expand internal operations with new functionalities and to easily access organizations outside the enterprise

7/19 Key principles of SOA:
i) Loose coupling
ii) Service Contract (WSDL doc containing the methods for impl)
   - Code first approach
   - Contract first approach
iii) Autonomy
iv) Abstraction. (Service hide logic from outside world)
v) Reusability
vi) Composability
vii) Statelessness
viii) Discoverability

Creating web service
Project → Web App → Web Service
Design view → Add operation (Specify parameters)

19 Components of WSDL
- Web Service Description Language
- Written in XML
- Describes a web service
- Specifies location & methods the service exposes

⇒ SOAP is the envelope containing the actual message
⇒ WSDL gives the location & list of methods the service provides.

Structure

Types of section: Abstract & concrete

Components of WSDL:

- Types
- Messages
- PortTypes
- Binding
- Service

} Abstract section

} concrete section

Messages uses the datatypes (defined by user) enumerated in Types component

PortTypes refers to mess. definitions in the Messages component (Has various operations)

Binding defines which protocol is to be used to transmit the data/messages on wire.

Types is omitted if only simpleType datatypes are used.

Service gives the location of the service.

Types of operation

Synchronous

- Request-response
- Solicit-response (service sends a mess & receives a response)

Asynchronous

- One way (Service receives a msg)
- Notification (Service sends a msg)

Binding element
3 mechanisms:
- SOAP
- HTTP
- MIME

Single port can have multiple bindings.

Parts:
- Binding type
- SOAP operation
  - func. name to be called
  - input para
  - output para

18/19 Contract first approach for creating SOAP web service.

Code first
Writing source code for web service & then WSDL file.

Contract first
Creating web-service based on given WSDL file.

Steps: (Netbeans 6.0)
i) Create web app.
ii) Give Project Name
iii) Right click on project → New → WSDL Doc.
iv) Give File & Folder name
v) Dialog Box appears asking for "abstract configuration"
vi) Enter binding name & binding type: (SOAP)
service
~~port~~ name & port type
(Generate configuration)

vii) On WSDL view, right click Port Types → (MyWSDLPortType) → Add → Operation
// To add multiple operations in single port type.

viii) Right click (MyWSDLBinding) → Add → Binding Operation
(explicitly adding concrete configurations)

ix) On the operation you created, right click (under bindings) → Add → soap bi

x) Similarly, add soap operation

xi) Copy the contents of soap body (use a namespace) to the recently created operation.

xii) Right click project name → New → Web Servi from WSDL.

xiii) Enter name, package,
Select local WSDL Doc.

xiv) Web service folder created
Open the java file

xv) Add logic for the operations in the web service

(Steps upto creating WSDL doc is done by client. Adding the WSDL to web service & coding the logic is done by

developers).

BPEL (Business Process Execution Language) used for integration of web services.

Business Process (BP) - consists of both internal computations and invocations of operations exposed by web service partners.
(web service providers)
ex- hotel & travel reservations in a 'picnic' service.

- The operations it exposes constitute its interface to its partners.
- Seq. of invocations it executes is referred to as a protocol and
  - is data dependent (O/P of hotel is I/P for travel ← Hotel res. needs to be invoked before travel)
  - responds to exceptional conditions (travel is not available for selected dates).

⇒ Businesses today requires to quickly adapt to customer needs & market conditions.
(EAI [Enterprise Application Integration] & B2B interactions)

⇒ Needs to be flexible internally & externally.

⇒ Without a common set of standard, each organization is left to develop its own proprietary web services.

Web Service composition
Provides an open, standards-based approach for connecting web services together to create higher level of web services.

Issues of integration
- Must be able to comm. with other ws.
- Must be able to access & modify data received in messages
- Must have control constructs (switch, if, else etc.)
- Must be able to handle faults.

Basic requirements
- Ability to invoke services in an asynchronous manner
- Manage exception & transactional integrity
- Provide dynamic, flexible & adaptable framework.

Standards
- BPEL4WS
- WSCI (Web services choreography Interface)
- BPML (Business Process Management Language).

BPEL

WEEK 2

Orchestration

* A single Director in control

* A central process takes control over the involved web services & co-ordinate the execution
Ex - Picnic service.

* The involved WS do not know that they are involved into a composition

Choreography

* Defines interaction WS - Choreography describes publically visible mess. exchange

* Does not rely on a central co-ordinator Each web service knows exactly when to execute

BPEL - Process Models

Provides support for 2 business process models:

i) Executable
Models the behavior of participants in a specific business interaction, a private workflow executed by Orchestration engine

ii) Abstract
Specifies public messages exchange between parties only. They follow the choreography paradigm

Executable BP	Abstract BP
* complete description of BP, (all computations)	* contains only externally visible behavior of BP - Not executable - Internal decision making algo. & data manipulation not describe

⇒ Langs describing both BP's share a common core but, differ in how they manipulate data.

BPEL goals
- Platform independent - XML based lang.
- Service & messages are first order citizens.
- Asynchronous

⇒ Web services are stateless but, BPEL is responsible for maintaing the state

BPEL4WS
- Describes control logic for WS coordination
- Interpreted & executed by BPEL engine.

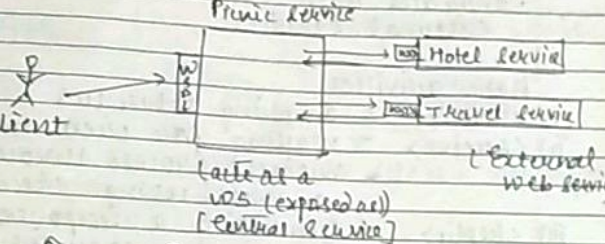
⇒ BPEL servers exist for J2EE, .NET & others

⇒ Use web services standard as a base
i) Every BPEL is exposed as a web service using WSDL which describes

the public entry & exit points of the process

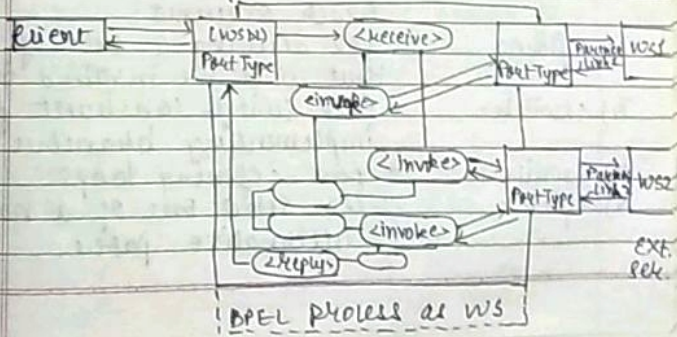
ii) Interacts through WSDL interfaces with external web services.

iii) WSDL data types are used to describe info. flow within the BPEL process



Thus, BPEL service plays two roles.

BPEL - Example Process



A BPEL process consists of steps. Each step is called activity.

Activity

Basic
- Interacts with external services

Structured

Basic activities
i) <invoke> - Invoking other WS
ii) <receive> - Waiting for client to invoke business process through sending a msg. using <receive>
iii) <reply> - generating a response for synchronous operations.
iv) <assign> - Manipulating data var.
v) <throw> - Indicating faults & exceptions
vi) <terminate> - Terminating entire process.

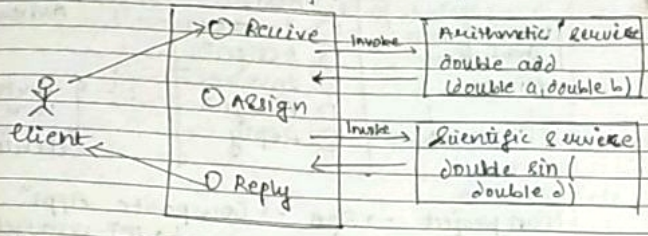
Structured activities
i) <sequence> - for defining a set of activities that will be invoked in an order sequence.
ii) <flow> - for defining a set of activities that will be invoked in parallel
iii) <switch> - case-switch construct for implementing branches
iv) <while> - for defining loops
v) <pick> - to select one of a no. of alternative paths.

Page No.
Date

<partners>

Define partner links by using <Partnerlink> and variables using <variables>

14/8/19



Create web service (2 separate web projects):
i) Arithmetic Service
ii) Scientific Service

New project → SOA → BPEL module → Give name

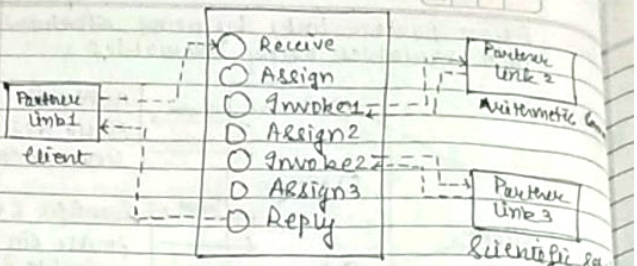
Right click on Process Files → New → WSDL doc. (ArithmeticScientificBPEL)

Enter total inputs needed from client (2 here)
Enter total output para. needed from (1 here)

Right click on Process Files → New → BPEL Process.

Drag and drop WSDL for BPEL
Drag and drop the two ext. services.

Drag & drop receive activity.
(Select appropriate partner links)



New project → SOA → Composite Appl^n
↳ JBI modules
(BPEL engine available in Netbeans)

Add JBI module → <Add your BPEL module>

Major steps:
1) Create ArithmeticService (Web Appl^n)
2) Create ScientificService (Web Appl^n)
3) Create BPEL Module (SOA)
   a) Prepare WSDL for client
   b) Configure & invoke web service
4) Create Composite Appl^n (SOA)
   a) Add JBI module of BPEL process
   b) Prepare test cases based on WSDL file.

21/8/19 Understanding Links to Partners.

⇒ Declares how two parties interact & what each party offers

⇒ Links to all parties BPEL interacts with are called partner links

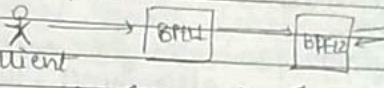
BPEL process interact with external WS in two ways:
i) BPEL process invoke operations on other WS
ii) BPEL process receive invocations from clients

→ INVOKED PARTNER LINKS
Links to WS that are invoked by BPEL process

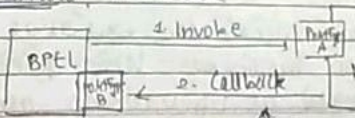
CLIENT PARTNER LINKS
Links to WS that invoke the BPEL process

→ BPEL uses client as partner links for two reasons:
i) For asynchronous comm.
ii) Treat as service

→ Partners may be:
i) Service invoked by process
ii) Service that invoke the process
iii) Service that have both roles: they are invoked by process & they invoke the process



Asynchronous Callback



[portType ≈ class] analogous to

If synchronous, callback not req. as client is suspended till invoke not served

typc

⇒ partner link must have atleast one and atmost two roles

⇒ For each role, we must specify a portType for interactions

→ Not part of BPEL process specification documentation

Each partner link specify:

- Name
- myRole
- partnerLinkType
- partnerRole

Name serves as a reference for interactions via a particular partner link.

myRole & partnerRole, both need to be specified only when partnerLinkType specifies two roles. (allync comm)

BPEL process tag

Specifies:

- Name - name of BPEL business process
- targetNameSpace
- Xmlns
- queryLanguage - used for node selection in assignments, properties & other uses. Default - XPath 1.0
- abstractProcess.

Variables

⇒ BPEL BP model the exchange of messages between involved WS. Messages are exchanged as operations are invoked.

→ BPEL is stateful & needs to preserve the state of variables which are stored in variables
When we declare a variable, we must specify the variable name & type

→ Create Instance specifies whether an instance of the BPEL process should be created or not when <receive> is encountered

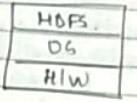
→
```
<assign name="Assign1">
  <copy>
    <from> — </from>
    <to> — </to>
  </copy>
  <copy>
    <from> — </from>
    <to> — </to>
  </copy>
</assign>
```
When multiple variables are passed

27/8/19 Hadoop
- HDFS (Hadoop Distributed File System)
- Map-Reduce framework

Characteristics: (of BigData)
i) Volume (in terms of Petabytes)
ii) Velocity - Rate at which data gets generated.
iii) Variety - The types of data (text, pictures, video etc.)

HDFS is useful for storage of the & Map-Reduce framework assists preprocessing of the data.

Install HDFS:



File
Data (stored in blocks)
Metadata (i-node in case of Linux) (stored in blocks)

Default block size (in Windows): 4 KB

If it is 512 MB then, even to store a text file containing 'Hello World' will take up 512 MB of block (too much wastage of block size)

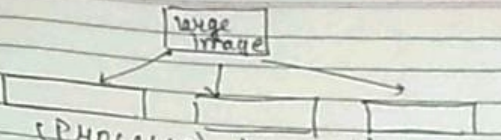
On the other hand, meta data info is less when blocks are of 512MB

Thus, to store petabytes of info. it is better to have larger block size. For this reason, HDFS lies on top of normal OS.

SIMD (Single ins. multiple Data).

Single ins. is used on multiple data.

HDFS works in a similar way.



(Processed by multiple replicas on different machines)