

Grid Computing

Web Services

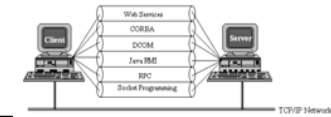
Paul A. Farrell
Fall 2006

The Grid: Core Technologies
Maozhen Li, Mark Baker
John Wiley & Sons; 2005, ISBN 0-470-09417-6

Paul A. Farrell 2006 KENT STATE Grid Computing 1

Web Services

- Based on Service Oriented Architecture (SOA)
 - Clients : service requestors
 - Servers : services providers
- Focus on simple open standards e.g. XML, HTTP
- Definition:
 - *Essentially, a Web service is a loosely coupled, encapsulated, platform and programming language neutral, composable server side component that can be described, published, discovered and invoked over an internal network or on the Internet.*



Paul A. Farrell 2006 KENT STATE Grid Computing 2

Explanation

- *Loosely coupled*: implementation is free to change as long as the service interface remains the same.
- *Encapsulated*: implementation completely invisible to the client
- *Platform and programming language neutral*: can be implemented in any language and deployed on any platform.
- *Composable*: can be composed from a number of deployed services.
- *Server Side Component*: Service can vary from a complete application to a subroutine
- *Described*: XML based interface describes functionality and capabilities

Paul A. Farrell 2006 KENT STATE Grid Computing 3

Explanation (2)

- *Published*: service can be registered with a service registry.
- *Discovered*: client can discover a service by searching a service registry
- *Invoked*: service can be bound to via standard transport protocols such as HTTP or FTP.
- *Internal network or the Internet*. service can be made available strictly within an organization or it can be offered across the firewall, available to any consumer connected to the Internet.

Paul A. Farrell 2006 KENT STATE Grid Computing 4

Core Standards

- Service Oriented Architecture (SOA)
- Web Services Description Language (WSDL)
- Universal Description, Discovery, and Integration (UDDI)

- Web Services Inspection (WS-Inspection)
 - defines WS-Inspection Language (WSIL) for service description and discovery

Paul A. Farrell 2006 KENT STATE Grid Computing 5

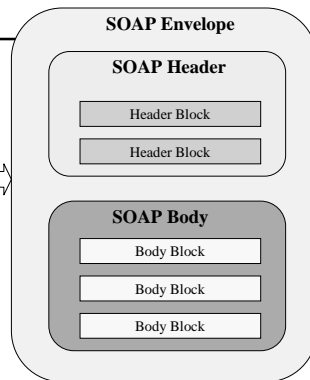
Service Oriented Architecture (SOA)

- Simple and lightweight communication protocol
- To enable exchange of messages in XML format normally over HTTP

- SOAP envelope
 - Namespace specification
xmlns:SOAP-ENV (SOAP Envelope), xmlns:xsi (XML Schema for Instance) or xmlns:xsd (XML Schema Definition)
 - Encoding rules for application defined data types
 - Optional header for authentication, transactions, payment
 - Body

Paul A. Farrell 2006 KENT STATE Grid Computing 6

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV
="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle
="http://schemas.xmlsoap.org/soap/encoding/"
<SOAP-ENV:Header>
...
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<!-- RPC Method Call here -->
<SOAP-ENV:Fault> ... </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

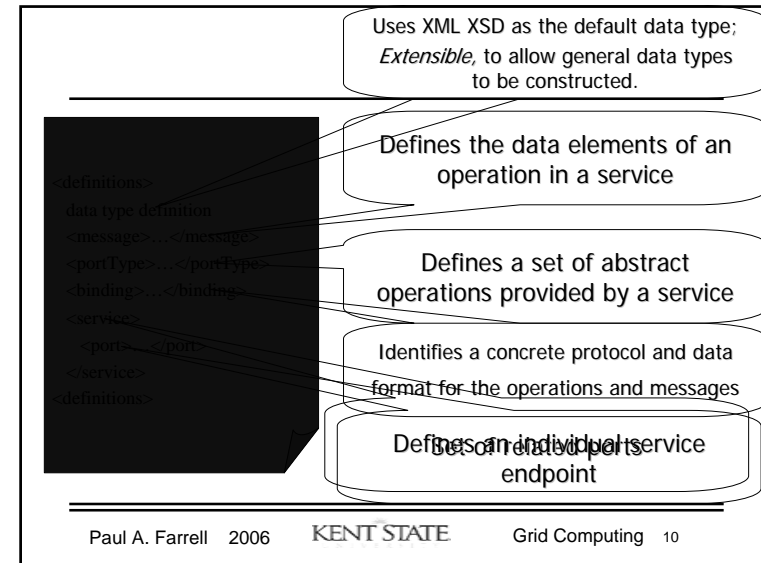
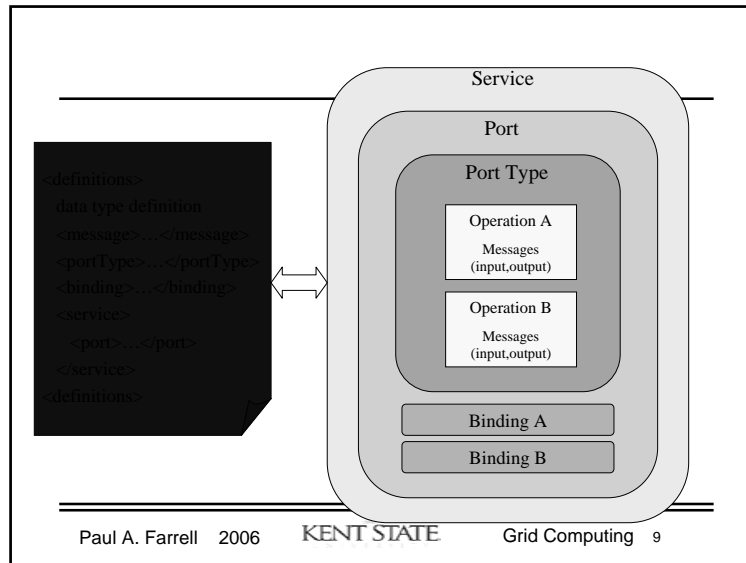


Paul A. Farrell 2006 KENT STATE Grid Computing 7

Web Services Description Language (WSDL)

- XML based specification to describe a Web service
 - Functionality, residency, invocation
- Defines services as a set of network endpoints or *ports* either
 - using a synchronous RPC-based mechanism
 - SOAP messages contain parameters and return values
 - using a synchronous or asynchronous document-oriented message-exchange
 - SOAP messages contain XML documents

Paul A. Farrell 2006 KENT STATE Grid Computing 8



Services and Ports

- There may be arbitrary number of bindings for a given portType, i.e., a binding can be document-oriented or use RPC.
- Ports within a service have the following relationship:
 - None of the ports communicate with each other.
 - If a service has several ports, that share a `<portType>`, but employ different bindings or addresses, the ports provide semantically equivalent behaviour.

Universal Description, Discovery, and Integration (UDDI)

- Industry standard for service registration (publication) and discovery
- A UDDI registry is similar to a CORBA trader service
- Data in UDDI can be organised in the following ways:
 - White Pages: includes general information about a provider (name, contact information, etc)
 - Yellow Pages: allows clients to discover a Web service based on its categorisation
 - Green Pages: technical information about a Web service, usually with a reference to an external WSDL document describing how to interact with the service
- UDDI is layered over SOAP
- UDDI registry exposes a set of APIs in the form of SOAP-based Web services

WS-Inspection

- Complementary to UDDI
- Allows service description information to be distributed to any location using a simple extensible XML document format
- Mainly provides
 - an XML format for listing references to existing service descriptions.
 - a set of conventions so that it is easy to locate WS-Inspection documents.
- Provides a means for aggregating references to pre-existing service description documents in different formats

Paul A. Farrell 2006 KENT STATE Grid Computing 13

WS-Inspection

- In a WS-Inspection document, a single service can have more than one reference to a service description
 - service description is usually a URL that points to a WSDL document
 - can be a reference to an entry within a UDDI registry
- Service requestors use standard Web-based access mechanisms (e.g., HTTP GET) to retrieve a WS-Inspection document

Paul A. Farrell 2006 KENT STATE Grid Computing 14

```
<?xml version="1.0"?>
<inspection xmlns="http://schemas.xmlsoap.org/ws/2001/10/inspection/">
  <service>
    <description
      referencedNamespace="http://schemas.xmlsoap.org/wsdl/"
      location="http://example.com/example.service.wsdl" />
  </service>

  <service>
    <description
      referencedNamespace="urn:uddi-org:api">
      <wsiluddi:serviceDescr
        "http:
        <wsiluddi:serv
          2946BE
        </wsiluddi:serv
      </wsiluddi:serviceDescr
    </description>
  </service>

  <link referencedNamespace=
    "http://schemas.xmlsoap.org/ws/2001/10/inspection/"
    location="http://example.com/tools/toolservices.wsil"/>
</inspection>
```

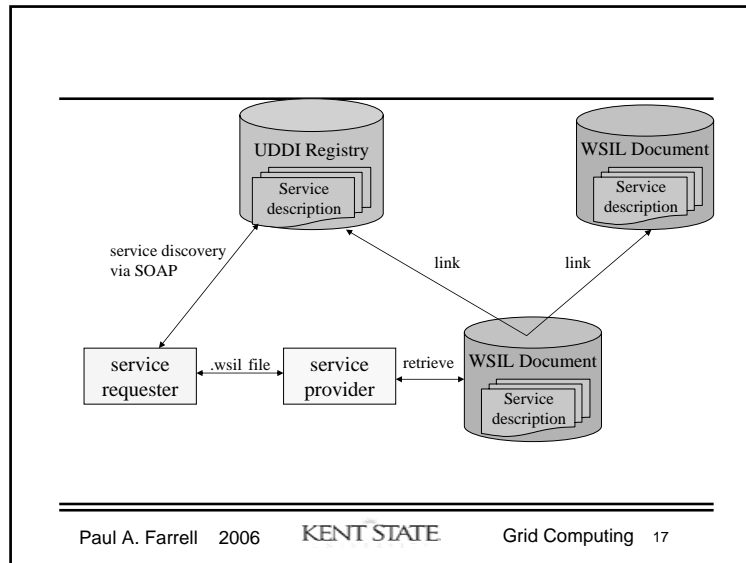
<link> element used to reference a collection of service descriptions – here a WS-Inspection document

Paul A. Farrell 2006 KENT STATE Grid Computing 15

WS-Inspection and UDDI

- Complementary
 - UDDI: high degree of functionality, increased complexity
 - WS-Inspection: less functionality, low overhead
- Can be used together
 - UDDI registry populated by robot crawling WS-Inspection documents on the Web
 - UDDI registry may be discovered when a requestor retrieves a WS-Inspection document

Paul A. Farrell 2006 KENT STATE Grid Computing 16



Web Services Implementations

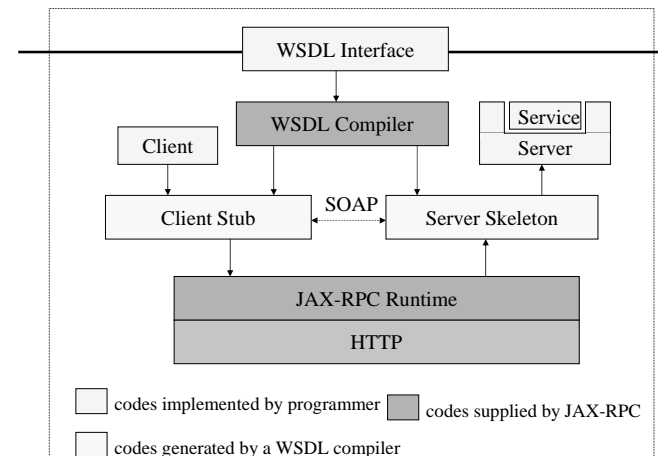
- Three aspects
- 1. Programming model**
 - specifies how to write client codes to access Web services, how to write service implementations, how to handle other parts of the SOAP specification, such as headers, and attachments
 - 2. Deployment model**
 - the framework used to deploy a service and provide a Web service deployment descriptor (a wsdd file) to map the implementation of the service to SOAP messages
 - 3. SOAP Engine**
 - receives SOAP messages and invokes Web service implementations

Paul A. Farrell 2006 KENT STATE Grid Computing 18

J2EE

- Standard for developing, building and deploying Java-based applications
 - Web sites, software components, or packaged applications
 - Recent extension supports building XML-based Web services
- J2EE provides following APIs
 - Java API for XML Processing (JAXP)
 - Java Architecture for XML Binding (JAXB)
 - processes XML documents using schema-derived JavaBeans component classes
 - Java API for XML-based RPC (JAX-RPC)
 - Java API for XML Messaging (JAXM) and SOAP with Attachments API for Java (SAAJ)
 - send SOAP messages over the Web in a standard way
 - Java API for XML Registries (JAXR)
 - provides a standard way to interact with business UDDI registries

Paul A. Farrell 2006 KENT STATE Grid Computing 19



Paul A. Farrell 2006 KENT STATE Grid Computing 20

Apache Axis

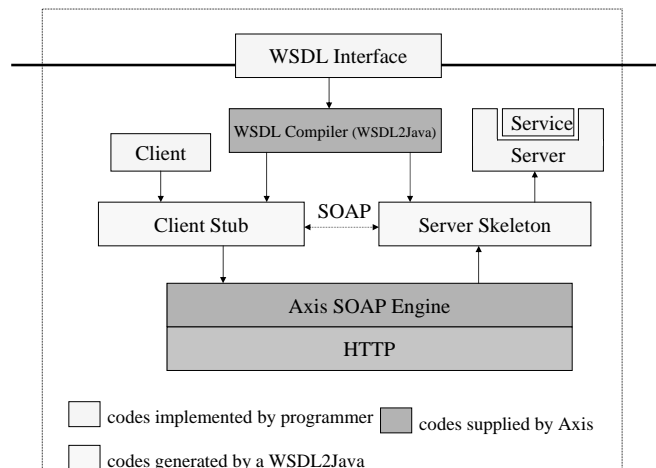
- SOAP engine that can be used to exchange SOAP messages between clients and services
- Provides support for WSDL operations
 - Java2WSDL used to generate a WSDL document from a Java interface,
 - WSDL2java used to generate a client-side stub and a server-side skeleton
- Does not provide support for service discovery and publication
 - UDDI4Java from IBM can be used together with Axis for this

Paul A. Farrell 2006 KENT STATE Grid Computing 21

Web service creation using Axis

- Write a Java interface.
- Use Java2WSDL to generate a WSDL interface.
- Use WSDL2Java on the WSDL interface to generate a client side stub and a server side skeleton.
- Write a service to implement the WSDL interface.
- Write a client.
- Compile all the codes with *javac* compiler.
- Write a Web service deployment descriptor (a wsdd file) to deploy the service in Jakarta Tomcat Web server
- Start Tomcat.
- Start the client to invoke the service.

Paul A. Farrell 2006 KENT STATE Grid Computing 22



Paul A. Farrell 2006 KENT STATE Grid Computing 23

Microsoft .Net

- Microsoft platform for building Web services
- A .NET Web service supports the WSDL 1.1 specification, uses a WSDL document to describe itself
 - But an XML namespace has to be used within a WSDL file to uniquely identify the Web service's endpoint
- Provides
 - a client-side component that lets a client invoke a Web service described by WSDL
 - a server-side component that maps Web service operations to a COM-object method call as described by the WSDL interface and a Web Services Meta Language (WSML) file
 - needed for Microsoft's implementation of SOAP
- Web services can be published using DISCO files or via a UDDI registry

Paul A. Farrell 2006 KENT STATE Grid Computing 24

Options for Invocation of Web services using .Net

1. Use the built-in .Net SOAP message classes
2. Construct a Web service listener manually, using for example
 - Microsoft XML Parser (MSXML),
 - Active Server Pages (ASP), or
 - Internet Server Application Programming Interface (ISAPI)
3. Use Microsoft SOAP Toolkit 2.0 to build listener that communicates with a service implemented with COM

Web Services and the Grid

- Web services are
 - independent of platforms, programming languages and locations
 - can be described, published and dynamically discovered and bound to with WSDL
- Benefits for Grid
- Grid requires support for the dynamic discovery and composition services in heterogeneous environments
 - necessitates mechanisms for registering and discovering interface definitions and endpoint implementation descriptions;
 - for dynamically generating proxies based on (potentially multiple) bindings of specific interfaces
 - WDSL supports this
- Web services technologies are based on internationally recognised standards and widely adopted
 - Grid framework based on Web services will be able exploit numerous tools and extended services

