Service Composition and Planning

Paolo Traverso
Bruno Kessler Foundation (FBK)
Center for Information Technology – IRST
Trento, Italy
traverso@fbk.eu
http://cit.fbk.eu/traverso
Agenda

- Part I: Introduction to Software Services
- Part II: The Automated Composition Problem
- Part III: An Approach to Service Composition
- Part IV: A New Vision for the Internet of Services
Part I: Services ...

- Services
  - business model from products to services ...
  - services are used, they are not owned
Software Services ...

- **Services**
  - business model from products to services ...
  - services are used, they are not owned

- **Software services**
  - software components that can be used ...
  - ... but are not owned

- **Service-oriented applications**
  - constructed by composing and configuring software services...
  - ... most often provided by “third parties”
  - ... software that is not under control
Key issues in software services

- **Service Level Agreements:**
  - to describe services to the rest of the world

- **Service Composition:**
  - to construct new services

- **Service Monitoring & Adaptation:**
  - to trigger the evolution of services and make them evolve
Service Level Agreements (SLA)

- A description of the interaction flow that is required to interact with the service (e.g., in BPEL)
- A description of the access/authorization permissions (e.g., in SAML)
- A description of the non-functional properties of the service (e.g., with WS-Agreement)
- A description of messages and data that are exchanged (e.g., in WSDL)

Diagram:
- SLA
  - Interface
  - Behaviour
  - QoS
  - Security
Example: Behaviour SLA

The interaction flow for an on-line bank payment (e.g. in BPEL)
1. Service Level Agreements:
   • to describe services to the rest of the world
2. Service Composition:
   • to construct new services
3. Service Monitoring & Adaptation:
   • to trigger the evolution of services and make them evolve
Some work in automated composition

Automata-based e-service coordination:
• Formal framework for composing e-services from behavioral descriptions given in terms of automata.

  Hull, Benedikt, Christophides. PODS 2003

Satisfiability-based e-service coordination:
• decision procedures for satisfiability are used to address the problem of e-coordination

  Berardi, Degiacomo, Mecella, Lenzerini. ICSOC 2004, 2005

QoS-aware service composition:
• genetic algorithms for the optimal QoS estimation

  Canfora, Di Penta et al. ICSOC 2004, ICWS 2005
The Automated Composition Problem

The Automated Composition Problem
Abstract BPEL of the Bank
Abstract BPEL -> State Transition Systems

**Input actions I** (reception of messages)

**Output actions O** (message sent)

**Internal action τ** (internal evolutions that are not visible to external services)

```
BANK
ON-LINE
SERVICE
```
The Automated Composition Problem

THE VIRTUAL ON LINE SHOP COMPOSED SERVICE

ON LINE BANK

ON LINE SHOP

Bank.Fail
Shop. Fail
VOS. Fail

Bank.Succ
Shop.Succ
VOS.succ

request
check id
invalid
enter
cancel
confirm

check call
status
terminated
on-going

send sms

No availability
Confirm req
cancel
confirm
The Approach
Definition. Let $\Sigma_1 = \langle S_1, S_1^0, I, O, R_1 \rangle$ and $\Sigma_2 = \langle S_2, S_2^0, O, I, R_2 \rangle$ be two complementary state transition systems. The **controlled system** $\Sigma_1 \diamond \Sigma_2$ is the STS defined as:

$$\Sigma_1 \diamond \Sigma_2 = \langle S_1 \times S_2, S_1^0 \times S_2^0, \emptyset, \emptyset, R_1 \diamond R_2, \rangle$$

where $\langle (s_1, s_2), \tau, (s'_1, s'_2) \rangle \in (R_1 \diamond R_2)$ if

- $\langle s_1, \tau, s'_1 \rangle \in R_1$ and $s_2 = s'_2$;
- $\langle s_2, \tau, s'_2 \rangle \in R_2$ and $s_1 = s'_1$;
- $\langle s_1, a, s'_1 \rangle \in R_1$ and $\langle s_2, a, s'_2 \rangle \in R_2$ with $a \in I \cup O$. 
The Approach
The Approach: Controlled System

Supply & pay service

Bank

confirm

cancel

confirm

cancel

confirm

confirm req
Deadlock-free composition:

**Definition.** Let $\Sigma_1 = \langle S_1, S_1^0, \mathcal{I}, \mathcal{O}, \mathcal{R}_1 \rangle$ and $\Sigma_2 = \langle S_2, S_2^0, \mathcal{O}, \mathcal{I}, \mathcal{R}_2 \rangle$ be two STS. The controlled system $\Sigma_1 \diamond \Sigma_2$ is said to be **deadlock free** if all states $(s_1, s_2) \in S_1 \times S_2$ satisfy the following conditions:

- if $\langle s_1, a, s_1' \rangle \in \mathcal{R}_1$ with $a \in \mathcal{O}$ then there is some $s_2' \in \tau\text{-closure}(s_2)$ such that $\langle s_2', a, s_2'' \rangle \in \mathcal{R}_2$ for some $s_2'' \in S_2$;
- if $\langle s_2, a, s_2' \rangle \in \mathcal{R}_2$ with $a \in \mathcal{I}$ then there is some $s_1' \in \tau\text{-closure}(s_1)$ such that $\langle s_1', a, s_1'' \rangle \in \mathcal{R}_1$ for some $s_1'' \in S_1$.

$\tau\text{-closure}(s)$ denotes the set of states reachable from $s$ performing transitions labelled by $\tau$. 
Deadlock free composition

Supply & pay service

bank service

Deadlock Free
Deadlock free composition

supply & pay service

bank service

Deadlock
Synthesis: the automated composition problem

Composition of web services:

- Starting from $\Sigma_\parallel = \Sigma_1 \parallel \cdots \parallel \Sigma_n$ and composition requirement $\rho$, find a controller $\Sigma_c$ such that:
  1. requirement $\rho$ is enforced:

\[ \Sigma_c \diamond \Sigma_\parallel \models \rho \]

  2. the asynchronous interaction model is respected:

\[ \Sigma_c \diamond \Sigma_\parallel \text{ is deadlock-free} \]
The Composition Algorithm: Intuitions

- The **Parallel Product** of the State Transitions Systems (STSSs) of Available Interaction Flows (Components + Composed)
- Search the Product STS to satisfy the Composition Requirement
- Find a **subgraph of the Product** STS which satisfies the following conditions (example with reachability conditions):
  1. All terminal states satisfy the condition
  2. If a state belongs to the subgraph, then
     - **a. one outgoing input**
     - **b. all outgoing taus**
     - **c. all outgoing outputs**
       belong to the subgraph
  3. remove non deadlock-free components
- Product STSSs can be extremely large: we use **BDD-based exploration primitives** from the “Planning as Model Checking” framework
The Automatically Generated Executable BPEL
Deployment of Executable BPEL

Tools available at www.fbk.eu
Agenda

- Part I: Introduction to Software Services
- Part II: The Automated Composition Problem
- Part III: An Approach to Service Composition
- Part IV: A New Vision for the Internet of Services
Example: Flight Service

Lufthansa

There's no better way to fly.

Booking | Top Offers | Information & Service | Miles & More

My Account

Check-in

Flight | Hotel | Car

Round-trip | One-way

From | To

Airports | Airports

Depart on | Return on

11.08.2008 | 11.16.2008

Search by price +/- 3 days

Search for specified travel dates

Adults (>12 years) | Children (2-11 years) | Infants (up to 2 years)

1 | 0 | 0

Class | Economy

Search flights

Discover America

Tickets available until 11.11.08.

- Buenos Aires from 649 €
- Caracas from 499 €
- Mexico City from 599 €
- Sao Paulo from 649 €

* Return incl. all taxes and service fee

Online boarding pass

Now you can print out your online boarding pass at home for all flights from Italy to all Lufthansa destinations worldwide

Flights from Milan to Europe

Choose your next destination: book your flight for Europe with special price.

- Barcelona from 99 €
- Brussels from 99 €
- Bucharest from 99 €
- Budapest from 99 €
- Madrid from 99 €
- Paris from 99 €

* return incl. all taxes and service fee

Quicklinks

- Timetable
- Arrival and departure
- Online Check-in
- Miles & More Ora
- Fast Track
- Miles & More registration
- etix® - the electronic ticket
- Reservation for third party
- Passenger Receipt
- Baggage guide
- Newsletter
- SWISS
Example: Flight Service

Flight Service

Book Flight

Flight delay

Your bag is on!

Connecting flights
Example: Service Composition

- Book Flight
- Book Train
- Buy Movie Ticket
Real Services are very different from Software Services, for instance with respect to ...

... **duration**: the time for booking a travel is limited with respect to the duration of the actual travel.

... **dynamic**: software services are static and accessible anywhere and anytime; the real services are dynamic and context dependent.

... **coupling**: software services are independent and loosely coupled; the real services we use are strongly related.
The SOC concepts and approaches have to be re-thought:

- **... monitoring:** from monitoring the execution of software to monitoring the environment where the service operates

- **... adaptation:** from adaptation among software services to adaptation to service and environment changes and to user’s reactions

- **... composition:** from task/goal driven composition of software to a composition based on how a service relates to core assets for the users
Thank you for your attention!