Configurations of Web Services

Marco Barbosa[1] Luís Barbosa[1]

[1]Departamento de Informática
Universidade do Minho
Braga - Portugal

CIC, 2006

Outline

Two views on Componentware

Introduction

- The OO legacy
    - components as (collections of) classes/objects
    - method invocation as the kernel of component composition
    - resort to middleware intervention to loosen tight-coupling
- The Coordination Paradigm View
    - temporal/spatial decoupling to support a looser inter-component dependency
    - amenable to external control
    - requires anonymous communication

Aims

Aims

- Discuss an orchestration model combining REO-like connectors with behaviourally annotated interfaces
- Configuration = Components + Connectors + Glue code
- Tentative application to model configurations of web-services

Interface

Definition

A web-service S interface is specified by

- a *port signature*, $sig(S)$ over $\mathbb{D}$, given by a port name and a polarity annotation (either in(put) or out(put))
- a *use pattern*, $use(S)$, given by a process term over port names.

Generic Process Algebra

cf. "Process Algebra à la Bird-Merteens" [Bar01, RBB06]

- Processes are inhabitants of a final coalgebra;
- Combinators defined by coinductive extension;
- Interaction discipline: θ

Interaction Structure

- In CCS, $a \theta \overline{a} = \tau$
- In CSP, $a \theta a = a$ for all action $a \in Act$.
- In architectural configurations, ...
  - allow different interaction disciplines to coexist.

Use Patterns and Interaction

- Let $\mathcal{P}$ be a set of port identifiers and S a (the specification of) a web service. Its use pattern $use(S)$ is given by a process expression over $Act = \mathcal{P}(\mathcal{P})$, given by

$$P \ ::= \mathbf{0} \mid \alpha.P \mid P + P \mid P \otimes P \mid P \mathbin{|||} P \mid P; P \mid P \mid P \mid \sigma\, P \mid \text{fix}\ (x \ = \ P)$$

  - choosing $Act = \mathcal{P}(\mathcal{P})$ allows for the synchronous activation of several ports in a single computational step

Use Patterns and Interaction

- All interaction between web services is mediated by a specific connector
- Therefore, if two web services are active their joint behaviour will allow the realization of both use patterns either simultaneously or in an independent way:
- The joint behaviour of a collection $\{S_i | \; i \in n\}$ of ws is

$$use(S_1) \; | \; \ldots \; | \; use(S_n)$$

where the interaction discipline is fixed by $\theta = \cup$.

Examples

$$use(S_1) \;=\; \text{fix } (x \;=\; a.x \;+\; b.x)$$

$$use(S_2) \;=\; \text{fix } (x' \;=\; cd.x'), \quad \text{where, } cd \stackrel{\text{abv}}{=} \{c, d\}$$

$$use(S_1) \mid use(S_2) \;=\; \text{fix } (x \;=\; acd.x + bcd.x + a.x + b.x + cd.x)$$

Services are *coordinated* via Connectors

- What are connectors and how do they compose?
- How do web services' interfaces and connectors interact in a configuration?

Connectors

A connector $\mathbb{C}$ is defined through:

- a relation $\text{data}.[\![\mathbb{C}]\!] : \mathbb{D}^m \longleftarrow \mathbb{D}^n$ which records the flow of data;
- a process expression $\text{port}.[\![\mathbb{C}]\!]$ which gives the pattern of port activation.

Connectors

Basic Connectors

$$\text{data}.[\![\bullet \longmapsto\!\!\!> \bullet]\!] = \text{Id}_{\mathbb{D}}, \quad \text{port}.[\![\bullet \longmapsto\!\!\!> \bullet]\!] = \text{fix}\,(x = ab.x)$$

$$\text{data}.[\![\bullet \overset{\Diamond}{\longmapsto\!\!\!>} \bullet]\!] \subseteq \text{Id}_{\mathbb{D}}, \quad \text{port}.[\![\bullet \overset{\Diamond}{\longmapsto\!\!\!>} \bullet]\!] = \text{fix}\,(x = ab.x + a.x)$$

$$\text{data}.[\![\bullet \overset{\blacktriangledown}{\longmapsto\!\!\!\dashv} \bullet]\!] = \mathbb{D} \times \mathbb{D}, \quad \text{port}.[\![\bullet \overset{\blacktriangledown}{\longmapsto\!\!\!\dashv} \bullet]\!] = \text{fix}\,(x = ab.x)$$

$$\text{data}.[\![\bullet \overset{\triangledown}{\longmapsto\!\!\!\dashv} \bullet]\!] = \mathbb{D} \times \mathbb{D}, \quad \text{port}.[\![\bullet \overset{\triangledown}{\longmapsto\!\!\!\dashv} \bullet]\!] = \text{fix}\,(x = a.x + b.x)$$

$$\text{data}.[\![\bullet \vdash\!\boxdot\!\!> \bullet]\!] = \text{Id}_{\mathbb{D}}, \quad \text{port}.[\![\bullet \vdash\!\boxdot\!\!> \bullet]\!] = \text{fix}\,(x = a.b.x)$$

Connector Combinators

Aggregation

This combinator places its arguments side-by-side, with no direct interaction between them:

$\text{port}.[\![\mathbb{C}_1 \boxtimes \mathbb{C}_2]\!] = \text{port}.[\![\mathbb{C}_1]\!] \mid \text{port}.[\![\mathbb{C}_2]\!] \quad \text{with} \quad \theta = \cup$

Combinators

Hook

Acts as a *feedback* mechanism.

On the *data* side:

Suppose data.$[\![\mathbb{C}]\!] = R : \mathbb{D}^n \longleftarrow \mathbb{D}^m$. Then,

$$R \curvearrowleft^j_i : \mathbb{D}^{n-1} \longleftarrow \mathbb{D}^{m-1}$$

$$t = t_m, \ldots, t_{i+i}, t_{i-i}, \ldots, t_0, \text{ and } t' = t'_n, \ldots, t'_{j+i}, t'_{j-i}, t'_0$$

$$t(R \curvearrowleft^j_i)t' \text{ iff}$$

$$\exists_x.(t_n, \ldots, t_{i+i}, x, t_{i-i}, \ldots, t_0)R(t_m, \ldots, t_{j+i}, x, t_{j-i}, \ldots, t_0)$$

Combinators

Hook

On the behavioural side:

port.$[\![\mathbb{C} \urcorner_i^j]\!]$ is obtained from port.$[\![\mathbb{C}]\!]$, by deleting references to ports i and j.

To be well-formed it is required that i and j appear in different factors of some form of parallel composition ($|||$, $\otimes$, or $|$).

Combinators

Join

Plugs ports with same polarity

- Right Join: $(\mathbb{C}\ {}^{i}_{j}\!> z)$ (non deterministic merger)
- Left Join: $(z <^{i}_{j} \mathbb{C})$ (broadcaster)

At the behavioural level, both operators act as *port renamers*

$$\mathrm{port}.[\![(\mathbb{C}\ {}^{i}_{j}\!> n)]\!] \;=\; \mathrm{port}.[\![(n <^{i}_{j} \mathbb{C})]\!] \;=\; \{n \leftarrow i, n \leftarrow j\}\mathrm{port}.[\![\mathbb{C}]\!]$$

Configurations

Configuration Structure

A configuration involving a collection $S = \{S_i | \ i \in n\}$ of web services is a tuple

$$\langle U, \mathbb{C}, \sigma \rangle$$

where

- $U = use(S_1) \mid use(S_2) \mid \cdots \mid use(S_n)$ is the (joint) use pattern for S
- $\mathbb{C}$ is a connector
- σ a mapping of ports in S to ports in $\mathbb{C}$

Configuration Behaviour

The behaviour $bh(\Gamma)$ of a configuration $\Gamma = \langle U, \mathbb{C}, \sigma \rangle$ is given by

$$bh(\Gamma) \;=\; \sigma\, U \;\otimes\; \mathsf{port}.[\![\mathbb{C}]\!]$$

where θ underlying the $\otimes$ connective is given by

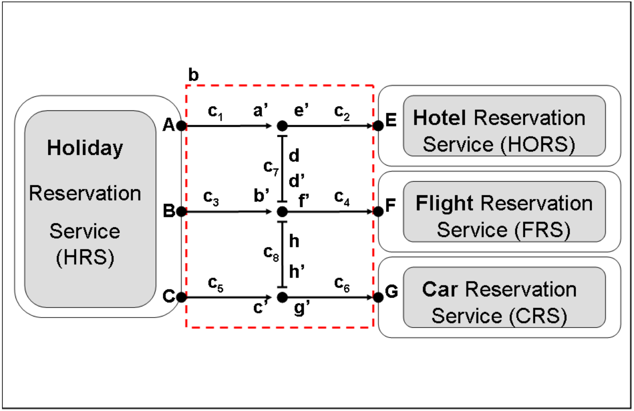
$$c\;\theta\;c' \;=\; \begin{cases} c \;\cap\; (c' \;\cup\; \mathsf{free}) & \Leftarrow\; c' \subseteq c \\ \emptyset & \Leftarrow\; \mathit{otherwise} \end{cases}$$

and free denotes the set of unplugged ports in U, *i.e.*, not in the domain of mapping σ.

Configuration Behaviour: intuitions

- Interaction is achieved by the simultaneous activation of identically named ports
- There is no interaction if the connector intends to activate ports which are not linked to the ones offered by the web services' side.
- The dual situation is allowed, *i.e.*, if the web services' side offers activation of all ports plugged to the ones offered by the connectors' side, their intersection is the resulting interaction.
- Moreover activation of unplugged web services' ports is always possible.

Holiday Reservation (cf. [DA04])



Holiday Reservation

Configuration

$$HR = \langle WHR, SB, \sigma_{HS} \rangle \text{ , where}$$
$$WHR = use(HRS) \mid use(HORS) \mid use(FRS) \mid use(CRS)$$
$$\sigma_{HS} = \{a \leftarrow A, b \leftarrow B, c \leftarrow C, d \leftarrow D, e \leftarrow E, f \leftarrow F, g \leftarrow G\}$$

Usage

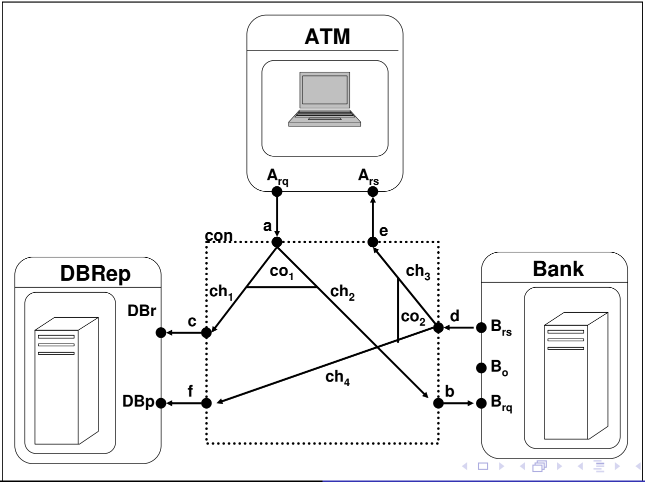
$$use(HRS) = \text{fix } (x = a.x + b.x + c.x + abc.x)$$
$$use(HORS) = \text{fix } (x = e.x)$$
$$use(FRS) = \text{fix } (x = f.x)$$
$$use(CRS) = \text{fix } (x = g.x)$$

Holiday Reservation

Connector

$$\text{port}.[\![c_1]\!] = \text{fix}\ (x = aa'.x),\ \text{port}.[\![c_2]\!] = \text{fix}\ (x = e'e.x),$$
$$\text{port}.[\![c_3]\!] = \text{fix}\ (x = bb'.x),\ \text{port}.[\![c_4]\!] = \text{fix}\ (x = f'f.x),$$
$$\ldots$$
$$Cn_1 = \text{port}.[\![(n <^{e'}_{d} (c_2 \boxtimes c_7))]\!] = \text{fix}\ (x = end'.x)$$
$$Cn_2 = \text{port}.[\![((c_1 \boxtimes Cn_1) \looparrowleft^{n}_{a'})]\!] = \text{fix}\ (x = aed'.x)$$
$$\ldots$$
$$\text{port}.[\![b]\!] = \text{fix}\ (x = abcefg.x), \text{and finally}$$
$$bh(HR) = \text{fix}\ (x = abcefg.x)$$

Bank System


ATM
Arq
Ars
a
e
con
co1
ch1
ch2
ch3
co2
ch4
DBRep
DBr
DBp
c
f
d
b
Bank
Brs
Bo
Brq

Bank System

Configuration

$$BS = \langle WBS,\ DBC,\ \sigma_{BS}\rangle, \text{where}$$
$$WBS = use(ATM) \mid use(Bank) \mid use(DBRep)$$
$$\sigma_{HS} = \{a \leftarrow A_{rq}, e \leftarrow A_{rs}, c \leftarrow DB_r, f \leftarrow DB_p, d \leftarrow B_{rs}, b \leftarrow B_{rq}\}$$

Bank System

Use Patterns

$$
\begin{aligned}
use(ATM) &= \text{fix } (x = a.e.x) \\
use(Bank) &= \text{fix } (y = b.d.y) \\
use(DBRep) &= \text{fix } (z = c.z + f.z)
\end{aligned}
$$

Bank System

Configuration

$$\begin{aligned} \text{port}.[\![DBC]\!] &= \text{port}.[\![(co_1 \boxtimes co_2))]\!] = \\ &\quad \text{fix } (x = abc.x + def.x + abcdef.x) \\ bh(BS) &= \text{fix } (x = abc.def.x) \end{aligned}$$

Conclusions and Future Work

Conclusions

- Formal model for behavioural interfaces and configurations
- Exogenous coordination (cf, REO model)
- Role of *generic process algebra* [Bar01,RBB06]
  (cf, coexistence of different interaction disciplines)

Conclusion and Future Work

Future Work

- Expressing *workflow patterns*
  - **Pattern 2 (Parallel split)**

$$use(\mathrm{WS}_2) \;=\; P_1 \mid \ldots \mid P_n$$

  - **Pattern 3 (Synchronization)**

$$use(\mathrm{WS}_3) \;=\; (a_1.a_2.\ldots.a_n.\S \otimes b_1.b_2.\ldots.b_n.\S)\;;\; P$$

- Is this framework suitable for expressing the semantics of orchestration languages?
- if so, how easily can properties be proved?