Planning for the Future Internet of Services

Paolo Traverso
Centre for Information Technology - IRST
Bruno Kessler Foundation (FBK)
Trento, Italy
traverso@fbk.eu
Agenda

PART I: Software Services
• Introduction: The Idea
• Planning for Software Services
  • Modeling, Composition, Monitoring, Adaptation

PART II: Future Internet
• Introduction: The Vision
• Planning for the Internet of Services
  • Modeling, Composition, Monitoring, Adaptation
Agenda

PART I: Software Services

• Introduction: The Idea
• Planning for Software Services
  • Modeling, Composition, Monitoring, Adaptation

PART II: Future Internet

• Introduction: The Vision
• Planning for the Internet of Services
  • Modeling, Composition, Monitoring, Adaptation
Services

- business model from products to services...
- services are used, they are not owned
The key to a new generation of software systems
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
Agenda

PART I: Software Services
• Introduction: The Idea
• Planning for Software Services
  • Modeling, Composition, Monitoring, Adaptation

PART II: Future Internet
• Introduction: The Vision
• Planning for the Internet of Services
  • Modeling, Composition, Monitoring, Adaptation
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
Service Level Agreement (SLA)

- formal negotiated agreement between two parties. ...
- contract between customers and their service provider, or between service providers
- it records the common understanding about services, priorities, responsibilities, guarantee, etc.
- For example, it may specify
  - the levels of availability, and performance ...
  - ... and even penalties in the case of violation of the SLA.
Modeling: SLAs

- **SLAs**
  - a description of the non-functional properties of the service (e.g., with *WS-Agreement*)
  - 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 messages and data that are exchanged (e.g., in *WSDL*)

**SLA**

- **QoS**
- **Security**
- **Behaviour**
- **Interface**
Example: Interface in WSDL

```xml
<wsdl:operation name="makeCall">
    <wsdl:input
        message="parlayx_third_party_call:
            ThirdPartyCall_makeCallRequest"/>
    <wsdl:output
        message="parlayx_third_party_call:
            ThirdPartyCall_makeCallResponse"/>
    <wsdl:fault name="ServiceException"
        message="parlayx_common_faults:
            ServiceException"/>
    <wsdl:fault name="PolicyException"
        message="parlayx_common_faults:
            PolicyException"/>
</wsdl:operation>
```
Example: Interface in WSDL

<wsdl:operation name="makeCall">
  <wsdl:input message="parlayx_third_party_call:ThirdPartyCall_makeCallRequest"/>
  <wsdl:output message="parlayx_third_party_call:ThirdPartyCall_makeCallResponse"/>
  <wsdl:fault name="ServiceException" message="parlayx_common_faults:ServiceException"/>
  <wsdl:fault name="PolicyException" message="parlayx_common_faults:PolicyException"/>
</wsdl:operation>

Intrinsic uncertainty, nondeterminism
Service Level Agreements (SLA)

- A description of the exchange of messages and data (e.g., in WSDL).
- A description of the interaction flow required to interact with the service (e.g., in BPEL).
- A description of the non-functional properties of the service (e.g., with WS-Agreement).
- A description of the access/authorization permissions (e.g., in SAML).
Example: Behaviour SLA

The interaction flow for a phone conference service (e.g. in BPEL)
Example: Behaviour SLA

The interaction flow for a phone conference service (e.g. in BPEL)
Example: Behaviour SLA

The interaction flow for an on-line bank payment (e.g. in BPEL)

- **request**
  - **Check**
  - **Availability?**
  - **valid**
  - **invalid**
  - **amount**
  - **No availability**
  - **Confirm req**
  - **cancel**
  - **confirm**
  - **confirm**
Example: Behaviour SLA

The interaction flow for an on-line bank payment (e.g. in BPEL) services are very rarely “atomic”, services are processes that require a flow of interaction.
Agenda

PART I: Software Services
- Introduction: The Idea
- Planning for Software Services
  - Modeling, Composition, Monitoring, Adaptation

PART II: Future Internet
- Introduction: The Vision
- Planning for the Internet of Services
  - Modeling, Composition, Monitoring, Adaptation
The Automated Composition Problem

THE VIRTUAL ON-LINE SHOP COMPOSED SERVICE

ON LINE SHOP

ON LINE BANK

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)
The Automated Composition Problem

THE VIRTUAL ON-LINE SHOP COMPOSED SERVICE

ON LINE SHOP

ON LINE BANK
The Composition Algorithm: Intuitions

- The **Parallel Product** of the State Transitions Systems (STSs) 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 STSs 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.astroproject.org
Agenda

PART I: Software Services
• Introduction: The Idea
• Planning for Software Services
  • Modeling, Composition, Monitoring, Adaptation

PART II: Future Internet
• Introduction: The Vision
• Planning for the Internet of Services
  • Modeling, Composition, Monitoring, Adaptation
A Framework for Monitoring & Adaptation

- Infrastructure
- Monitoring
- Business Processes
- Services

Re-Design
Monitoring of Infrastructure

- Performances
- Available bandwidth
- Resource allocations
Monitoring of Services

- service functions and behaviors
- service availability
- service access
Monitoring of Business Processes

- costs & revenues
- effectiveness of services, e.g., social service
- clients satisfaction

PHONE
CONFERENCE
SERVICE

SLA
“Cross-cutting monitoring”: example

- **Business Process:**
  - Monitor: decreasing revenue
  - Action: decrease price

- **Service:**
  - Monitor: high frequency of cancellations
  - Action: change service

- **Infrastructure:**
  - Monitor: low performances
  - Action: buy more bandwidth
“Cross-cutting monitoring”

- Monitoring must understand at which level the problem is and at which level to react
  - Representation of co-relation among different levels
  - Learning co-relations ("proactive monitoring")
Some monitoring techniques

Assertion-based monitoring:

- Monitors specified as assertions that annotate the BPEL code.

- Annotated BPEL processes are then automatically translated to “monitored processes”

- The approach allows for monitoring of time-outs, runtime errors, behavioral and functional properties.

- “Dynamic monitoring” monitoring rules selected at run-time

Baresi, Ghezzi, Guinea. ICSOC 2004, 2005
Some monitoring techniques (cont.)

Event-based monitoring:

- Monitoring requirements expressed in event-calculus.
- The specified events are observed at run-time and stored in a database.
- Algorithm based on integrity constraint checking analyzes the database

_Mahbub, Spanoudakis, ICSOC 2004, IEEE ICWS 2005_
Assumption-based monitoring:

- Architecture that separates the business logic from monitoring.

- “Instance Monitors” & “Class Monitors”

- Formal language for the specification of instance and class monitors

- Automatic translation of monitor specs to Java programs.

*Barbon, Pistore, Trainotti, Traverso. ICSOC 2005, ICWS 2005*
Inputs:
- Abstract BPEL descriptions of component services
- A SLA specification

Output:
- Notification of violation of SLA ...
- Notification of situations of interest ...
- Aggregated/statistical information

Barbon, Pistore, Trainotti, Traverso. ICSOC 2005, ICWS 2005
Run-Time Monitoring: an example
Run-Time Monitoring: Examples

Check Violations of SLA

The Bank does not refuse the credentials of the Shop
Run-Time Monitoring: Examples

Statistical and performance information

SLA

Count the number of items offered to the Client before the Client accepts to buy

Measure the time requested to finalize the payment

ON LINE SHOP

ON LINE BANK
Run-Time Monitoring: Examples

Properties related to **classes** of processes

The Bank NEVER refuses the credentials of the Store

ON LINE SHOP

SLA

ON LINE BANK

AVERAGE duration for the payment procedure
Architecture for Run-time Monitoring
Process Monitoring Console

### Process Detail

<table>
<thead>
<tr>
<th>ID</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>VOS</td>
</tr>
<tr>
<td>Namespace</td>
<td><a href="http://astroproject.org/BusinessProcesses/VOS">http://astroproject.org/BusinessProcesses/VOS</a></td>
</tr>
<tr>
<td>Started</td>
<td>2006/09/19 05:13:14</td>
</tr>
<tr>
<td>Ended</td>
<td>2006/09/19 05:13:29</td>
</tr>
<tr>
<td>State</td>
<td>Completed</td>
</tr>
</tbody>
</table>

### Monitor Details

<table>
<thead>
<tr>
<th>Monitor</th>
<th>Description</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOS_PaymentTime</td>
<td>Payment time</td>
<td>4901</td>
</tr>
<tr>
<td>VOS_RetryOnSucc</td>
<td>number of retries for a successful buy</td>
<td>0</td>
</tr>
<tr>
<td>VOS_OfferBeforeBank</td>
<td>Bank interaction don't start before user accepts offer</td>
<td>Valid</td>
</tr>
<tr>
<td>VOS_Protocol</td>
<td>Communication protocol compliance</td>
<td>Valid</td>
</tr>
<tr>
<td>VOS_NotAvailCount</td>
<td>Not availability count</td>
<td>0</td>
</tr>
<tr>
<td>VOS_StoreRefuseCc</td>
<td>Bank cannot refuse Shop's payment information</td>
<td>Error: Property StoreRefuseCc violated.</td>
</tr>
</tbody>
</table>

### Log

# Process Monitoring Console

## Process Class Monitor

<table>
<thead>
<tr>
<th>Monitor</th>
<th>Description</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOS_CountStoreRefuseCc</td>
<td>Number of times that store credentials are refused by bank</td>
<td><img src="image" alt="0-10" /></td>
</tr>
<tr>
<td>VOS_AvgUserRetries</td>
<td>Average of how many times user gets offers</td>
<td><img src="image" alt="0-10" /></td>
</tr>
<tr>
<td>VOS_AvgPaymentTime</td>
<td>Average payment time</td>
<td><img src="image" alt="0-30000" /></td>
</tr>
<tr>
<td>VOS_GlobStoreRefuseCc</td>
<td>Store credentials are never refused by bank</td>
<td><img src="image" alt="Yellow Light" /></td>
</tr>
</tbody>
</table>
# Process Class Monitor

<table>
<thead>
<tr>
<th>Monitor</th>
<th>Description</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOS_CountStoreRefuseCcstore credentials are refused by bank</td>
<td>Number of times that VOS_CountStoreRefuseCcstore credentials are refused by bank</td>
<td></td>
</tr>
<tr>
<td>VOS_AvgUserRetries</td>
<td>Average of how many times user gets offers</td>
<td></td>
</tr>
<tr>
<td>VOS_AvgPaymentTime</td>
<td>Average payment time</td>
<td></td>
</tr>
<tr>
<td>VOS_GlobStoreRefuseCc</td>
<td>Store credentials are never refused by bank</td>
<td></td>
</tr>
</tbody>
</table>
# Process Monitoring Console

## Process Class Monitor

<table>
<thead>
<tr>
<th>Monitor</th>
<th>Description</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOS_CountStoreRefuseCc</td>
<td>Number of times that store credentials are refused by bank</td>
<td></td>
</tr>
<tr>
<td>VOS_AvgUserRetries</td>
<td>Average of how many times user gets offers</td>
<td></td>
</tr>
<tr>
<td>VOS_AvgPaymentTime</td>
<td>Average payment time</td>
<td></td>
</tr>
<tr>
<td>VOS_GlobStoreRefuseCc</td>
<td>Store credentials are never refused by bank</td>
<td></td>
</tr>
</tbody>
</table>

---

Copyright © 2004. Active Endpoints, Inc. All Rights Reserved
Agenda

PART I: Software Services
• Introduction: The Idea
• Planning for Software Services
  • Modeling, Composition, Monitoring, Adaptation

PART II: Future Internet
• Introduction: The Vision
• Planning for the Internet of Services
  • Modeling, Composition, Monitoring, Adaptation
Run-Time Adaptation (Autonomic)

- **Infrastructural Level:**
  - modify bandwidth, change resource allocation (e.g., Grids)

- **Service Level:**
  - dynamic binding
  - reconfiguration of composition
  - Adaptation to behavioral changes of component services ...
  - ... by automated composition!
Run-Time Adaptation by composition: example
Run-Time Adaptation by composition: example

PHONE CONFERENCE SERVICE

SMS SERVICE CENTER
Run-Time Adaptation by composition: example
Run-Time Adaptation (Autonomic)

- **Infrastructural Level:**
  - modify bandwidth, change resource allocation (e.g., Grids)

- **Service Level:**
  - dynamic binding
  - reconfiguration of composition
  - Adaptation to behavioral changes of component services ...
  - ... can be autonomic ...
  - ... by re-running the automated composition procedure
Run-Time Adaptation (Autonomic)

- Infrastructural Level:
  - modify bandwidth, change resource allocation (e.g., Grids)

- Service Level:
  - dynamic binding
  - reconfiguration
  - Adaptation of component services... can be autonomic by running the automated composition procedure...
Agenda

PART I: Software Services
• Introduction: The Idea
• Planning for Software Services
  • Modeling, Composition, Monitoring, Adaptation

PART II: Future Internet
• Introduction: The Vision
• Planning for the Internet of Services
  • Modeling, Composition, Monitoring, Adaptation
The Vision

The Internet of Services ...

... is extending today’s Internet to become service-enabled

... will enable agile enterprises to reach out to a global market and focus on core competencies but also create global competition

[Prof.Dr. Lutz Heuser, Vice President SAP Research]

... will allow the permanent, transparent, seamless, and trustworthy provision of services covering all the aspects of user’s life and business.

[FIA WG on Internet of Services]
The Claim

• A **key aspect has been underestimated** so far in the research in “Software and Services”

• Software services are software components that provide electronic access to **“real services”**

• **Characteristics** of real services are **very different** from those of the corresponding software services

• The **“Future Internet of Services”** should focus **on real services**, rather than software services.

• This requires a **research paradigm shift**
Example: Flight Service

Lufthansa

There's no better way to fly.

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

Round-trip | One-way

From | Depart on | To | Return on

11.08.2008 | 11.16.2008

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 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 Oro
- Fast Track
- Miles & More registration
- etix® - the electronic ticket
- Reservation for third party
- Passenger Receipt
- Baggage guide
- Newsletter
- SWISS

* Return including taxes, fees and charges for flights until March 09.
Example: Flight Service

Flight delay

Your bag is on!

Connecting flights

Book Flight
A simple example

- **Book Flight**
- **Book Train**
- **Travel to London**
  - 10:00 – 13:00
  - Business meeting
- **Buy Movie Ticket**
### Monday 3/11

<table>
<thead>
<tr>
<th>Time</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>08:00</td>
<td></td>
</tr>
<tr>
<td>10:00</td>
<td></td>
</tr>
<tr>
<td>11:00</td>
<td></td>
</tr>
<tr>
<td>12:00</td>
<td></td>
</tr>
<tr>
<td>14:30</td>
<td>Travel to London</td>
</tr>
<tr>
<td>15:00</td>
<td></td>
</tr>
<tr>
<td>16:00</td>
<td></td>
</tr>
<tr>
<td>17:00</td>
<td></td>
</tr>
<tr>
<td>18:00</td>
<td></td>
</tr>
<tr>
<td>20:00</td>
<td></td>
</tr>
</tbody>
</table>

### Tuesday 4/11

<table>
<thead>
<tr>
<th>Time</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00</td>
<td>10:00 – 13:00 Business meeting</td>
</tr>
<tr>
<td>14:30</td>
<td>14:30 – 17:00 Flight Verona-Lon...</td>
</tr>
<tr>
<td>16:00</td>
<td>16:00 – 18:30 Flight London-Ver...</td>
</tr>
<tr>
<td>18:00</td>
<td>18:00 – 20:00 Movie “Batman”</td>
</tr>
<tr>
<td>20:00</td>
<td></td>
</tr>
</tbody>
</table>
A simple example

- Book Flight
- Book Train
- Buy Movie Ticket
The Vision (one step forward ...)

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 Vision (one step forward ...)

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
And ... what about internet?
And ... what about internet?
While internet has a minor role for software services ... it provides a convenient infrastructure for publishing, discovering, and executing software components.

Internet is instead a key enabler for “real” services ... it offers a unique capability to ...

- communicate to the user changes in services and in context,
- allow the user to react immediately to this dynamicity

Changes in the services are immediately visible to everybody, similarly to what happens in the Internet of Web pages

Internet as the service delivery platform of the future
Agenda

PART I: Software Services
• Introduction: The Idea
• Planning for Software Services
  • Modeling, Composition, Monitoring, Adaptation

PART II: Future Internet
• Introduction: The Vision
• Planning for the Internet of Services
  • Modeling, Composition, Monitoring, Adaptation
And what about ... Service Modeling?

State of the art:

• A technical description of the functional and non-functional aspects of services

• Interfaces, behavior, quality, security, ....

• A description representing of the technical aspects of software services
And what about ... Semantics?

• “Classical” Semantic Web Services (SWS) allow for:
  • Representing technical aspects of software services (input/output, preconditions, effects)
  • Defining a taxonomy of services.

Example:

```
Food
  Restaurant
    Pizzeria
      PizzaHut

Travel
  Flight
    Lufthansa Flight
      LH 3275

Entertainment
  Movie
    Batman movie
```
We need something different!

- A novel way to model services, based on **key assets for the user**. E.g., in the example they might be key parameters such as time, location, money, social network.

- In the case of *PizzaHut*, I want to know **where** it is, **when** it is open, **how much** does it cost...

- In the case of *my flight*, I want to know **when** (schedule), **where** (airports), **who** travels...

- In the case of *Cinema*, I want to know **when** (movie time), **where** (cinema location), **who** comes with me ...
Example: Personal Mobile Services

- Services are organized along **four core assets** of the user
  - **Time** representing the temporal relation of user activities
  - **Location** representing the spatial relation
  - **Social** representing other parties involved in those activities
  - **Money/Values** representing costs and assets of the user’s activities

- **Well know applications** such as calendars, maps, contact lists, e-wallets can be used to expose services to the user
Example: Personal Mobile Services

- Movie Ticket Poster
- NFC Service
- Trip Advisor Web Service
- Meeting Scheduling Service
- Time
- Location
- Social Network
- Money/Values
- Calendar Application
- Maps / POIs
- Contact Lists
- TelCo Services
- E-Wallet Payment Services
PART I: Software Services
• Introduction: The Idea
• Planning for Software Services
  • Modeling, Composition, Monitoring, Adaptation

PART II: Future Internet
• Introduction: The Vision
• Planning for the Internet of Services
  • Modeling, Composition, Monitoring, Adaptation
System (and project) architecture

Coordinating the execution of external services (composition)
Monitor and filter external events
Manage interactions with the user

Keep track of the use of the resources
Identify and analyze potential inconsistencies
Find ways of resolving these problems

Coordinate the activities on the customer services
Find relations among services (semantics)
Application layer

- Problem: how to keep **services and resources aligned**
- Solution: **objects** are used to model service behaviors and resource allocation
Application layer:
Modeling object/service evolution

Travel Leg

- **createLeg**: CREATED(this)
- **createLegMsg**: CREATED(this)
- **changeLeg**: MODIFIED_RESOURCE(this, resource, value)
- **changeLegMsg**: MODIFIED_RESOURCE(this, resource, value)
- **cancelLeg**: DELETED(this)
- **cancelLegMsg**: DELETED(this)
- **legStarted**: MODIFIED_STATE(this, STARTED)
- **legLost**: DELETED(this)
- **legCompleted**: MODIFIED_STATE(this, COMPLETED)

**service operation invocation**

**external message**

**contextual condition**

© 2008 by Fondazione Bruno Kessler and DoCoMo Communications Laboratories Europe GmbH
Application layer
Object and service coordination

Coordination group

Coordination requirement

© 2008 by Fondazione Bruno Kessler and DoCoMo Communications Laboratories Europe GmbH
The Vision: Personal Mobile Services

State of the art:
“Many Services – Many Applications”

Vision
“Many Services – One Application”
**Life Centered Domains** ...

- user specific needs
- user assets and resources

**Business Centered Domains** ...

- business specific needs
- company’s key assets
Life-centered domains

- The Agenda as the “service composition work-bench” for mobile personal services!
Business-centered domains

- Which is the “service composition workbench” for emergency management services?
Business-centered domain

Each business specific domain requires a specific workbench ...

... emergency management: map for planning & running emergency services

... e-health and social services: citizen personal record

... small medium banks: bank account plan
Conclusions (1)

The Future Internet of Services requires a Research Paradigm Shift in:

**Modeling:**
- technical description of the functional & non-functional aspects
- description of how the use of real services affects consumers

**Monitoring:**
- properties related to the execution of software components
- properties of the physical environment where the real services operate
The Future Internet of Services requires a Research Paradigm Shift in:

**Adaptation:**
- reaction to changes in software services
- reaction to changes in the physical environment where services operate, and to users’ behaviors

**Composition:**
- a software engineering task: “task/goal driven” composition of software components
- composition based on emergent needs, constraint/opportunities of the consumers
Thank you for your attention!