Emerging Web Service Technology

Introduction to Web Services

WS 2009/2010
Mohammed AbuJarour
Tobias Vogel
Nov 12, 2009
<table>
<thead>
<tr>
<th>Name</th>
<th>Topic</th>
<th>Date</th>
<th>Supervisor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan-Felix Schwarz</td>
<td>Enabling business experts to discover web services for business process automation</td>
<td>Dec 3, 2009</td>
<td>Mohammed AbuJarour</td>
</tr>
<tr>
<td>Christoph Thiele</td>
<td>Service selection by choreography-driven matching</td>
<td>Dec 10, 2009</td>
<td>Tobias Vogel</td>
</tr>
<tr>
<td>Abdelfattah Elnaggar</td>
<td>A logic-based approach for service discovery with composition support</td>
<td>Dec 17, 2009</td>
<td>Mohammed AbuJarour</td>
</tr>
<tr>
<td>Martin Lorenz</td>
<td>Composite web services</td>
<td>Jan 7, 2010</td>
<td>Tobias Vogel</td>
</tr>
<tr>
<td>David Jaeger</td>
<td>Model-Driven Performance Evaluation for Service Engineering</td>
<td>Jan 21, 2010</td>
<td>Mohammed AbuJarour</td>
</tr>
<tr>
<td>Fabian Lindenberg</td>
<td>Reputation Propagation in Composite Services</td>
<td>Jan 28, 2010</td>
<td>Mohammed AbuJarour</td>
</tr>
<tr>
<td>Henrik Steudel</td>
<td>Tools for Semantic Web Services</td>
<td>Feb 4, 2010</td>
<td>Tobias Vogel</td>
</tr>
</tbody>
</table>
Contents

- Previously
- WS Standard Stack
- Basic Concepts of Web Services
  - WSDL
  - SOAP
  - Binding
  - UDDI
- Composite Web Services
  - BPEL
- Quality of Service
- Semantic Web Services

<table>
<thead>
<tr>
<th>Area</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Discovery</td>
<td>Enabling business experts to discover web services for business</td>
</tr>
<tr>
<td></td>
<td>process automation</td>
</tr>
<tr>
<td></td>
<td>Service selection by choreography-driven matching</td>
</tr>
<tr>
<td></td>
<td>A logic-based approach for service discovery with composition support</td>
</tr>
<tr>
<td>Service Composition</td>
<td>Composite web services</td>
</tr>
<tr>
<td>Service Management</td>
<td>BPEL-Mora: Lightweight Embeddable Extensible BPEL Engine</td>
</tr>
<tr>
<td>Quality of Service</td>
<td>Model-Driven Performance Evaluation for Service Engineering</td>
</tr>
<tr>
<td></td>
<td>Reputation Propagation in Composite Services</td>
</tr>
<tr>
<td>Semantic Web Services</td>
<td>Tools for Semantic Web Services</td>
</tr>
</tbody>
</table>
Definitions

- **What is a Service?**
  - “The performance of work (a function) by one for another” [4]

- **What is a Web Service?**
  - “A Web Service is a platform-independent programmable module with standard interface descriptions that provide universal accessibility through standard communication protocols” [5]
Triangular SOA Operational Model

Service Registry

Service Consumer

Service Provider

1. Discover
2. Publish
3. Bind
Triangular SOA Operational Model
Triangular SOA Operational Model

- **Service Consumer**
- **Service Registry**
- **Service Provider**

Connections:
- **Discover** from UDDI to Service Registry
- **Publish** from WSDL to Service Registry
- **Bind** from Service Consumer to Service Provider
Triangular SOA Operational Model

Service Consumer

Service Registry

Service Provider

UDDI

Discover

Publish

WSDL

Bind

SOAP
Triangular SOA Operational Model

Previously ...

```
<?xml version="1.0"?>
<definitions name="StockQuoteService"
    targetNamespace="http://www.getquote.com/StockQuoteService"
    xmlns:interface="http://www.getquote.com/StockQuoteService-interface"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/"
    xmlns="http://schemas.xmlsoap.org/soap/">
  <documentation>
    This service provides an implementation of a standard stock quote service. The Web service uses the live stock quote service provided by XMLtoday.com. The XMLtoday.com stock quote service uses an HTTP GET interface to request quote, and returns an XML string as a response.
  </documentation> 
  
  For additional information on how this service obtains stock quotes, go to the XMLtoday.com web site: http://www.xmltoday.com/examples/soap/stock.psp.

  <port name="StockQuoteService-interface" location="http://www.xmltoday.com/StockQuoteService-interface.wsdl">
    <service name="StockQuoteService">
      <documentation>Stock Quote Service Documentation</documentation>
      <documentation>Single Symbol Stock Quote Service</documentation>
      <binding>Interface:SingleSymbolBinding</binding>
      <soap:address location="http://www.getquote.com/stockquoteservice"/>
    </service>
  </port>
</definitions>
```
Previously ...
Triangular SOA Operational Model

Service Consumer

Service Provider

Service Registry

UDDI

WSDL

SOAP

A SOAP request:

```xml
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version='1.0'?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
               soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:Body xmlns:m="http://www.example.org">
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```
Triangular SOA Operational Model

A SOAP request:

```
POST /inStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
    encodingStyle="http://www.w3.org/2001/12/soap-encoding">
    <soap:Body xmlns:m="http://www.example.org/stock">
        <m:getStockPriceResponse>
            <m:StockName>BWM</m:StockName>
            <m:GetStockPriceResponse>
                <m:Price>34.5</m:Price>
            </m:GetStockPriceResponse>
        </m:getStockPriceResponse>
    </soap:Body>
</soap:Envelope>
```

The SOAP response:

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
    encodingStyle="http://www.w3.org/2001/12/soap-encoding">
    <soap:Body xmlns:m="http://www.example.org/stock">
        <m:GetStockPriceResponse>
            <m:Price>34.5</m:Price>
        </m:GetStockPriceResponse>
    </soap:Body>
</soap:Envelope>
```
### Web Services Standards Stack

<table>
<thead>
<tr>
<th>Service Composition</th>
<th>WS-Service Group</th>
<th>WS-Notification</th>
<th>BPEL4WS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality of Service (QoS)</td>
<td>WS-Security</td>
<td>WS-Transaction</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WS-Reliable Messaging</td>
<td>WS-Resource Lifetime</td>
<td></td>
</tr>
<tr>
<td>Description/Publishing/Discovery</td>
<td>WS-Resource Properties</td>
<td>WS-Base Faults</td>
<td>UDDI</td>
</tr>
<tr>
<td></td>
<td>XSD</td>
<td>WSDL</td>
<td>WS-Policy</td>
</tr>
<tr>
<td>Messaging</td>
<td>XML</td>
<td>SOAP</td>
<td>WS-Addressing</td>
</tr>
<tr>
<td>Transports</td>
<td>HTTP/HTTPS</td>
<td>SMTP</td>
<td>RMI / IIOP</td>
</tr>
</tbody>
</table>

Source: Dr. Daniel Sabbah, Vice President of Strategy & Technology, IBM Software Group, Globus World 2004
Web Services Standards Stack

<table>
<thead>
<tr>
<th>Stack Type</th>
<th>Protocols, Formats, Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Composition</td>
<td>WS-Service Group, WS-Notification, BPEL4WS</td>
</tr>
<tr>
<td>Quality of Service (QoS)</td>
<td>WS-Security, WS-Transaction, WS-Reliable Messaging, WS-Resource Lifetime</td>
</tr>
<tr>
<td>Description/Publishing/Discovery</td>
<td>WS-Resource Properties, WS-Base Faults, UDDI, XSD, WSDL, WS-Policy, WS-Metadata Exchange</td>
</tr>
<tr>
<td>Messaging</td>
<td>XML, SOAP, WS-Addressing, WS-Renewable References</td>
</tr>
<tr>
<td>Transports</td>
<td>HTTP/HTTPS, SMTP, RMI / IIOP, JMS</td>
</tr>
</tbody>
</table>

- Stacks are common
- Standardization Organizations for WSs
  - W3C, OASIS, WS-*
  - Protocols, formats, languages
Transport Layer

- Invocation of distributed functionality
- Core communication mechanisms
- Typical protocols
  - HTTP/HTTPS
  - (SMTP)
  - (FTP)
- High compatibility with enterprise IT infrastructures
Messaging Layer

- Format of WS messages
  - Operations
  - Parameters
- Styles
  - SOAP (based on XML)
  - Anything (following REST paradigm)
Connecting different Web Service participants
   - Describe WS capabilities and messages
   - Offer WSs
   - Find WSs
Quality of Service

- Defines non-functional properties
  - Security (Authorization, Authentication, Confidentiality, Integrity)
  - Timing constraints/SLAs
  - Costs
Quality of Service

- Defines non-functional properties
  - Security (Authorization, Authentication, Confidentiality, Integrity)
  - Timing constraints/SLAs
  - Costs
Quality of Service

- Defines non-functional properties
  - Security (Authorization, Authentication, Confidentiality, Integrity)
  - Timing constraints/SLAs
  - Costs
Quality of Service

- Defines non-functional properties
  - Security (Authorization, Authentication, Confidentiality, Integrity)
  - Timing constraints/SLAs
  - Costs
Quality of Service

- Defines non-functional properties
  - Security (Authorization, Authentication, Confidentiality, Integrity)
  - Timing constraints/SLAs
  - Costs

<table>
<thead>
<tr>
<th>Service Composition</th>
<th>WS-Service Group</th>
<th>WS-Notification</th>
<th>BPEL4WS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality of Service (QoS)</td>
<td>WS-Security</td>
<td>WS-Transaction</td>
<td>WS-Resource Lifetime</td>
</tr>
<tr>
<td>WS-Reliable Messaging</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description/Publishing/Discovery</th>
<th>WS-Resource Properties</th>
<th>WS-Base Faults</th>
<th>UDDI</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSD</td>
<td>WSDL</td>
<td>WS-Policy</td>
<td>WS-Metadata Exchange</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Messaging</th>
<th>XML</th>
<th>SOAP</th>
<th>WS-Addressing</th>
<th>WS-Renewable References</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Transports</th>
<th>HTTP/HTTPS</th>
<th>SMTP</th>
<th>RMI / IIOP</th>
<th>JMS</th>
</tr>
</thead>
</table>
Create higher-level service compositions out of existing functionality

- Sequences of message exchanges
- Flow of control (workflow)

Approaches

- Hard-coding
- Tool support/modeling
  - BPEL
  - WS-Notification

Service Composition

Quality of Service (QoS)
- WS-Security
- WS-Transaction
- WS-Reliable Messaging
- WS-Resource Lifetime

Description/Publishing/Discovery
- WS-Resource Properties
- WS-Base Faults
- UDDI
- XSD
- WSDL
- WS-Policy
- WS-Metadata Exchange

Messaging
- XML
- SOAP
- WS-Addressing
- WS-Renewable References

Transports
- HTTP/HTTPS
- SMTP
- RMI / IIOP
- JMS

Weather forecast service
Position service
Sightseeing service
Service Composition

- Create higher-level service compositions out of existing functionality
  - Sequences of message exchanges
  - Flow of control (workflow)

- Approaches
  - Hard-coding
  - Tool support/modeling
    - BPEL
    - WS-Notification
Web Service Description Language
WSDL

- “Web Service Description Language”
- XML-based
- Answers these questions:
  - What is the service about?
  - Where does it reside?
  - How can it be invoked?
- Constructs:

```xml
<message name="getPriceRequest">
  <part name="productid" type="xs:string" />
</message>

<message name="getPriceResponse">
  <part name="value" type="xs:string" />
</message>

<portType name="productPrice">
  <operation name="getPrice">
    <input message="getPriceRequest" />
    <output message="getPriceResponse" />
  </operation>
</portType>
```
<types>
  <schema targetNamespace="http://servicescomputing.org/ProductInfo"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:wsdl="http://servicescomputing.org/wsdl/" >
    <xs:element name="id" type="xsd:string" />
    <xs:element name="name" type="xsd:string" />
    <xs:element name="vendor" type="xsd:string" />
    <xs:complexType name="ProductInfo">
      <xs:sequence>
        <xs:element ref="tns:id" />  
        <xs:element ref="tns:name" />  
        <xs:element ref="tns:vendor" />
      </xs:sequence>
    </xs:complexType>
  </schema>
</types>

<message name="getPriceRequest">
  <part name="productId" type="xs:string" />
</message>

<message name="getPriceResponse">
  <part name="value" type="xs:string" />
</message>

<portType name="productPrice">
  <operation name="getPrice">
    <input message="getPriceRequest" />
    <output message="getPriceResponse" />
  </operation>
</portType>
Web Service Communication Protocol

SOAP

- “Simple Object Access Protocol”
- Structured and typed information exchange
- XML-based
- Bound to transport protocol (HTTP/S, SMTP, etc)
- Interaction patterns:
  - Remote Procedure Call (RPC): Synchronous request/response
  - Document-Oriented: Asynchronous
- SOAP Message Constructs
  - Envelope: required
  - Header: optional
  - Body: required
  - Fault: optional
<?xml version="1.0" ?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
    soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
    <soap:Header>
        <m:Payment xmlns:m="http://www.servicescomputing.org/payment/"
            soap:actor="http://www.servicescomputing.org/appml"
            soap:mustUnderstand="1">
            123
        </m:Payment>
    </soap:Header>
    <soap:Body>
        <!-- Request message-->
        <m:GetPrice xmlns:m="http://www.servicescomputing.org/prices">
            <m:Item> Product </m:Item>
        </m:GetPrice>
        <soap:Fault>
            ....
        </soap:Fault>
    </soap:Body>
</soap:Envelope>
Web Service Communication Protocol

SOAP

```xml
POST /InServicesComputing HTTP/1.1
Host: www.servicescomputing.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 100

<?xml version="1.0" ?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
    soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

    <soap:Header>
        <m:Payment xmlns:m="http://www.servicescomputing.org/payment/"
            soap:actor="http://www.servicescomputing.org/appml"
            soap:mustUnderstand="1">
            123
        </m:Payment>
    </soap:Header>

    <soap:Body>
        <!-- Response message-->
        <m:GetPriceResponse xmlns:m="http://www.servicescomputing.org/prices">
            <m:Price>70</m:Price>
        </m:GetPriceResponse>
    </soap:Body>

    <!-- Request message-->
    <m:GetPrice xmlns:m="http://www.servicescomputing.org/prices">
        <m:Item>Product</m:Item>
    </m:GetPrice>

    <soap:Fault>
        ....
    </soap:Fault>

    </soap:Body>
</soap:Envelope>
```
WSDL-SOAP Binding

```xml
<message name="getPriceRequest">
  <part name="productid" type="xs:string"/>
</message>

<message name="getPriceResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="productPrice">
  <operation name="getPrice">
    <input message="getPriceRequest"/>
    <output message="getPriceResponse"/>
  </operation>
</portType>

<binding type="productPrice" name="b1">  
  <soap:binding style="rpc" transport="http://schema.cmsoap.org/soap/http"/>
  <operation>
    <soap:operation soapAction="http://servicesComputing.org/getPrice"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
```
Publishing a Web Service in Registry

UDDI

- “Universal Description, Discovery, and Integration”
- XML-based

Answers these questions:

- **Who**: information about a business, such as name, contact.
- **What**: classification information about industry, products, registered Web Services
- **Where**: registration information, such as URL
- **How**: registration references about interfaces

```xml
<discoveryURL useType="businessEntity">
  http://www.servicescomputing.org?
  businessKey=uddi:servicescomputing.org:registry:sales:100
</discoveryURL>
```
Complex business logic requires functionality of several WSs.

**Approaches**
- Static vs. dynamic
- Automated vs. manual
- Model driven vs. business rule driven vs. declarative composition

**Business Process Execution Language (BPEL, BPEL4S, WS-BPEL)**
- Language
- Development Environment
- Runtime Environment
- Workflow definition
  - Sequence, flow, pick, ...
- XML based
- 3 sections
- Workflow definition
  - Sequence, flow, pick, ...
- XML based
- 3 sections
  - Partner link definitions
BPEL

- Workflow definition
  - Sequence, flow, pick, ...
- XML based
- 3 sections
  - Partner link definitions
  - Variable definitions

Diagram:

- WS1
- WS2
- WS3a
- WS3b
- Workflow definition
  - Sequence, flow, pick, ...
- XML based
- 3 sections
  - Partner link definitions
  - Variable definitions
BPEL

- Workflow definition
  - Sequence, flow, pick, ...
- XML based
- 3 sections
  - Partner link definitions
  - Variable definitions
  - Process flow definitions
Composite Getaway service
  - Converts the position into address
  - Finds sights near an address
    ➔ Finds sights near the current position
Composite Getaway service
- Converts the position into address
- Finds sights near an address
  ➔ Finds sights near the current position
Composite Getaway service

- Converts the position into address
- Finds sights near an address
  - Finds sights near the current position
Composite Getaway service

- Converts the position into address
- Finds sights near an address
  ➔ Finds sights near the current position
Composite Getaway service

- Converts the position into address
- Finds sights near an address
  - Finds sights near the current position
Composite Getaway service

- Converts the position into address
- Finds sights near an address
  - Finds sights near the current position
BPEL by Example

- Composite Getaway service
  - Converts the position into address
  - Finds sights near an address
    - Finds sights near the current position
WSDLs of Existing Services

```xml
<definitions>
  <portType name="positionPT">
    <operation name="geo2add">
      <input message="geo2addRequest">
      <output message="geo2addResponse">
    </operation></portType>
</definitions>

<definitions>
  <portType name="sightsPT">
    <operation name="sightsAt">
      <input message="sightsAtRequest">
      <output message="sightsAtResponse">
    </operation></portType>
</definitions>
```
WSDLs of Existing Services

<definitions>
  <portType name="positionPT">
    <operation name="geo2add">
      <input message="geo2addRequest">
      <output message="geo2addResponse">
    </operation></portType>
  ...
</definitions>

<definitions>
  <portType name="sightsPT">
    <operation name="sightsAt">
      <input message="sightsAtRequest">
      <output message="sightsAtResponse">
    </operation></portType>
  ...
</definitions>
WSDLs of Existing Services

```xml
<definitions>
  <portType name="positionPT">
    <operation name="geo2add">
      <input message="geo2addRequest">
        <output message="geo2addResponse">
      </operation></portType>...

<definitions>
  <portType name="sightsPT">
    <operation name="sightsAt">
      <input message="sightsAtRequest">
        <output message="sightsAtResponse">
      </operation></portType>...
```
<definitions>
  <message name="closeSightsRequest">...</message>
  <message name="closeSightsResponse">...</message>
  <message name="geo2addRequest">...</message>
  <message name="geo2addResponse">...</message>
  <message name="sightsAtRequest">...</message>
  <message name="sightsAtResponse">...</message>

  <portType name="getawayPT">
    <operation name="closeSights">
      <input message="closeSightsRequest"/>
      <output message="closeSightsResponse"/>
    </operation>
  </portType>

  ...

</definitions>
Getaway Service BPEL file

```xml
<process>
  <partnerLinks>
    <partnerLink
      name="closeSightsPL"
      myRole="getaway" />
    <partnerLink
      name="geo2addPL"
      myRole="addRequestor"
      partnerRole="addService" />
    <partnerLink
      name="sightsAtPL"
      myRole="sightRequestor"
      partnerRole="sightService" />
  </partnerLinks>
  <variables>
    <variable
      name="closeSightsRequest" />
    ... 
    <variable
      name="sightsAtResponse" />
  </variables>
  <sequence>
    <receive />
    <invoke />
    <invoke />
    <reply />
  </sequence>
</process>
```
Getaway Service

Getaway Service

Position Service

Sightseeing Service

<process...
  <partnerLinks...
  <variables...
  <sequence...
</process>
Quality of Service QoS

- Non-functional Properties (NFP)
- Four categories:
  - Security: authentication and authorization of users, message integrity, message encryption
  - Transaction: all-or-nothing
  - Reliable Messaging: deliver messages reliably between Web Services
  - Resource Lifetime Management: immediate and time-based schedule destruction
Combining features from the Semantic Web with Web Service technology

Objective: high degree of automation for
- Description
- Discovery
- Selection
- Invocation

Machine-interpretable meta-information
Semantic Ingredients

- **Ontology**: "Formal, explicit specification of a shared conceptualization" (Tom Gruber, 1993)
  - Machine-readable
  - Unambiguous (no contradictions)
  - Commonly accepted
  - Model of a domain
- **Knowledge Base**
  - Contains objects
- **Resource Description Framework**
  - Knowledge (meta-data) representation format
  - Triples
    - Subject, relation/predicate, object

```prolog
?library.hasBook ?book,
?reader.is_customer ?library.
```
OWL-S

- Web Ontology Language for Web Services
  - Ontology for semantic markup of Web Services

- Service
  - Concept which has to be specified for each annotated Web Service

- ServiceProfile
  - Overall description of the service (discovery)

- ServiceModel
  - Fine-grained specification of the internal functionality of the service (selection)

- ServiceGrounding
  - Information about how to invoke the Web Service (invocation)
Service Profile

<owl:Class id="AirlineTicketing">
    <subClassOf resource="#E_Commerce" />
    <subClassOf>
        <restriction>
            <onProperty resource="#merchandise" />
            <allValuesFrom resource="#CommercialAirlineTravel" />
        </restriction>
    </subClassOf>
</owl:Class>
Service Model

```xml
<process:AtomicProcess id="GetFlightDetails">
  <hasInput resource="#DepartureAirport" />
  <hasInput resource="#ArrivalAirport" />
  <hasInput resource="#OutboundDate" />
  ...
  <hasOutput resource="#FlightsFound" />
</AtomicProcess>

<process:AtomicProcess id="SelectFlight">
  <hasInput resource="#FlightsAvailable" />
  <hasOutput resource="#SelectedFlight" />
</AtomicProcess>
```
A set of standards already evolved to serve realizing Web Services.

- WSDL is an XML-based description language for single Web Services.
- SOAP is an XML-based protocol for service messaging.
- UDDI is an XML-based language for publishing and discovering Web Services.
- BPEL is an XML-based description language for composite Web Services.

Non-functional Properties of Web Services are vital criteria in service selection.

Semantic Web Services aim at automating the process of finding, selecting and consuming Web Services.
Closing Remarks

- Web Services’ related files are create automatically using special tools
- RESTful Web Services are emerging
- ebXML is becoming the successor of UDDI
- DO NOT hesitate to contact us!

Mohammed Abujarour

Contact Information

Ph.D. Student
Hasso-Plattner-Institut for IT Systems Engineering
Prof.-Dr.-Helmert-Str. 2-3
D-14482 Potsdam, Germany

Phone: ++49 331 5509 276
Fax: ++49 331 5509 287
Room: A-1.11
Email: mohammed(dot)abujarour(at)hpi.uni-potsdam.de

Member of the HPI Research School on “Service-Oriented Systems Engineering”

Tobias Vogel

Hasso-Plattner-Institut für Softwaresystemtechnik
Prof.-Dr.-Helmert-Straße 2-3
D-14482 Potsdam, Germany

Telefon: ++49 331 5509 292
Fax: ++49 331 5509 287
Raum: A-1.11
E-Mail: T. Vogel
## Contents

- Previously
- WS Standard Stack
- Basic Concepts of Web Services
  - WSDL
  - SOAP
  - Binding
  - UDDI
- Composite Web Services
  - BPEL
- Quality of Service
- Semantic Web Services

### Area

<table>
<thead>
<tr>
<th>Area</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Discovery</td>
<td>Enabling business experts to discover web services for business process automation</td>
</tr>
<tr>
<td></td>
<td>Service selection by choreography-driven matching</td>
</tr>
<tr>
<td></td>
<td>A logic-based approach for service discovery with composition support</td>
</tr>
<tr>
<td>Service Composition</td>
<td>Composite web services</td>
</tr>
<tr>
<td></td>
<td>Model Driven Design of Web Service Operations using Web Engineering Practices</td>
</tr>
<tr>
<td>Service Management</td>
<td>BPEL-Mora: Lightweight Embeddable Extensible BPEL Engine</td>
</tr>
<tr>
<td>Quality of Service</td>
<td>Model-Driven Performance Evaluation for Service Engineering</td>
</tr>
<tr>
<td></td>
<td>Reputation Propagation in Composite Services</td>
</tr>
<tr>
<td>Semantic Web Services</td>
<td>Tools for Semantic Web Services</td>
</tr>
</tbody>
</table>
The End