

UNIT 1

1. Define architecture along with types.
 - a. IT departments started to recognize the need for a standardized definition of a baseline application that could act as a template for all others.
 - b. This definition was abstract in nature, but specifically explained the technology, boundaries, rules, limitations, and design characteristics that apply to all solutions based on this template.
 - c. This was the birth of the application architecture.

2. List various Characteristics of SOA.
 - Services are discoverable and dynamically bound.
 - Services are self-contained and modular.
 - Services stress interoperability.
 - Services are loosely coupled.
 - Services have a network-addressable interface.
 - Services have coarse-grained interfaces.
 - Services are location-transparent
 - Services are composable.
 - Service-oriented architecture supports self-healing.

3. Differentiate SOA vs Traditional Distributed Internet Architecture.
 - Multiple client-server architectures have appeared
 - Client-server DB connections have been replaced with Remote Procedure Call connections (RPC) using CORBA or DCOM
 - Middleware application servers and transaction monitors require significant attention
 - Multi-tiered client-server environments began incorporating internet technology in 90s.
 - The browser shifted 100% of application logic to the server
 - Distributed Internet architecture introduced the Web server as a new physical tier
 - HTTP replaced RPC protocols

4. Compare SOA with client-server and distributed internet architectures.
 - The technology set for client-server applications included 4GLs like VB and PowerBuilder, RDBMSs
 - The SOA technology set has expanded to include Web technologies (HTML, CSS, HTTP, etc)

- SOA requires the use of XML data representation architecture along with a SOAP messaging framework

5. What are the Logic components of the Web services framework?

- Fundamental parts of the framework
 - SOAP messages
 - Web service operations
 - Web services
 - Activities
- Messages = units of communication
- Operations = units of work
- Services = units of processing logic
- Processes = units of automation logic

6. What are messages, operations, services and processes?

- Message
 - A message represents the data required to complete some or all parts of a unit of work.
 - Operation
 - An operation represents the logic required to process messages in order to complete a unit of work.
- Service
 - A service represents a logically grouped set of operations capable of performing related units of work.

7. List the various components of SOA.

- Messages = units of communication
- Operations = units of work
- Services = units of processing logic
- Processes = units of automation logic

8. List the common principles of service-orientation.

- Services are reusable
- Services share a formal contract
- Services are loosely coupled
- Services abstract underlying logic
- Services are composable
- Services are autonomous
- Services are stateless
- Services are discoverable

9. Are services reusable? Justify.

- Regardless of whether immediate reuse opportunities exist, services are designed to support potential reuse.
- Service-oriented encourages reuse in all services.
- By applying design standards that require reuse accommodate future requirements with less development effort.

10. Are services loosely coupled or tightly coupled? Justify.

- Services must be designed to interact without the need for tight, cross-service dependencies.

UNIT II

1. What is a web service?

- Services in real world automation solutions requires the use of a technology capable of preserving fundamental service-orientation, while implementing real world business functionality.
 - Web services framework is flexible and adaptable.
 - Web services can be designed to duplicate the behavior and functionality found in proprietary distributed systems
 - This flexibility has allowed Web services to become part of many existing application environments and has been one of the reasons behind their popularity

2. What are the characteristics of Web service framework?

- an abstract (vendor-neutral) existence defined by standards organizations and implemented by (proprietary) technology platforms
- core building blocks that include Web services, service descriptions, and messages
- a communications agreement centered around service descriptions based on WSDL
- a messaging framework comprised of SOAP technology and concepts
- a service description registration and discovery architecture sometimes realized through UDDI
- a well-defined architecture that supports messaging patterns and compositions a second generation of Web services extensions

3. What are the web service roles?

- Service provider
- Service requester

- Intermediary services
- Service compositions

4. What is a service model?

The roles we've explored so far are agnostic to the nature of the functionality being provided by the Web service. They are generic states that a service can enter within a generic context. The manner in which services are being utilized in the real world, though, has led to a classification based on the nature of the application logic they provide, as well as their business-related roles within the overall solution. These classifications are known as service models.

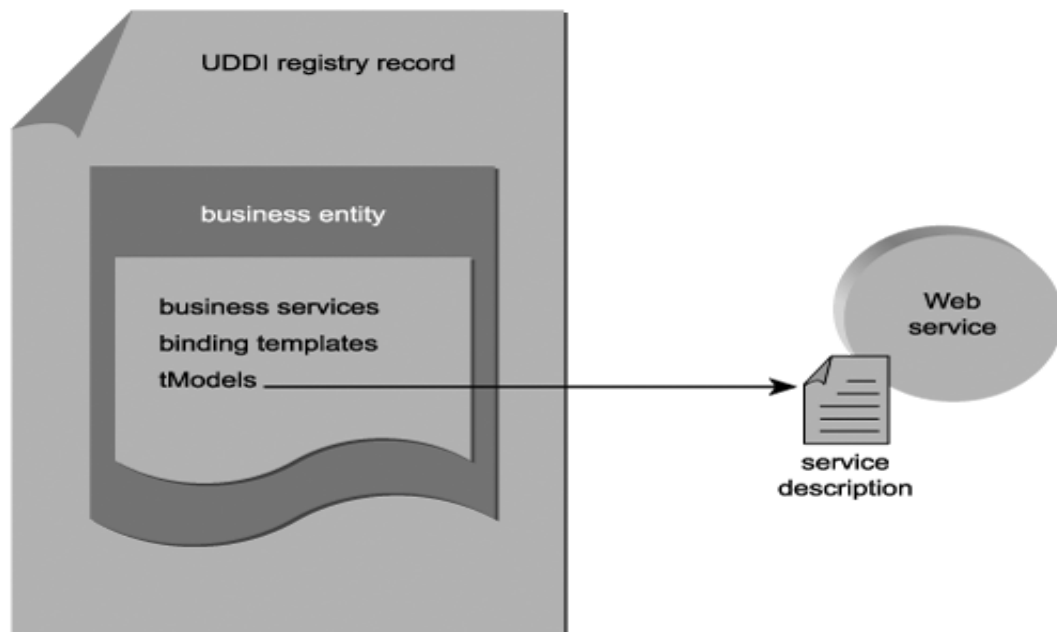
5. What are two types of WSDL elements?

Let's dig a bit deeper into how the service description document itself is organized. A WSDL service description (also known as WSDL service definition or just WSDL definition) can be separated into two categories:

- abstract description
- concrete description

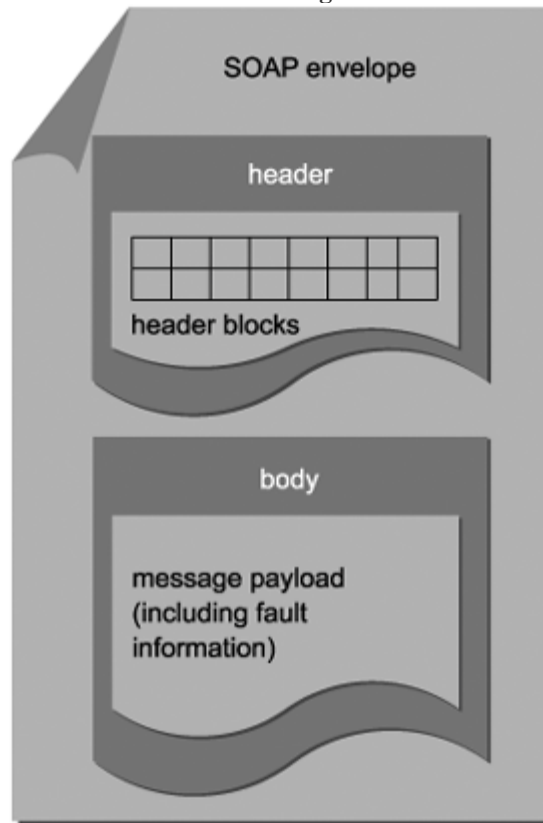
6. Draw basic structure of UDDI with its elements.

Figure 5.19. The basic structure of a UDDI business entity record.



7. Brief about SOAP elements along with its structure.

- Messages - SOAP specification's main purpose is to define a standard message format
 - **Envelope, header, and body,**
- Figure 5.21. The basic structure of a SOAP message.

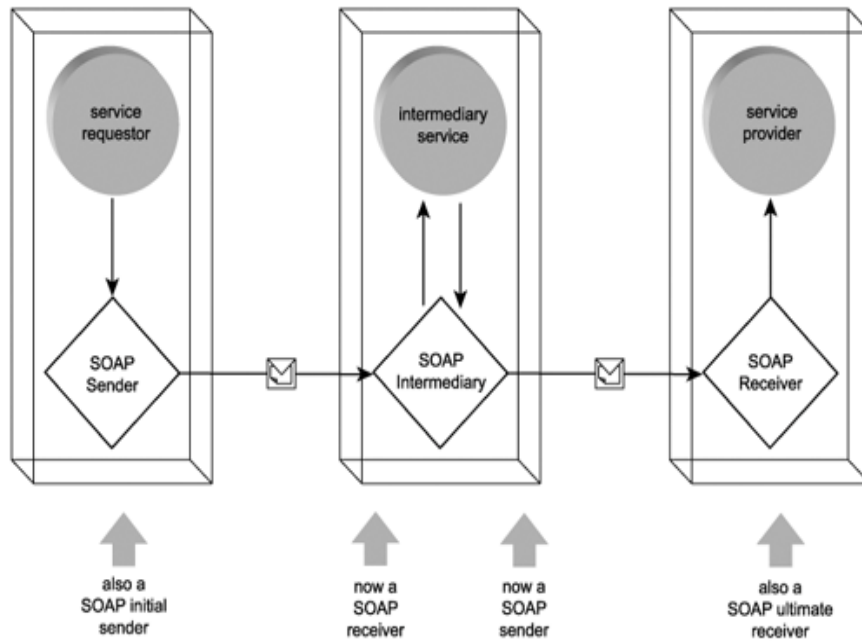


8. Write about Soap Fault element and Must Understand attribute.

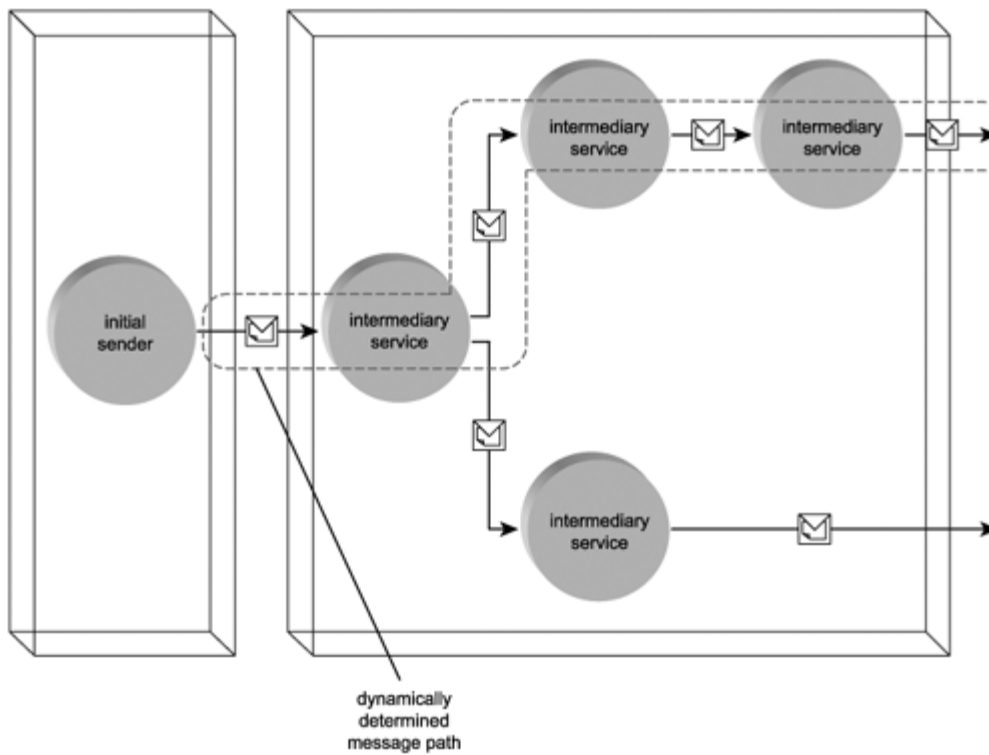
- **Faults**
 - SOAP messages offer the ability to add exception handling logic by providing an optional fault section that can reside within the body area. The typical use for this section is to store a simple message used to deliver error condition information when an exception occurs.

9. List different types of SOAP nodes with neat sketch.

Figure 5.25. Different types of SOAP nodes involved with processing a message.



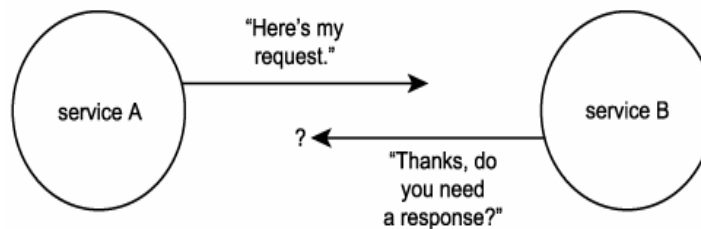
10. With a neat diagram depict how a message path is resolved dynamically.



11. What do you mean by Message Exchange Patterns?

Message exchange patterns (MEPs) represent a set of templates that provide a group of already mapped out sequences for the exchange of messages. The most common example is a request and response pattern. Here the MEP states that upon successful delivery of a message from one service to another, the receiving service responds with a message back to the initial requestor.

Figure 6.2. Not all message exchanges require both requests and responses.



12. What is coordination and composition?

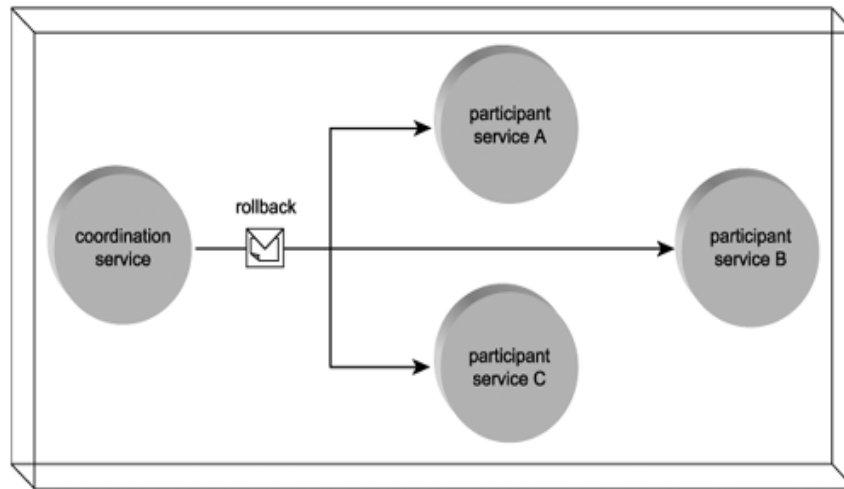
The coordinator composition consists of the following services:

- Activation service Responsible for the creation of a new context and for associating this context to a particular activity.
- Registration service Allows participating services to use context information received from the activation service to register for a supported context protocol.
- Protocol-specific services These services represent the protocols supported by the coordinator's coordination type. (This is further explained in the next sections.)
- Coordinator The controller service of this composition, also known as the coordination service.

13. Write about atomic transaction process with neat diagram.

As previously mentioned, the atomic transaction coordinator is tasked with the responsibility of deciding the outcome of a transaction. It bases this decision on feedback it receives from all of the transaction participants.

Figure 6.24. The coordinator aborting the transaction and notifying participants to rollback all changes.



14. Define Business activity states.

During the lifecycle of a business activity, the business activity coordinator and the activity participants transition through a series of states. The actual point of transition occurs when special notification messages are passed between these services.

For example, a participant can indicate that it has completed the processing it was required to perform as part of the activity by issuing a completed notification. This moves the participant from an active state to a completed state. The coordinator may respond with a close message to let the participant know that the business activity is being successfully completed.

15. Define Business activity protocols.

As with WS-AtomicTransaction, WS-BusinessActivity is a coordination type designed to leverage the WS-Coordination context management framework. It provides two very similar protocols, each of which dictates how a participant may behave within the overall business activity.

- The BusinessAgreementWithParticipantCompletion protocol, which allows a participant to determine when it has completed its part in the business activity.
- The BusinessAgreementWithCoordinatorCompletion protocol, which requires that a participant rely on the business activity coordinator to notify it that it has no further processing responsibilities.

16. What is Orchestration?

Organizations that already have employed enterprise application integration (EAI) middleware products to automate business processes or to integrate various legacy environments will likely already be familiar with the concept of orchestration.

17. What is Choreography?

A choreography is essentially a collaboration process designed to allow organizations to interact in an environment that is not owned by any one partner.

18. Write about Business Service Layer.

- Business services, on the other hand, are always an implementation of the business service model. The sole purpose of business services intended for a separate business service layer is to represent business logic in the purest form possible. This does not, however, prevent them from implementing other service models.

19. Write about Orchestration Service Layer.

- Therefore, all process services are also controller services by their very nature, as they are required to compose other services to execute business process logic. Process services also have the potential of becoming utility services to an extent, if a process, in its entirety, should be considered reusable.
- In this case, a process service that enables orchestration can itself be orchestrated

20. Write about Application Service Layer.

The application service layer establishes the ground level foundation that exists to express technology-specific functionality. Services that reside within this layer can be referred to simply as application services. Their purpose is to provide reusable functions related to processing data within new or legacy application environments.

UNIT III

2 marks

1. What is Service Oriented Analysis?

- The process of determining how business automation requirements can be represented through service-orientation is the domain of the service-oriented analysis.
- Objectives of service-oriented analysis
 - The primary questions addressed during this phase are:

- What services need to be built?
- What logic should be encapsulated by each service?

2. What is Business Centric SOA?

The majority of Web services currently being built are, more or less, a mixture of application and business services. These types of hybrid services are attractive because, with minimal effort, they fulfill immediate requirements with clear ROI benefits. The proliferation of **hybrid services** is the result of the bottom-up approach having become so commonplace. They provide an immediate means for all past forms of application architecture to take part in the open Web services communications framework

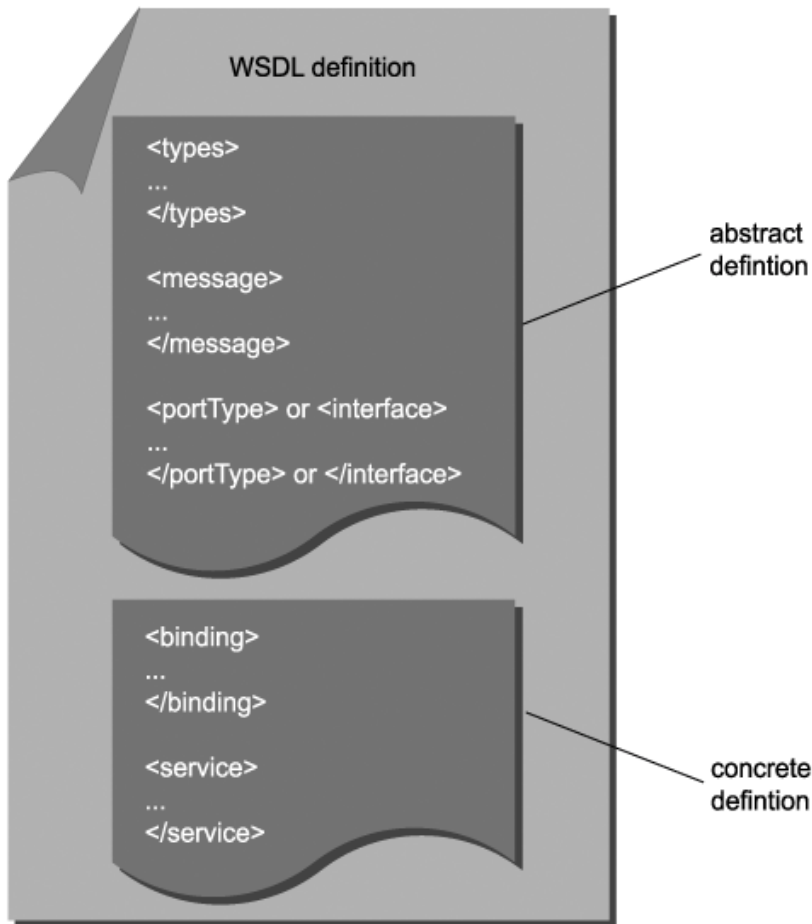
3. How do we derive at business services?

- Sources from which business services can be derived
 - The inner workings of any organization, regardless of structure or size, can be decomposed into a collection of business services. This is because a business service simply represents a logical unit of work, and pretty much anything any organization does consists of units of work.
 - What differs, though, is how organizations structure and document the work they perform.

4. What are WSDL elements.

- The Web Services Description Language (WSDL) is the most fundamental technology standard associated with the design of services.

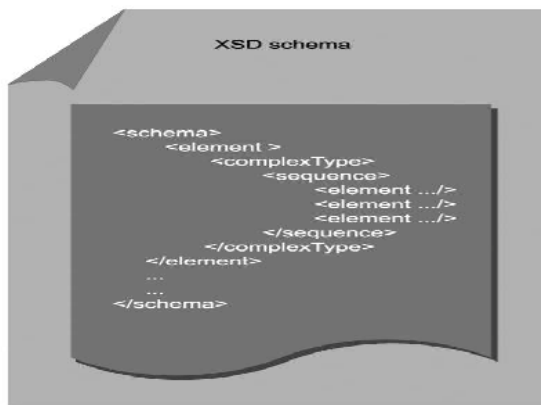
Figure 13.4. The structure of a WSDL definition.



5. Define XML Schema.

- The data types supported by XSD schemas are extensive, but they do not always map cleanly to the proprietary types used by programming languages.

Figure 13.3. An XSD schema document.



6. What is service , port and endpoint?

The service, port, and endpoint elements

- The service element simply provides a physical address at which the service can be accessed. It hosts the port element that contains this location information.

Example The service and port elements establishing the physical service address.

```
<service name="EmployeeService">
  <port binding="tns:EmployeeBinding" name="EmployeePort">
    <soap:address location="http://www.xmltc.com/tls/employee/">
    </port>
  </service>
```

7. What is SOAP Header?

The Header element

- The header portion of the SOAP message has become a key enabler of the feature set provided by WS-* specifications. Most of these extensions are implemented on a message level and introduce new standardized SOAP header blocks destined to be embedded in the Header construct.

Example The Header construct hosting a header block.

```
<Header>
  <x:CorrelationID xmlns:x="http://www.xmltc.com/tls/headersample/"
mustUnderstand="1"> 0131858580-JDJ903KD </x:CorrelationID>
</Header>
```

8. List SOA Composition Guidelines.

Step 1: Choose service layers

- Composing an SOA requires that we first decide on a design configuration for the service layers that will comprise and standardize logic representation within our architecture. This step is completed by studying the candidate service layers produced during the service-oriented analysis phase and exploring service layers and service layer configuration scenarios

Step 2: Position core standards

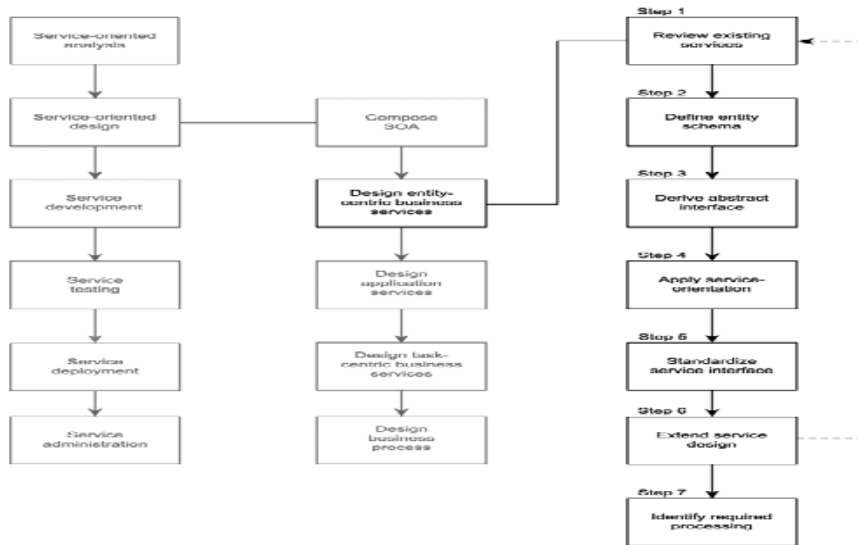
- Next, we need to assess which core standards should comprise our SOA and how they should be implemented to best support the features and requirements of our service-oriented solution.

Step 3: Choose SOA extensions

- This final part of our "pre-service design process" requires that we determine which contemporary SOA characteristics we want our service-oriented architecture to support. This will help us decide which of the available WS-* specifications should become part of our service-oriented environment.

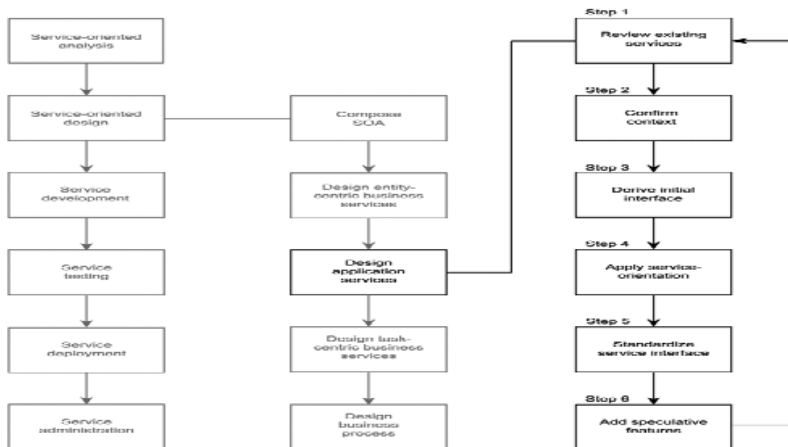
9. Draw the working of entity centric business service design process.

Figure 15.2. The entity-centric business service design process.



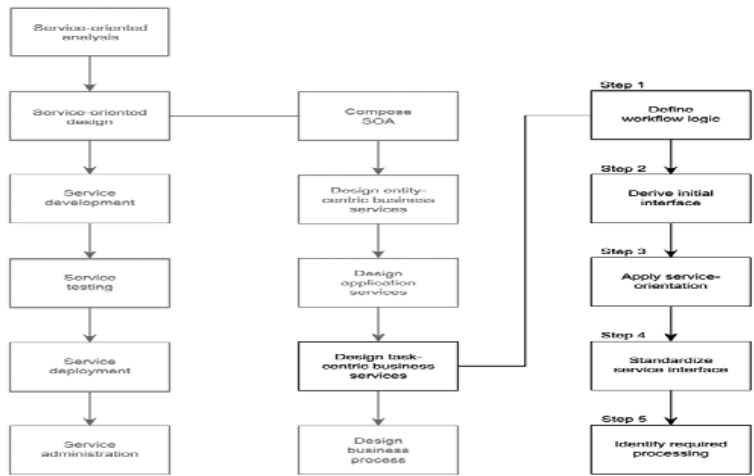
10. Draw the working of application service design step 2.

Figure 15.11. The application service design process.



11. Draw the working of task centric business service design process.

Figure 15.16. The task-centric business service design process.

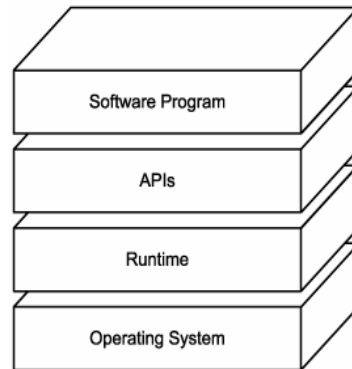


UNIT IV

2 marks

1. Draw the fundamental software technology architecture layers.

Figure 18.1. Fundamental software technology architecture layers.



2. Define Service Processing tasks.

- Fundamental service technology architecture
 - **Service processing tasks**
 - Service providers are commonly expected to perform the following tasks:
 - Supply a public interface (WSDL definition) that allows it to be accessed and invoked by a service requestor.
 - Receive a SOAP message sent to it by a service requestor.
 - Process the header blocks within the SOAP message.
 - Validate and parse the payload of the SOAP message.
 - Transform the message payload contents into a different format.
 - Encapsulate business processing logic that will do something with the received SOAP message contents.

- Assemble a SOAP message containing the response to the original request SOAP message from the service requestor.
- Transform the contents of the message back into the format expected by the service requestor.
- Transmit the response SOAP message back to the service requestor.

3. What is message processing logic.

- **Message Processing Logic** The part of a Web service and its surrounding environment that executes a variety of SOAP message processing tasks. Message processing logic is performed by a combination of runtime services, service agents, as well as service logic related to the processing of the WSDL definition.

4. Write about business processing logic.

Business Logic The back-end part of a Web service that performs tasks in response to the receipt of SOAP message contents. Business logic is application-specific and can range dramatically in scope, depending on the functionality exposed by the WSDL definition. For example, business logic can consist of a single component providing service-specific functions, or it can be represented by a legacy application that offers only some of its functions via the Web service.

5. List the various vendor platforms provided.

Let's now explore SOA support provided by both J2EE and .NET platforms. The next two sections consist of the following sub-sections through which each platform is discussed:

- Architecture components
- Runtime environments
- Programming languages
- APIs
- Service providers
- Service requestors
- Service agents
- Platform extensions

6. Write about JAX-WS communication between Client and Server.

- **JAX-WS** stands for Java API for XML Web Services. JAX-WS is a technology for building web services and clients that communicate using XML. JAX-WS allows developers to write message-oriented as well as RPC-oriented web

services. In JAX-WS, a remote procedure call is represented by an XML-based protocol such as SOAP. The SOAP specification defines the envelope structure, encoding rules, and conventions for representing remote procedure calls and responses.

- These calls and responses are transmitted as SOAP messages (XML files) over HTTP. Although SOAP messages are complex, the JAX-WS API hides this complexity from the application developer.
 - On the server side, the developer specifies the remote procedures by defining methods in an interface written in the Java programming language. The developer also codes one or more classes that implement those methods. Client programs are also easy to code.
 - A client creates a proxy (a local object representing the service) and then simply invokes methods on the proxy. With JAX-WS, the developer does not generate or parse SOAP messages. It is the JAX-WS runtime system that converts the API calls and responses to and from SOAP messages.

7. Write about requirements of JAX-WS Endpoint.

- With JAX-WS, clients and web services have a big advantage: the platform independence of the Java programming language. In addition, JAX-WS is not restrictive: a JAX-WS client can access a web service that is not running on the Java platform, and vice versa. This flexibility is possible because JAX-WS uses technologies defined by the World Wide Web Consortium (W3C): HTTP, SOAP, and the Web Service Description Language (WSDL). WSDL specifies an XML format for describing a service as a set of endpoints operating on messages.

8. Draw JAXB architecture with components.

Figure 2-1 shows the components that make up a JAXB implementation.

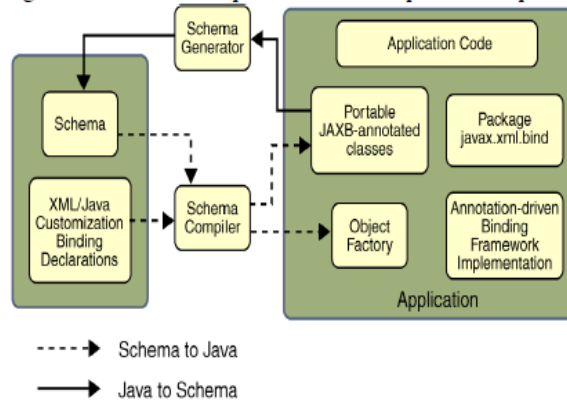


Figure 2-1 JAXB Architectural Overview

9. Depict with diagram the JAXB binding process.

Figure 2 2 shows what occurs during the JAXB binding process.

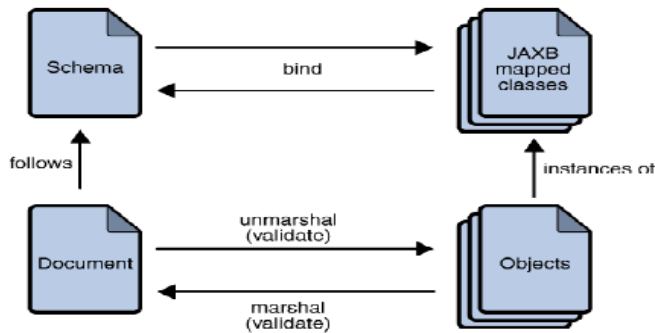


Figure 2 2 Steps in the JAXB Binding Process

10. Draw JAXR architecture.

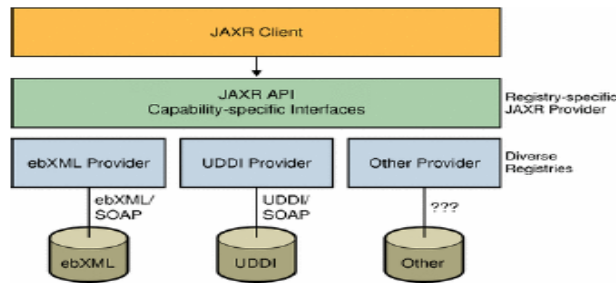


Figure 6-1 JAXR Architecture

11. What is purpose of JAXR API.

Purpose of the JAX-RPC Optional Package

- The Java API for XML-Based RPC (JAX-RPC) is an implementation of Remote Procedure Call (RPC) technology in the Java language, and is part of the Java™ 2, Enterprise Edition (J2EE™) platform. The JAX-RPC optional package subset, provided with the J2ME Web Services Reference Implementation, is a scaled-down version of JAX-RPC specifically tailored to the J2ME platform.
- The J2ME JAX-RPC subset allows Java developers to create portable web services clients that can easily exchange data between clients and back-end servers. Using XML and the Simple Object Access Protocol (SOAP), JAX-RPC allows developers to dispatch Remote Procedure Calls (RPC) to web services running on a different machine, a different network, or in a different language. Using RPC, developers can call methods on remote objects as easily as they can call them on local objects.

12. Draw WSIT stack.

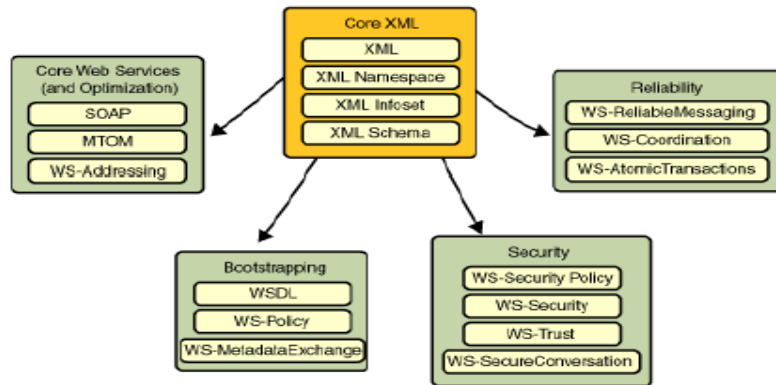


Figure 1-1 WSIT Web Services Features

13. What is WebService Security Policy and WebService Trust?

Further, WSIT implements two additional features to improve security in web services:

- a. **Web Services Security Policy** —Enables web services to use security assertions to clearly represent security preferences and requirements for web service endpoints.
- b. **Web Services Trust** —Enables web service applications to use SOAP messages to request security tokens that can then be used to establish trusted communications between a client and a web service.

14. What are advantages of CLR?

- There are several key advantages to employing Common Language Runtime and CIL listed below which are broken down into general advantages and application related advantages.
- **General advantages**
 - Note: Much of the general advantages are taken from Microsoft’s paper on CLR. The rest is from the Common Language Runtime Architecture paper from Microsoft.
- **Portability**
 - Using an intermediate language instead of compiling straight to native code requires $n + m$ translators instead of $n*m$ translators to implement it in n languages on m platforms.
- **Security**
 - The high level intermediate code is more ready for deployment and runtime enforcement of security and typing constraints than just low-level binaries used in other languages.
- **Interoperability**

Every major .NET language supports CLR and all get compiled to CIL. In that intermediate language, implementation of services such as security and garbage collection are the same. This allows one library or application of one .NET language to inherit implementations from classes written in another .NET language. This cuts down

on the redundant code developers would have to write to make a system work in multiple languages, allowing for multi-language system designs and implementations

15. Define WSE.

Web Services Enhancements for Microsoft .NET (WSE) is a .NET class library for building Web services using the latest Web services protocols, including WS-Security, WS-SecureConversation, WS-Trust, and WS-Addressing. WSE allows you to add these capabilities at design time using code or at deployment time through the use of a policy file.

UNIT V

1. What is BPEL?

the WS-BPEL language to demonstrate how process logic can be described as part of a concrete definition that can be implemented and executed via a compliant orchestration engine.

2. What are BPEL elements? Draw a neat sketch.

Figure 16.1. A common WS-BPEL process definition structure.



3. What are The partnerLinks and partnerLink elements in BPEL.

A `partnerLink` element establishes the port type of the service (partner) that will be participating during the execution of the business process. Partner services can act as a client to the process, responsible for invoking the process service. Alternatively, partner services can be invoked by the process service itself.

4. List the attributes of invoke element.

Table 16.1. invoke element attributes	
Attribute	Description
partnerLink	This element names the partner service via its corresponding partnerLink.
portType	The element used to identify the portType element of the partner service.
operation	The partner service operation to which the process service will need to send its request.
inputVariable	The input message that will be used to communicate with the partner service operation. Note that it is referred to as a variable because it is referencing a WS-BPEL variable element with a messageType attribute.
outputVariable	This element is used when communication is based on the request-response MEP. The return value is stored in a separate variable element.

5. What is WS-Coordination?

which can be used to realize some of the underlying mechanics for WS-BPEL orchestrations. Specifically, we describe some of the elements from the WS-Coordination specification and look at how they are used to implement the supplementary specifications that provide coordination protocols (WS-BusinessActivity and WS-AtomicTransaction).

6. What is coordination context in WS-Coordination?

The activation service returns this CoordinationContext header upon the creation of a new activity. As described later, it is within the CoordinationType child construct that the activity protocol (WS-BusinessActivity, WS-AtomicTransaction) is carried. Vendor-specific implementations of WS-Coordination can insert additional elements within the CoordinationContext construct that represent values related to the execution environment.

7. Write three specifications in WS-Policy.

- WS-Policy
- WS-PolicyAssertions
- WS-PolicyAttachments

8. Write about exactly once element.

This construct surrounds multiple policy assertions and indicates that there is a choice between them, but that one must be chosen.

Example 17.9. The ExactlyOne construct housing two alternative policy assertions, one of which must be used.

```
<wsp:Policy
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2002/12/policy">
  <wsp:ExactlyOne>
    <wsp:SpecVersion wsp:Usage="wsp:Required"
      wsp:Preference="10" wsp:URI=
        "http://schemas.xmlsoap.org/ws/2004/03/rm"/>
    <wsp:SpecVersion wsp:Usage="wsp:Required"
      wsp:Preference="1" wsp:URI=
        "http://schemas.xmlsoap.org/ws/2003/02/rm"/>
  </wsp:ExactlyOne>
</wsp:Policy>
```

9. What are UsernameTokens and BinarySecurityTokens.

The UsernameToken, Username, and Password elements (WS-Security)

The UsernameToken element provides a construct that can be used to host token information for authentication and authorization purposes. Typical children of this construct are the Username and Password child elements, but custom elements also can be added.

The BinarySecurityToken element (WS-Security)

Tokens stored as binary data, such as certificates, can be represented in an encoded format within the BinarySecurityToken element.