\preceq$ is transitive if $f \circ g$ is an orchestrator

$$\begin{array}{rcl} f & \stackrel{\text{def}}{=} & \langle a, \varepsilon\rangle.\langle c, \varepsilon\rangle.(\langle \varepsilon, \overline{a}\rangle.\langle \overline{b}, b\rangle \vee \langle \varepsilon, \overline{c}\rangle.\langle \overline{d}, d\rangle) \\ g & \stackrel{\text{def}}{=} & \langle a, \varepsilon\rangle.\langle \overline{b}, b\rangle \vee \langle c, \varepsilon\rangle.\langle \overline{d}, d\rangle \end{array}$$

$$f(g(\overline{b} + \overline{d})) \simeq f(a.\overline{b} + c.\overline{d}) \simeq a.c.(\overline{b} \oplus \overline{d})$$

Fact

There is no h such that $h : \overline{b} + \overline{d} \to a.c.(\overline{b} \oplus \overline{d})$

Transitivity of $\preceq$

$$\left.\begin{array}{r} f : \sigma \preceq \tau \\ g : \tau \preceq \sigma' \end{array}\right\} \Rightarrow \left.\begin{array}{c} \sigma \sqsubseteq f(\tau) \\ \tau \sqsubseteq g(\sigma') \end{array}\right\} \Rightarrow \sigma \sqsubseteq f(\tau) \sqsubseteq f(g(\sigma')) \sqsubseteq h(\sigma')$$

It suffices to find h such that $f(g(\sigma')) \sqsubseteq h(\sigma')$

$$\begin{array}{rcl} f & \stackrel{\text{def}}{=} & \langle a, \varepsilon\rangle.\langle c, \varepsilon\rangle.(\langle \varepsilon, \overline{a}\rangle.\langle \overline{b}, b\rangle \vee \langle \varepsilon, \overline{c}\rangle.\langle \overline{d}, d\rangle) \\ g & \stackrel{\text{def}}{=} & \langle a, \varepsilon\rangle.\langle \overline{b}, b\rangle \vee \langle c, \varepsilon\rangle.\langle \overline{d}, d\rangle \\ f \cdot g & = & \langle a, \varepsilon\rangle.\langle c, \varepsilon\rangle.(\langle \overline{b}, b\rangle \vee \langle \overline{d}, d\rangle) \end{array}$$

Theorem

$f(g(\sigma)) \sqsubseteq (f \cdot g)(\sigma)$

Deciding $\sigma \preceq \tau$

The algorithm

$$\frac{\mathsf{A}_r = \{\langle \varphi, \overline{\varphi}' \rangle \mid \sigma \stackrel{\varphi}{\Longrightarrow}, \tau \stackrel{\varphi'}{\Longrightarrow}, \mathbb{B} \vdash \langle \varphi, \overline{\varphi}' \rangle\} \\ \mathsf{A} = \{\langle \varphi, \overline{\varphi}' \rangle \in \mathsf{A}_r \mid \mathbb{B}\langle \varphi, \overline{\varphi}' \rangle \vdash f_{\langle \varphi, \overline{\varphi}' \rangle} : \sigma(\varphi) \blacktriangleleft \tau(\varphi')\} \quad \mathcal{P}(\sigma, \mathsf{A}, \tau)}{\mathbb{B} \vdash \bigvee_{\mu \in \mathsf{A}} \mu.f_\mu : \sigma \blacktriangleleft \tau}$$

Theorem

1. *(correctness)* $\emptyset \vdash \sigma \blacktriangleleft \tau$ *implies* $\sigma \preceq \tau$
2. *(completeness)* $f : \sigma \preceq \tau$ *implies* $\emptyset \vdash g : \sigma \blacktriangleleft \tau$ *and* $f \leqslant g$

Wrap-up

Subcontract relation

- tool for *searching* and *reasoning about* services by their contracts (= behavioral types)
- $\preceq$ combines reduction, extension, and permutation into a single preorder
- $\preceq$ gives safe substitution of services modulo orchestration
- $\preceq$ is decidable

Simple orchestrators

- have nice properties (universality, compositionality)
- can be automatically synthesized

Related work

Testing semantics

- CCS without τ's (De Nicola, Hennessy 1984)

Type theory

- explicit coercions
- type isomorphisms (Di Cosmo 1995)

Future/ongoing work

- deduction system
  - elegant for synchronous orchestrators (Castagna, Gesbert, Padovani 2008)
  - asynchrony axioms are clear

$$a.\alpha.\sigma \preceq \alpha.a.\sigma \qquad \alpha.\overline{a}.\sigma \preceq \overline{a}.\alpha.\sigma$$

  - ...but they interact badly with $+$
- complexity
  - practical analysis
  - algorithm improvements?
- higher-order

Thank you.

Pure synchronous orchestrators

$$\mathcal{I}(\sigma) \stackrel{\text{def}}{=} \bigvee_{\sigma \stackrel{\alpha}{\Longrightarrow} \sigma'} \langle \alpha, \overline{\alpha} \rangle . \mathcal{I}(\sigma')$$

Proposition

$f : \sigma \preceq \tau$ *and* $\mathcal{I}(\tau) \leqslant f$ *implies* $\sigma \sqsubseteq \tau$

$$\langle a, \overline{a} \rangle : a \oplus b \preceq a \qquad\qquad a \oplus b \sqsubseteq a$$

Proposition

$f : \sigma \preceq \tau$ *and* $\rho \dashv \sigma$ *and* $\overline{\mathcal{I}(\rho)} \leqslant f$ *implies* $\rho \dashv \tau$

$$\langle a, \overline{a} \rangle : a \preceq a + b \qquad\qquad \overline{a}.\mathsf{e} \dashv a \qquad\qquad \overline{a}.\mathsf{e} \dashv a + b$$

Pure asynchronous orchestrators

Can orchestrators be implemented as CCS processes?

$$f : \rho \dashv\!\!\| \sigma \quad \stackrel{?}{\iff} \quad C_f[\rho] \dashv \sigma$$

Pure asynchronous orchestrators **can**

$$f : \rho \dashv\!\!\| \sigma \quad \iff \quad (\rho[a \mapsto a'; \dots] \mid \mathcal{M}(f)) \setminus \{a', \dots\} \dashv \sigma$$

$$\mathcal{M}(f) \stackrel{\text{def}}{=} \sum_{f \stackrel{\langle \alpha, \varepsilon \rangle}{\longmapsto} g} \alpha'.\mathcal{M}(g) + \sum_{f \stackrel{\langle \varepsilon, \alpha \rangle}{\longmapsto} g} \alpha.\mathcal{M}(g)$$

Example

$$\langle a, \varepsilon \rangle.\langle c, \varepsilon \rangle.\langle \varepsilon, \overline{c} \rangle.\langle \varepsilon, \overline{a} \rangle.\langle \varepsilon, b \rangle.\langle \overline{b}, \varepsilon \rangle : \overline{a}.\overline{c}.b.\mathsf{e} \dashv\!\!\| c.a.\overline{b}$$

$$(\overline{a}'.\overline{c}'.b'.\mathsf{e} \mid a'.c'.\overline{c}.\overline{a}.b.\overline{b}') \setminus \{a', b', c'\} \dashv c.a.\overline{b}$$

Orchestrators as morphisms (part 2 of 2)

Proposition

1. $\sigma \sqsubseteq \tau$ *implies* $f(\sigma) \sqsubseteq f(\tau)$
2. $f(\sigma) + f(\tau) \sqsubseteq f(\sigma + \tau)$
3. $f(\sigma) \oplus f(\tau) \sqsubseteq f(\sigma \oplus \tau)$

$$\left.\begin{array}{l} f : \sigma \preceq \sigma' \\ f : \tau \preceq \tau' \end{array}\right\} \Rightarrow \left.\begin{array}{c} \sigma \sqsubseteq f(\sigma') \\ \tau \sqsubseteq f(\tau') \end{array}\right\} \Rightarrow \sigma + \tau \sqsubseteq f(\sigma' + \tau') \Rightarrow f : \sigma + \tau \preceq \sigma' + \tau'$$

$\preceq$ is *not* a precongruence

$$a \preceq a + b.c \qquad\qquad a + b.d \not\preceq a.b + b.c + b.d$$

An example: dining philosophers (part 1 of 2)

$$P_i \stackrel{\text{def}}{=} \mathit{fork}_i.\mathit{fork}_i.\overline{\mathit{thought}}.\overline{\mathit{fork}}.\overline{\mathit{fork}}$$

$$C \stackrel{\text{def}}{=} \sum_{i=1..2} \overline{\mathit{fork}}_i . \sum_{i=1..2} \overline{\mathit{fork}}_i.\mathit{thought}.\mathit{fork}.\mathit{fork}$$

$$C \not\dashv P_1 \mid P_2$$

$$f : C \dashv\!\!\| P_1 \mid P_2$$

$$f \stackrel{\text{def}}{=} \bigvee_{i=1..2} \langle \mathit{fork}_i, \overline{\mathit{fork}_i} \rangle . \langle \mathit{fork}_i, \overline{\mathit{fork}_i} \rangle . \langle \mathit{thought}, \overline{\mathit{thought}} \rangle . \langle \mathit{fork}, \overline{\mathit{fork}} \rangle . \langle \mathit{fork}, \overline{\mathit{fork}} \rangle$$

An example: dining philosophers (part 2 of 2)

$$P_i \stackrel{\text{def}}{=} fork_i.fork_i.\overline{thought}.\overline{fork}.\overline{fork}$$
$$Q_i \stackrel{\text{def}}{=} fork_i.fork_i.\overline{fork}.\overline{fork}.\overline{thought}$$

$$g : P_1 \mid P_2 \preceq Q_1 \mid Q_2$$

$$g \stackrel{\text{def}}{=} \bigvee_{i=1..2} \langle fork_i, \overline{fork_i}\rangle . \bigvee_{i=1..2} \langle fork_i, \overline{fork_i}\rangle . \langle \varepsilon, fork\rangle . \langle \varepsilon, fork\rangle . \langle thought, \overline{thought}\rangle . \langle \overline{fork}, \varepsilon\rangle . \langle \overline{fork}, \varepsilon\rangle$$

$$f \cdot g : C \dashv\!\| \; Q_1 \mid Q_2$$