Web services and distributed computing

Web Data Management and Distribution

Serge Abiteboul Philippe Rigaux
Marie-Christine Rousset Pierre Senellart

http://gemo.futurs.inria.fr/wdmd

February 5, 2009

Outline

1. Distributed data management
2. The basis: distributed computing
3. Web services
4. Composition of services
5. Web 2.0
6. Active XML

Data of interest is distributed

- In different geographical locations
- On different machines with different operating systems
- Reachable via different networks based on different protocols
- Organized with different logical schemas, different models/formats, using different languages/ontologies

Remark

Provide single-point access to such distributed data

Remark

Use distribution to improve performance of information management systems (response time, availability, reliability)

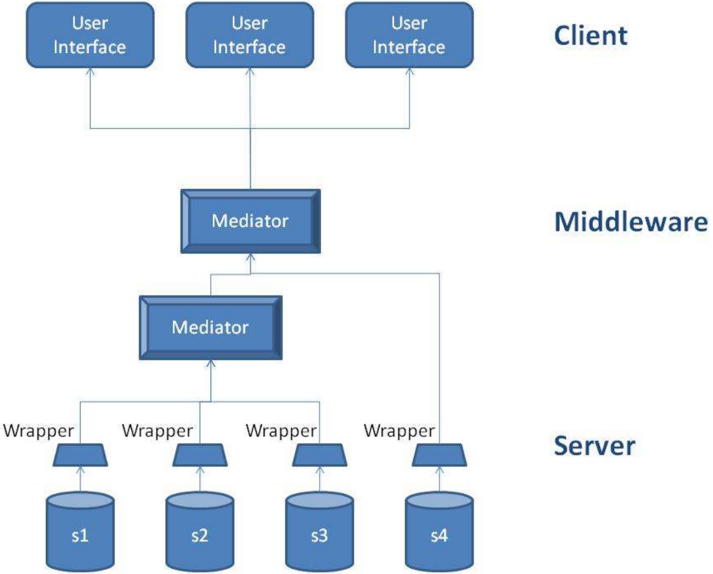
Taxonomy: level of autonomy

- Little: distributed data base systems
  - Few machines, transactions, triggers
  - A bit more autonomy: federation
- Mediator/wrapper architecture
  - Many autonomous publishers: e.g., Web portals
  - The logic of the integration is provided by a mediator
  - Each source is "wrapped" to support a unique protocol

Wrapper

- ETL tools (Extract/Transfer/Load): capable of obtaining data from almost any software tool
  - Documents, mail boxes, files and ldap directories, databases, contacts, calendar, etc.
- Extraction: e.g., HTML to XML
  - Often semiautomatic
  - Machine learning technology
- Data restructuring
- Many difficulties
  - scaling to large number of sources
  - management of inconsistencies
  - resistance to changes in the structure of sources

3-tier architecture



Taxonomy: virtual or data replication

- Warehouse
  - Data is replicated in a warehouse
  - Typical application: On-line analytical processing
  - Query: very efficient
  - Update: need to propagate changes to warehouse or stale data
  - Consistency issues
- Pure mediation
  - Data is virtual in a pure mediator (views)
  - A query on the mediator is rewritten in queries on the sources; the results are combined
  - Typical application: Web portals
  - Query: very expensive (performance issues)
- Combination: E.g., comparative shopping
  - Warehousing for most products
  - Pure mediation for rapidly changing data: airplane tickets, promotions

Outline

1. Distributed data management
2. The basis: distributed computing
3. Web services
4. Composition of services
5. Web 2.0
6. Active XML

History

- RPC: Remote procedure call
- TP monitor: RPC + transaction processing
  - persistance, distributed transactions, logging, error, recovery
- Object brokers: RPC in object-oriented paradigm
- MOM: message-oriented middleware
- Object monitors: Object brokers + transaction

Main concepts: distribution, messaging, transaction & objects

History (2)

- MOM: 60-70 and continuing
- TP monitor & MOM: 60-70 upto today
  - IBM CICS, BEA Tuxedo
- RPC: 80
- Object brokers: 90's
  - Corba (Object Management Group)
  - DCOM (Microsoft)
- XML-RPC: 99
  - XML messages via HTTP-POST

Remote procedure call

- An abstraction that allows interacting with a remote program while ignoring its details
  - send some data as argument
  - activate the program
  - receive some result
- Sockets, TCP-IP, SOAP (for Web services)

Message-Oriented Middleware

- Asynchronous calls
- Locally: Queue of messages
- IBM Websphere, Microsoft MQ serie

Corba

- Common Object Request Broker Architecture
- RPC + Object-oriented paradigm
- Independent of the programming language, e.g., C++ or Java
- A system (an ORB) provides the interoperability
- Support for a large set of services: persistance, transaction, messaging, naming, security, etc.
- Main support for distribution before Web services

Outline

1 Distributed data management

2 The basis: distributed computing

3 Web services
- Generalities
- SOAP
- WSDL
- UDDI
- Security

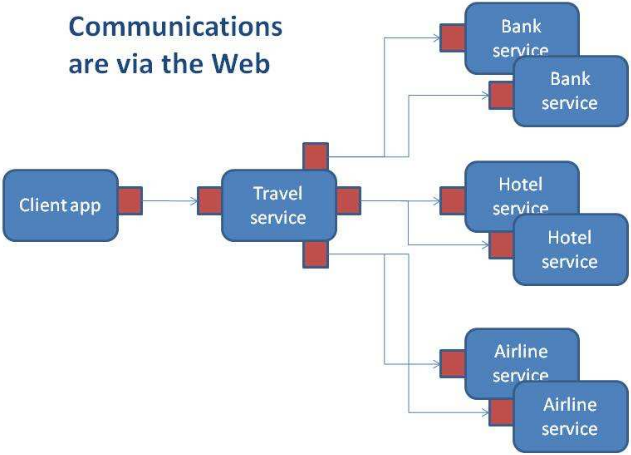
4 Composition of services

5 Web 2.0

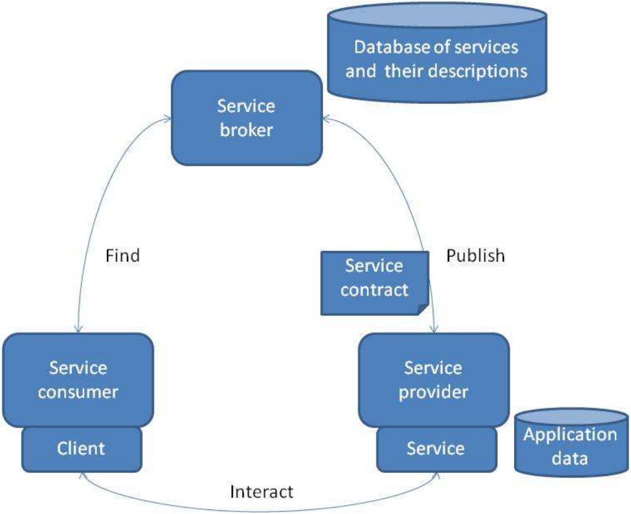
Approach

- From a Web for humans to a Web for humans & machines
    - Provide support for distributed applications
- Technical choices
    - Using Remote Procedure Calls and Corba-style
    - Based on Web standard, notably XML
    - Simplicity (more limited than Corba)
- Killer applications
    - Electronic commerce
    - Distributed data integration in Web portals and mashups

Running Web services



Searching for Web services



3 Standards: SOAP, WSDL, UDDI

- To allow services to be defined, deployed, found and used in an automated manner
- The client finds an appropriate service via a service broker (a discovery agency, a service repository)
  This is using UDDI
- The client gets the interface of the service from the service provider
  The interface is described in WSDL
- The client interacts with the service provider
  The protocol is SOAP

SOAP: Simple Object Access Protocol

- Main idea: interoperability between distributed applications
- Stateless communication protocol
- Based on XML for arguments of calls and results
- Independent on communication protocol
  - Can use HTTP (synchronous) or SMTP (asynchronous)
- Simple and extensible

SOAP Message Model

- Transport binding
  - How to get the message to its destination
  - Isolates message from transport, for portability across different transports
- Message envelope
  - What features and services are represented in the message
  - Who should deal with it
- SOAP header
  - Metadata: for the recipients
  - Information to indicate who should process the message
- SOAP body: The actual message call arguments/result

Example - call

```
POST /InStock HTTP/1.1
Host: www.stock.org
Content-Type: application/soap; charset=utf-8

<?xml version="1.0">
<soap:Envelope smlns:soap=... soap:encodingStyle=...>
   <soap:Header> ... </soap:Header>
   <soap:Body xmlns:m=http://www.stock.org/stock>
      <m:GetStockPrice>
         <m:StockName>IBM</m:StockName>
      </m:GetStockPrice>
   </soap:Body>
</soap:Envelope>
```

Example - response

```
HTTP/1.1 200 OK
Connection: close
Content-Type: application/soap; charset=utf-8
Date: Mon, 28 Sep 2002 10:05:04 GMT

<?xml version="1.0">
<soap:Envelope smlns:soap=... soap:encodingStyle=...>
   <soap:Header> ... </soap:Header>
   <soap:Body xmlns:m="http://www.stock.org/stock">
      <m:GetStockPricesResponse>
         <m:Price>34.5</m:price>
      </m:GetStockPricesResponse>
   </soap:Body>
</soap:Envelope>
```

Status

- Many implementations of SOAP
- Mostly RPC over HTTP
- Some UDDI reportories emerging, rather limited
- W3C (World Wide Web Consortium) SOAP 1.2 Recommandation
- Big players
  - Apache Axis: open-source Web server
  - J2EE (Sun, IBM, etc)
  - Microsoft and .NET
  - Sun and Sun ONE
  - HP and e-speak
  - IBM, Oracle and many others

Using Web services is easy

- Develop some application in Java
- Deploy an Axis server if one is not already available
- Expose a Java class automatically as Web service
  - A few lines of code
  - Each method becomes a Web service
  - Automatic serialization/deserialization of Java current types
  - Exception handling
  - Generation of stubs: local object that can call Web services

Orthogonal issues

- Error management
- Negociation
- Security (e.g., TLS, SSL)
- Quality of service
- Performance
- Reliability and availability

Web Service Description Language

- An XML dialect for describing Web service interfaces
  - "Black box" interface
- What are the available operations
- How are they activated: address, protocol
- Message format for the call and the response
- Nothing on the semantics

Service description in WSDL, in short

- Operation: exchange of messages
    - Request/response pair no state (not yet in WSDL)
    - input or output only, input/output
    - Data in messages are typed using XML schema
- Port type: collection of operations
- Port: an implementation of a port type associated to an address
- Service is a collection of ports

Universal Description, Discovery and Integration (of services)

- Where can I find the service I need
- Define types of services
- Publish some service of certain type
    - a unique identifier is assigned to it
- Search/query the repository to find some service

UDDI content in short

- White pages: the companies
  Address, tel number, web site, kind of activity
- Yellow pages: the services
  - Textual description
  - Classification in categories
- Green pages: technical info in WSDL

Security

- Functionalities: access control, confidentiality, authentication, message integrity and non-repudiation
- Infrastucture: public key crypto system such as RSA
- SSL: secure socket layer; a protocol to transmit encrypted data
- HTTPS = HTTP over SSL; very used
- XML digital signature with non-repudiation
- XML encryption; allows selective encryption of parts of a document

Outline

1 Distributed data management

2 The basis: distributed computing

3 Web services

4 Composition of services

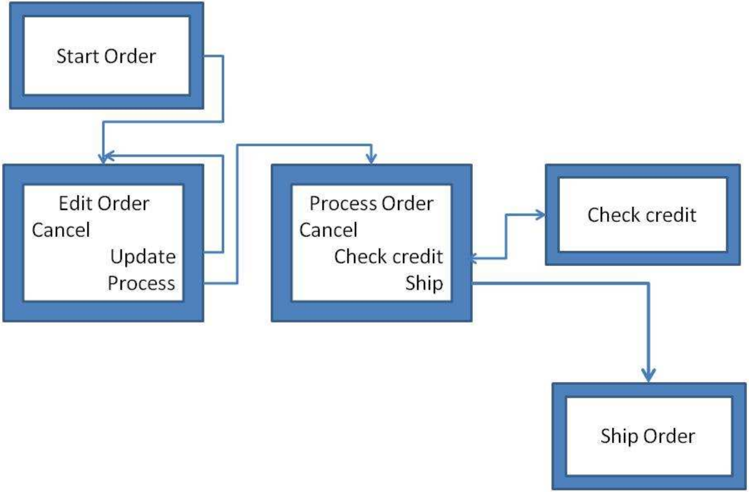
5 Web 2.0

6 Active XML

Composing Web services

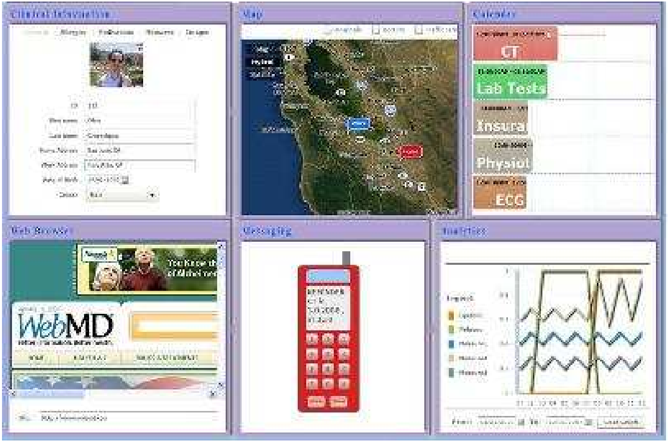
- Define interaction between several services, e.g., sequencing of two services
- Allows creating new services by combining several existing ones
  - Workflow of services
- BPEL: Business Process Execution Language, OASIS Standard

Example of workflow



Mashups

- Data integration
- Imports and use external Web services



Outline

1 Distributed data management

2 The basis: distributed computing

3 Web services

4 Composition of services

5 Web 2.0

6 Active XML

Web 2.0

- New trends on the Web: users create content, share information, interact, collaborate
- Buzz: communities, social networks, wikies, mashups, blogs, folksonomies (aka collaborative tagging)
- From a technical viewpoint: update (and not only queries), data integration, monitoring

Outline

1. Distributed data management
2. The basis: distributed computing
3. Web services
4. Composition of services
5. Web 2.0
6. Active XML

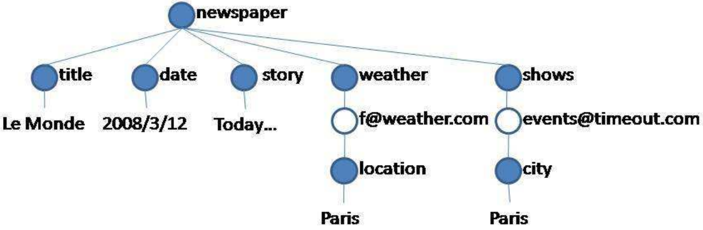
To illustrate Web services//Active documents

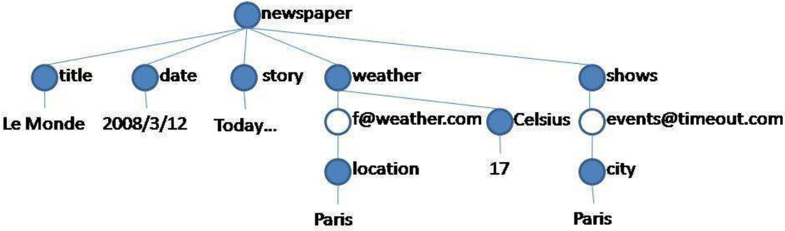
- Active XML (AXML, for short) documents are XML documents with embedded calls to Web services
- Combine “extensional” XML data with data defined “intensionally”
- AXML documents evolve in time when calls to their embedded services are activated.
- Old idea: embedding calls in data; stored procedures in relational systems

An AXML document in serialized form

```
<?xml version="1.0" encoding="UTF-8" ?>
<newspaper xmlns="http://lemonde.fr" xmlns:...>
  <title>Le Monde</title>
  <date>2008/3/12</date>
  <story>Today...</story>
  <weather>
    <axml:call service="f@weather.com" >
      <location>Paris</location>
    </axml:call>
  </weather>
  <shows>
    <axml:call service="events@timeout.com">
      <city>Paris</city>
    </axml:call>
  </shows>
</newspaper>
```

In a graphical form





Some aspects

- If a client asks for an AXML document, the server has the choice between materializing part of the data before sending it or not
- A query over an AXML document can be used to specify some complex data integration/publication task
    - An issue becomes its efficient evaluation

Merci