A Formal Account of Contracts for Web Services

Samuele Carpineti, Giuseppe Castagna, Cosimo Laneve, Luca Padovani

University of Bologna, University of Urbino, École Normale Supérieure de Paris

15 september 2006

Summary

Part I

- Contracts and technologies for Web Services
- A language of contracts
- Desirable properties of the subcontract relation

Part II

- Subcontract relation and contract compliance
- Contract synthesis and process compliance
- Contract compliance $\Rightarrow$ process compliance

Concluding remarks

Reasoning about compatibility of behavior

Why is it important to formalize the contract of a client or of a service?

Use:

- dynamic discovery
- dynamic composition
- type checking
- debugging
- automatic code generation
- run-time analysis

Focus:

- communication between two parties (no choreography)

Contracts in WSDL

Focus on the static interface:

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

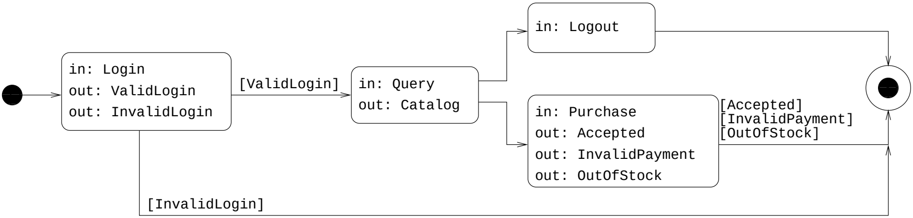
```
<operation name="A"
    pattern="http://www.w3.org/2006/01/wsdl/in-only">
  <input messageLabel="In"/>
</operation>

<operation name="B"
    pattern="http://www.w3.org/2006/01/wsdl/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



+ distinction between internal and external choice
+ possibly cyclic patterns

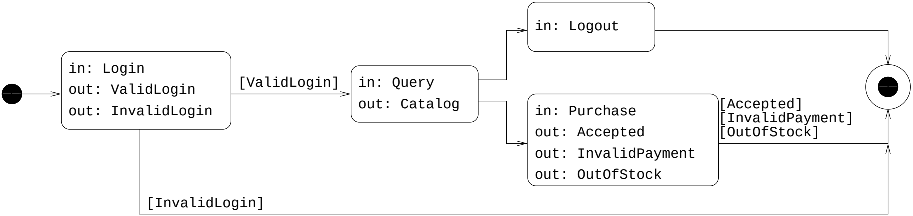
Encoding MEPs into contracts

```
<operation name="A"
    pattern="http://www.w3.org/2006/01/wsdl/in-only">
  <input messageLabel="In"/>
</operation>

<operation name="B"
    pattern="http://www.w3.org/2006/01/wsdl/robust-in-only">
  <input messageLabel="In"/>
  <outfault messageLabel="Fault"/>
</operation>
```

$$
\begin{aligned}
\mathtt{A} &\stackrel{\text{def}}{=} \mathtt{In}.\overline{\mathtt{End}} \\
\mathtt{B} &\stackrel{\text{def}}{=} \mathtt{In}.(\overline{\mathtt{End}} \oplus \overline{\mathtt{Fault}}.\overline{\mathtt{End}})
\end{aligned}
$$

Encoding WSCL into contracts



$$\mathtt{Login}.(\overline{\mathtt{InvalidLogin}}.\overline{\mathtt{End}} \oplus \overline{\mathtt{ValidLogin}}.\mathtt{Query}.\overline{\mathtt{Catalog}}.(\\ \quad \mathtt{Logout}.\overline{\mathtt{End}} + \mathtt{Purchase}.(\\ \qquad \overline{\mathtt{Accepted}}.\overline{\mathtt{End}} \oplus \overline{\mathtt{InvalidPayment}}.\overline{\mathtt{End}} \oplus \overline{\mathtt{OutOfStock}}.\overline{\mathtt{End}})))$$

A formal contract language

contracts	σ	$::=$		
			0	(*void*)
			$\alpha.\sigma$	(*action prefix*)
			$\sigma + \sigma$	(*external choice*)
			$\sigma \oplus \sigma$	(*internal choice*)
actions	α	$::=$		
			a	(*name*)
			$\overline{a}$	(*co-name*)

Names represent types, operations, . . .

c.f. De Nicola, Hennessy, "CCS without τ's", 1984

Comparing contracts: the subcontract relation $\preceq$

σ is a subcontract of σ' if σ' is *more deterministic* than σ

$$a \oplus b \preceq a \qquad a \oplus b \preceq a + b$$

$$\mathtt{In}.(\overline{\mathtt{End}} \oplus \overline{\mathtt{Fault}}.\overline{\mathtt{End}}) \preceq \mathtt{In}.\overline{\mathtt{End}}$$

(*c.f. must pre-order*)

σ is a subcontract of σ' if σ' has *more interacting capabilities* than σ

$$a \preceq a.b \qquad a \preceq a + b \qquad 0 \preceq \sigma$$

$$\mathtt{Logout} + \mathtt{Purchase} \preceq \mathtt{Logout} + \mathtt{Purchase} + \mathtt{BuyLater}$$

($\preceq$ *is different from testing, must, may, ...*)

Summary of the technical part

1. define contract transition and ready sets
2. define subcontract $\preceq$ and contract compliance $\ll$
3. synthesize contracts out of processes
4. define process compliance as “successful interaction”
5. prove that contract compliance implies process compliance

Contracts: transition relation

Interacting party's point of view:

$$a.b + a.c \overset{a}{\longmapsto} b \oplus c$$

$$\alpha.\sigma \overset{\alpha}{\longmapsto} \sigma$$

$$\frac{\sigma_1 \overset{\alpha}{\longmapsto} \sigma_1' \quad \sigma_2 \overset{\alpha}{\longmapsto} \sigma_2'}{\sigma_1 + \sigma_2 \overset{\alpha}{\longmapsto} \sigma_1' \oplus \sigma_2'} \qquad \frac{\sigma_1 \overset{\alpha}{\longmapsto} \sigma_1' \quad \sigma_2 \overset{\alpha}{\not\longmapsto}}{\sigma_1 + \sigma_2 \overset{\alpha}{\longmapsto} \sigma_1'}$$

$$\frac{\sigma_1 \overset{\alpha}{\longmapsto} \sigma_1' \quad \sigma_2 \overset{\alpha}{\longmapsto} \sigma_2'}{\sigma_1 \oplus \sigma_2 \overset{\alpha}{\longmapsto} \sigma_1' \oplus \sigma_2'} \qquad \frac{\sigma_1 \overset{\alpha}{\longmapsto} \sigma_1' \quad \sigma_2 \overset{\alpha}{\not\longmapsto}}{\sigma_1 \oplus \sigma_2 \overset{\alpha}{\longmapsto} \sigma_1'}$$

Contracts: ready sets

$\sigma \Downarrow \mathrm{R}$: the service can be in a state where the actions in R are allowed

$$
\begin{array}{ll}
0 \Downarrow \emptyset & \\
\alpha.\sigma \Downarrow \{\alpha\} & \\
(\sigma + \sigma') \Downarrow \mathrm{R} \cup \mathrm{R}' & \text{if } \sigma \Downarrow \mathrm{R} \text{ and } \sigma' \Downarrow \mathrm{R}' \\
(\sigma \oplus \sigma') \Downarrow \mathrm{R} & \text{if either } \sigma \Downarrow \mathrm{R} \text{ or } \sigma' \Downarrow \mathrm{R}
\end{array}
$$

Example of internal choice:

$$a \oplus b \Downarrow \{a\} \qquad a \oplus b \Downarrow \{b\}$$

Example of external choice:

$$a + b \Downarrow \{a, b\}$$

Subcontract relation

$\preceq$ is the largest relation such that $\sigma_1 \preceq \sigma_2$ implies:

1. if $\sigma_2 \Downarrow \mathrm{R}_2$ then $\sigma_1 \Downarrow \mathrm{R}_1$ with $\mathrm{R}_1 \subseteq \mathrm{R}_2$
2. if $\sigma_1 \stackrel{\alpha}{\longmapsto} \sigma_1'$ and $\sigma_2 \stackrel{\alpha}{\longmapsto} \sigma_2'$ then $\sigma_1' \preceq \sigma_2'$

Key:

1. σ_2 has no more internal states than σ_1 has:

$$a \oplus b \preceq a \qquad a \oplus b \preceq b$$

   and they all allow more capabilities than those in σ_1:

$$a \oplus b \preceq a + b \qquad a \preceq a + b$$

2. if σ_1 and σ_2 share a common action, the continuations are in the subcontract relation:

$$0 \preceq \sigma \qquad a.b \preceq a.b + c$$

Client/service duality and contract compliance

If a client P has contract σ, what is the "cheapest" contract that a service should expose to interact successfully with P?

$$\begin{array}{rcll} a \oplus b & \Rightarrow & \overline{a} + \overline{b} & \\ a + b & \Rightarrow & \overline{a} \oplus \overline{b} & \text{also } \overline{a}\ldots \\ a.b + a.c & \Rightarrow & \overline{a}.\overline{b} \oplus \overline{a}.\overline{c} & \text{NO!} \\ a.b + a.c & \Rightarrow & \overline{a}.(\overline{b} + \overline{c}) & \end{array}$$

The dual contract of σ is defined on σ's normal form:

$$\sigma \simeq \bigoplus_{\sigma \Downarrow \mathrm{R}} \sum_{\sigma \stackrel{\alpha}{\longmapsto} \sigma', \alpha \in \mathrm{R}} \alpha.\sigma'$$

$$\overline{\sigma} \stackrel{\text{def}}{=} \sum_{\sigma \Downarrow \mathrm{R}, \mathrm{R} \neq \emptyset} \bigoplus_{\sigma \stackrel{\alpha}{\longmapsto} \sigma', \alpha \in \mathrm{R}} \overline{\alpha}.\overline{\sigma'}$$

Contract compliance:

$$\sigma \ll \sigma' \stackrel{\text{def}}{=} \overline{\sigma} \preceq \sigma'$$

Simple processes: finite CCS without choice

Syntax:

$$P \; ::= \quad 0 \quad | \quad a.P \quad | \quad \overline{a}.P \quad | \quad P \setminus a \quad | \quad P \mid P$$

Transition relation:

(IN)
$$a.P \xrightarrow{a} P$$

(OUT)
$$\overline{a}.P \xrightarrow{\overline{a}} P$$

(RES)
$$\frac{P \xrightarrow{\mu} Q \quad \mu \notin \{a, \overline{a}\}}{P \setminus a \xrightarrow{\mu} Q \setminus a}$$

(PAR)
$$\frac{P \xrightarrow{\mu} Q}{P \mid R \xrightarrow{\mu} Q \mid R}$$

(COM)
$$\frac{P \xrightarrow{\alpha} P' \quad Q \xrightarrow{\overline{\alpha}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

Process compliance

How do we characterize a "successful interaction" of a system $P \parallel Q$?

System transition:

- if $P \xrightarrow{\tau} P'$ then $P \parallel Q \longrightarrow P' \parallel Q$;
- if $Q \xrightarrow{\tau} Q'$ then $P \parallel Q \longrightarrow P \parallel Q'$;
- if $P \xrightarrow{\alpha} P'$ and $Q \xrightarrow{\overline{\alpha}} Q'$ then $P \parallel Q \longrightarrow P' \parallel Q'$.

P is compliant with Q, notation $P \ll Q$, if either

1. $P \stackrel{\alpha}{\not\longrightarrow}$, or
2. $P \parallel Q \longrightarrow P' \parallel Q'$ implies $P' \ll Q'$

Synthesizing contracts from processes

The type system:

$$\vdash 0 : 0 \qquad \frac{\vdash P : \sigma}{\vdash \alpha.P : \alpha.\sigma} \qquad \frac{\vdash P : \sigma}{\vdash P \setminus a : \sigma \setminus a} \qquad \frac{\vdash P : \sigma \quad \vdash Q : \sigma'}{\vdash P \mid Q : \sigma \mid \sigma'}$$

The $\setminus$ meta-operator behaves like the laws for $\setminus$ in the axiomatization of must/testing pre-orders:

$$\begin{array}{rcll} a.\sigma \setminus a & = & 0 & \\ b.\sigma \setminus a & = & b.(\sigma \setminus a) & \quad a \neq b \end{array}$$

The $\mid$ meta-operator is just the expansion law (in the testing equivalence):

$$\begin{array}{rcl} a \mid b & = & a.b + b.a \\ a \mid \overline{a}.b & = & (a.\overline{a}.b + \overline{a}.(a \mid b) + b) \oplus b \end{array}$$

Contract compliance implies process compliance

Theorem

If $\vdash P : \sigma_1$, $\vdash Q : \sigma_2$, and $\sigma_1 \ll \sigma_2$ then $P \ll Q$

Proof (idea)

- if $P \stackrel{\alpha}{\not\longrightarrow}$ we are done
- if $P \stackrel{\alpha}{\longrightarrow}$ implies $Q \stackrel{\overline{\alpha}}{\not\longrightarrow}$ we have a contradiction: every ready set of $\overline{\sigma_1}$ is not empty hence from $\overline{\sigma_1} \preceq \sigma_2$ we have that P and Q can communicate through a name
- if $P \parallel Q \longrightarrow P' \parallel Q'$ and $\vdash P' : \sigma_1'$ and $\vdash Q' : \sigma_2'$ then $\sigma_1' \ll \sigma_2'$

Open issues

- is $\preceq$ the right compatibility relation?
  - $\preceq$ is *not* transitive

$$a \oplus b.c \preceq a \qquad a \preceq a + b \quad \text{however} \quad a \oplus b.c \not\preceq a + b$$

  - $\preceq$ is *not* a pre-congruence w.r.t. $|$

  $\preceq$ is "good" for searching, not for typing (subsumption)
- $\ll$ is sufficient but not necessary:

$$P \equiv x \mid \overline{x} \qquad Q \equiv 0 \qquad P \ll Q \quad \text{however} \quad (x.\overline{x} + \overline{x}.x) \oplus 0 \not\ll 0$$

  Is $x \mid \overline{x}$ a "meaningful" contract? Is it possible to capture the ability of a client to complete autonomously?
- experiment the effectiveness of contracts in `PiDuce`

Future work

- Recursive contracts
$$\mu x.(a.x + b.x)$$
How do we infer contracts from processes? Syntactic restrictions over processes or regular approximations?
- Name passing:
$$a(x).\overline{x} \qquad \overline{a}(x).x$$
- Adapting $\preceq$ to asynchronous communication
- Relationship with linear logic and denotational semantics of contracts
- Contract isomorphisms and automatic generation of adapters:
$$a.b \iff b.a$$