Arunai Engineering College

Tiruvannamalai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

5104 AEC

IT8074 - SERVICE ORIENTED ARCHITECTURE

PART A- 2 MARKS

UNIT I XML

XML document structure – Well-formed and valid documents – DTD – XML Schema – Parsing XML using DOM, SAX – XPath - XML Transformation and XSL – Xquery

2Marks

1. What is the goal of XML and its intended use?

XML can be **used to encode any structured information**, XML is good at representing information that has an ***extensible*****,** ***hierarchical*** **format and requires encoding of** ***metadata*****.** These three concepts form the basis of the XML language's structure and data model.

2. Define XML.(or) What do you mean by XML

XML stands for Extensible markup language **.**XML is a **meta-markup language** that provides a **format for describing structured data.** This facilitates more structured declarations of content and more meaningful search results across multiple platforms.

3. How does XML differ HTML

- HTML stands for Hyper Text Mark Up
- Several languages are derived from xml & wml
- HTML can be derived from xml.
- Xml uses indefinite, user defined, meaningful set of tags which can be used to include XML data in the webpage.
- HTML uses a fixed set of tags which can be used to specify the appearance of the webpage.

4. What do you mean by Unicode in XML?

Unicode standard is a **universal character set** whose goal is to **provide a definite encoding of the content of plain text** that can be **written in any and all languages** of the world. The latest version of Unicode, version 3.0, Because XML is a text-based language, it is dependent on characters and the representation of those characters.

5. How can we handle the Whitespace in XML

Whitespace is the term used for character spaces, tabs, linefeeds, and carriage returns in documents. They are generally used to make a document more readable. XML document contain two types of white spaces

- Significant Whitespace

- Insignificant Whitespace.

6. Give examples for significant and insignificant whitespaces.

A significant Whitespace occurs within the element which contain text and markup present together. **For example:**

<name>WelcomeXML</name>

and

<name>Welcome XML</name>

The above two elements are different because of the space between Welcome and XML. Insignificant whitespace means the space where only element content is allowed.

For example:

<**address.category="residence">** or

<address....category="..residence"> Here, the space is represented by dots (.)

7. What is XML namespace (Nov/Dec 2016)

- XML allows document authors to create custom elements.
- This extensibility can result in naming collisions (i.e. different elements that have the same name) among elements in an XML document.
- An XML namespace is a collection of element and attribute names. Each namespace has a unique name that provides a means for document authors to explicitly refer to elements with the same name (i.e. prevent collisions).

8. What is the difference between valid XML documents and well formed XML documents?

- **Valid XML documents**- W**e** must declare the document type to which they fulfill.
- **Well-formed XML documents –** We can include the DOCTYPE to simplify the task of the various tools that will be manipulating the XML document.

9. Difference between External and Internal DTD subset:

The only real difference between internally and externally defined DTD subsets is that the DTD content itself is contained within the square brackets, in the case of internal subsets, whereas external subsets save this content to a file for reference, usually with a .dtd extension.

10. What are the basic elements of XML (or) what are the markup contents of XML?

Markup and Content In general, six kinds of markup can occur in an XML document:

- Elements
- Entity references
- Comments
- Processing instructions
- Marked sections
- Document type declarations.

11. What do you mean by elements in XML ?

XML elements are either a matched pair of XML tags or single XML tags that are "self-closing."

For example : <shirt> </shirt>.

12. What are called as Entity References in XML?

Each **entity has a unique name** that is **defined as part of an entity declaration in a DTD or XML Schema.** Entities are used by simply referring to them by name. **Entity references are delimited by an ampersand** at the beginning and a semicolon at the ending

13. Difference between Internal and External Entities

Internal entities	**External entities**
Defined and used within the context of a document	Defined in a source that is accessible via a URI
Consists of simple string replacements	Consists of entire XML documents or non-XML text, such as binary files.
No need to define the type of file	Must define the type of the file

14. Write short notes on Document Type Definitions.

Document Type Definitions (DTDs) provide a means for **defining what XML markup can occur in an XML documen**t. It provides a mechanism to guarantee that a given XML document complies with a well-defined set of rules for document structure and content.

15. What is the use of content model in XML?

A **content model provides a framework** around which the **extensibility features of XML can be taken into advantage.**

Types of XML Content models

- Open Content Model
- Closed Content Model

- Mixed Content Model

16. List the Rules of Well formed XML documents (Nov/Dec 2016)

- All XML Elements Must Have a Closing Tag
- XML Tags Are Case Sensitive
- All XML Elements Must Have Proper Nesting
- All XML Documents Must Contain a Single Root Element
- Attribute Values Must Be Quoted
- Attributes May Only Appear Once in the Same Start Tag
- Attribute Values Cannot Contain References to External Entities
- All entities except amp, lt, gt, apos, and quot must be declared before they are used.

17. What do you mean by Default declaration of Namespace? Why do we use it?

A default namespace declaration specifies a namespace to use for all child elements of the current element that do not have a namespace prefix associated with them. <Customer xmlns="http://www.eps-software.com/po">

18. List out the Features of DTD (or) Importance of DTD.

- The **DTD** will **define the elements required by an XML** document, such as the **elements** that are optional, the **number of times an element should** (could) **occur**, and the **order** in which **elements should be nested**.
- DTD markup also defines **the type of data that will occur in an XML** element and the attributes that may be associated with those elements.
- **A document**, even if well formed, is **not considered valid if it does not follow the rules defined in the DTD.**

19. What are the two different Types of DTD:

- **Internal DTD -** residing within the body of a single XML document
- **External DTD -** referenced by the XML document

20. What are the basic elements in Structure of a Document Type Definition?

The structure of a DTD consists of:

- **Document type declaration**
- **Elements**
- **Attributes**

- **Entities**
- **Other keywords**

21. Write some of the DTD Element Rules?

Content Rules - The content rules for .elements deal with the actual data that defined elements may contain. These rules include the

The ANY Rule. The **element may contain other elements and/or normal character data**.

The EMPTY Rule An element that is defined with this rule will **contain no data**.

The #PCDATA Rule Parsed character data is data that **may contain normal markup** and will be **interpreted and parsed by any XML parser** accessing the document.

22. Is it possible to restrict the xml file from parsing? If so how can it be performed?

It is possible in an element using the #PCDATA rule to use the **CDATA keyword to prevent the character data from being parsed.**

23. What are DTD Entities and list the types of DTD Entities

Entities in DTDs are **storage unit**. Entities are **special markups** that contain **content for insertion into the XML document**. An entity's content could be well-formed XML, normal text, binary data, a database record, and so on.

Types of Entities:

- Predefined Entities
- External Entities
- Parameter Entities

24. What are the DTD Drawbacks?

- DTDs are composed of non-XML syntax
- DTDS are not object oriented. There is no inheritance in DTDs.
- DTDs do not support namespaces very well. For a namespace to be used, the entire namespace must be defined within the DTD.
- DTDs have weak data typing and no support for the XML DOM.
- DTDs can be overridden there is possible for security issues.

25. Why do we need XML Schema? How it helps in defining the structure of XML?

A more **powerful way of defining the structure and constraining the contents of XML documents.** An XML Schema definition is itself an XML document

26. What are the elements of the XML Schema Structure?

- Elements
- XSD Simple Elements
- XSD Complex Elements
- Attributes
- XSD Restrictions

27. What do you mean by XSD Simple Elements

A simple element is an XML element **that contains only text**. It **cannot contain any other elements or attributes.** The text can be of many different types. It can be one of the types included in the XML Schema definition (boolean, string, date, etc.), or it can be a custom type

28. What do you mean by XSD Complex Elements?

A complex element contains other elements and/or attribute.

There are four kinds of complex elements:

- Empty elements
- Elements that contain only other elements
- Elements that contain only text
- Elements that contain both other elements and text

29. Write about the XSD Indicators?

We can control HOW elements are to be used in documents with indicators. There are seven indicators:

Order indicators: used to define the order of the elements.

Occurrence indicators: used to define how often an element can occur.

Group indicators: used to define related sets of elements.

30. Define XPath

The XML Path Language (XPath) is a **standard for creating expressions that can be used to find specific pieces of information within an XML document**. XPath expressions are used by both XSLT (for which XPath provides the core functionality) and XPointer to locate a set of nodes.

31. What are the different types of Nodes in Xpath?

In XPath, there are **seven kinds of nodes**:

- element

- attribute
- text
- namespace
- processing-instruction
- comment,
- Document nodes.

XML documents are **treated as trees of nodes.** The topmost element of the tree is called the root element.

32. What are Predicates? How are they used in Xpath?

Predicates are used to find a specific node or a node that contains a specific value. Predicates are always embedded in square brackets.

33. Write about the Location Path Expression in Xpath.

A location path can be absolute or relative.

An **absolute location path**: **/step/step/...**

A **relative location path:** **step/step/...**

34. Define Xpointers.

Xpointers **address the individual parts of an XML document**. The node tests for an XPointer are, the most part, the same as for an XPath node test. However, in addition to the node tests already listed for XPath expressions, XPointer provides two more important node tests:

- point()
- range()

35. What are the different types of Points in Xpointer?

XPointer points to **allow a context node to be specified and an index position** indicating how far from the context node the desired point is. Two different types of points can be represented using XPointer points:

- **Node points- location points in an XML document** that are nodes that **contain child nodes.**
- **Character points-** When the **origin node is a text node**, the index **position indicates the number of characters.**

36. Define XLink

The XML Linking Language, allows a **link to another document to be specified on any element within an XML document.** The XML Linking Language

creates a **link to another resource through** the, not through the actual elements **use of attributes specified on elements** themselves.

37. What is XML document Prolog?

The prolog is an optional component of the XML document. If included, the prolog must be appearing **before** the root element. A prolog consists of two parts: the XML declaration and the Document Type Declaration (DTD). Depending on your needs, you can choose to include both, either, or neither of these items in your XML document. The DTD is most often used, so we will discuss its use and purpose first.

38. Define a XML Parser. (Nov/Dec 2016)

It provides a **way of representing an XML document in memory**.

Two types of XML parsers:

- **Non validating parsers** merely **read XML documents** and **verify** that the documents are **well formed**.
- **Validating parsers** read **well-formed documents** also their fulfillment against a **DTD, XML Schema, or other validation set**.

39. Mention any three XML Parsers based on processing?

- **SAX** (Simple API for XML) Parser – Event based Parsing
- **DOM** (Document Object Model) Parser – Tree based Parsing
- **XSLT** (XML Style Sheet) Parsers – Style based Parsing

40. Write in brief about Document object Model or DOM Parser.

DOM converts **XML documents into a programmatic object model** that can then be used and processed by standard applications. Once the XML document has been parsed, an object model representation exists in the memory of the processing client.

41. What are the two levels of interface available in DOM?

The Document Object Model offers two levels of interface implementation:

- **DOM Core,** which supports basic XML document integration
- **DOM HTML,** which extends the model to HTML documents.

42. Write in brief about SAX Parser.

Event-based parsers such as SAX provide a view of XML documents that is **data centric and event driven.** When a user reads an XML document using SAX, elements that are encountered by the parser are read, processed, and then forgotten. The event-based parser **reads the elements from the document** and **returns them**

to the application **with a list of attributes and content.**

43. What are the steps followed in DOM Parsing?

Following are the steps used while parsing a document using DOM Parser.

- Import XML-related packages.
- Create a DocumentBuilder
- Create a Document from a file or stream
- Extract the root element
- Examine attributes
- Examine sub-elements

44. List some of the nodes in DOM

- Document node
- Element Node
- Attribute node
- Text node
- CDATA node
- Processing instruction node

45. Compare DOM and SAX in XML processing.

DOM	**SAX**
DOM is tree based parsing method used to parse the given XML document	It is event based processing method used to parse the given XML document
In this method the entire XML document is stored in the memory before actual processing. Hence it requires more memory	In this method the parsing is done by generating the sequence of event or its calls handler functions
We can insert or delete node	We can insert or delete node
Traversing is done in any direction in DOM approach	Top to bottom traversing is done in this approach

46. What are the various technologies used in styling the XML.

XSL Technologies XSL has two independent languages:

• **The XSL Transformation Language (XSLT)** - used to convert an XML document to another format.

• **The XSL Formatting Object Language (XSL-FO**) - provides a way of describing the presentation of an XML document.

47. Write a simple style sheet using XML.

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
 <html>
 <body>
  <h2>My CD Collection</h2>
  <table border="1">
   <tr bgcolor="#9acd32">
    <th>Title</th>
    <th>Artist</th>
   </tr>
  </table>
 </body>
 </html>
</xsl:template>
</xsl:stylesheet>
```

48. Write about the XSL Formatting Objects

XSL-FO was designed to assist with the printing and displaying of XML data. The main importance is on the document layout and structure. This includes the dimensions of the output document, including page headers, footers, and margins.

Two techniques for creating XSL-FO documents.

- Develop the XSL-FO file with the included data.
- Dynamically create the XSL-FO file using an XSLT translation.

49. How can we model the database in XML? (or) Write in brief about JAXB.

We can use the XML data binding features of Java Architecture for XML Binding (JAXB). JAXB provides a framework for representing XML documents as Java objects.

50. What do you mean by Marshalling and Unmarshalling in JAXB?

In the JAXB framework, we can parse **XML documents into a suitable Java object.** This technique is referred to as **unmarshaling**. The JAXB framework also provides the capability to **generate XML documents from Java objects**, which is

referred to as **marshaling.**

51. What is Schema Compiler?

JAXB provides a schema compiler for generating the Java source files. The schema compiler takes as input the DTD and the JAXB binding schema.

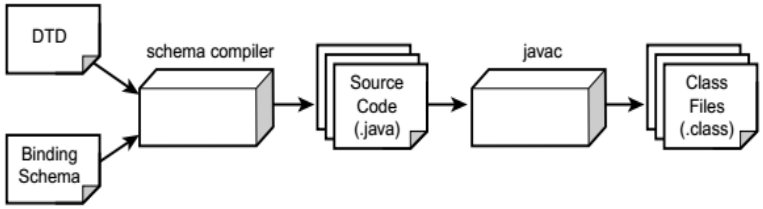


FIGURE 10.6
Generating Java classes with the JAXB compiler.

52. How can we bind the XML schema to Java class (or) JAXB binding schema?

The JAXB binding schema contains instructions on how to bind the XML schema to a Java class. The steps followed are:

1. Review the database schema.
2. Construct the desired XML document.
3. Define a schema for the XML document.
4. Create the JAXB binding schema.
5. Generate the JAXB classes based on the schema.
6. Develop a Data Access Object (DAO).
7. Develop a servlet for HTTP access.

53. What is the need of Data Access Object (DAO)?

A Data Access Object (DAO) provides access to the backend database. The goal of the DAO design pattern is to provide a higher level of abstraction for database access. The DAO provides access to the backend database via public method.

54. List the Disadvantages of SAX.

- SAX parsing is “single pass,” so we can’t back up to an earlier part of the document any more than you can back up from a serial data stream.
- SAX implementations are read-only parsers. They do not provide the ability to manipulate a document or its structure.
- There is no formal specification for SAX.

UNIT – II SERVICE ORIENTED ARCHITECTURE (SOA) BASICS

Characteristics of SOA, Comparing SOA with Client-Server and Distributed architectures – Benefits of SOA -- Principles of Service orientation – Service layers.

Part - A

1. What is architecture? (Nov/Dec 2013)

- Architecture refers a systematic arrangement of computerized automation technological solutions.
- It is to an application development team, a **blueprint to the team of construction workers**. Different organizations document different levels of application architecture.

2. Define Service-oriented architecture? (NOV/DEC 2011) (MAY/JUN 2012) (May/Jun 2013) (Nov/Dec 2013) (May/Jun 2015)

SOA can establish an **abstraction of business logic and technology** that may introduce changes to **business process modeling and technical architecture**, resulting in a loose coupling between these models.

3. Define contemporary SOA.

Contemporary SOA represents an architecture that promotes service-orientation through the use of web services. Contemporary SOA is an **extended variation of service-oriented architecture** which builds increasingly **powerful XML and Web services support** into current technology platforms.

4. List out some characteristics of contemporary SOA (Nov/Dec 16 & 17)

- Contemporary SOA is at the core of the services-oriented computing platform.
- Contemporary SOA increase quality of service.
- Contemporary SOA is fundamentally autonomous.
- Contemporary SOA is based on open standards.
- Contemporary SOA supports vendor diversity.
- Contemporary SOA fosters intrinsic interoperability.
- Contemporary SOA promotes discovery.
- Contemporary SOA promotes federation.
- Contemporary SOA promotes architectural compos ability.
- Contemporary SOA fosters inherent reusability.

5. What are the benefits of SOA?

- Improved integration(and intrinsic interoperability)
- Inherent reuse
- Streamlined architectures and solutions
- Leveraging the legacy investment
- Establishing standardized XML data representation.
- Focused investment on communications infrastructure
- Organizational agility(Flexibility)

6. What are the common mistakes/pitfalls of adopting SOA?

- Building service –oriented architecture like traditional distributed architectures
- Not standardizing SOA
- Not creating a transition plan
- Not starting with an XML foundation architecture
- Not understanding SOA performance requirements
- Not understanding web services security
- Not keeping in touch with product platforms and standards development

7. What are the requirements is needed to fill QoS gaps between contemporary and primitive SOA?

Contemporary SOA is striving to fill the QoS gaps of the primitives SOA models with the following requirements

- In a secure manner
- Reliably
- With appropriate performance
- Protecting business integrity
- Executing exception logic in case of failure

8. What is autonomous principle?

Autonomous principle represents the **ability of a service to carry out its logic independently** of outside influences

9. List out the different levels of autonomy?

- Runtime autonomy- Run time autonomy represents the **amount of control a service has over its execution environment** at runtime.

- Design-time autonomy- Design-time autonomy represents the **amount of governance control** a service owner has over the **service design.**

10. What are the design characteristics require facilitating interoperability in contemporary SOA?

The designs characteristics require to facilitate interoperability are

- Standardization
- Scalability
- Behavioral predictability
- Reliability

11. List out some of the characteristics of the reusable services in SOA?

Reusable services have the following characteristics:

- Define by an unbelievable functional context
- Logic is highly generic
- Has a generic and extensible contract
- Can be accessed concurrently

12. How is loose coupling concept achieved in SOA?

The loose coupling concept is achieved by implementing standardized service abstraction layers when **service-orientation principles** are **applied to both business modeling and technical design.**

13. What is referred as organizational agility?

Organizational Agility refers to efficiency with which an organization can respond to change.

14. What is application architecture?(or) Define Application architecture (May/June 2014 & Nov/Dec 2014)

Application architecture is a **template** for all others specifically explained the **technology, boundaries, rules, limitations, and design** characteristics that apply to all solution based on the template.

15. What is Enterprise architecture?

Enterprise architecture is a **creation of master specification** when numerous, disparate and **integrate application architectures** exists within an organization.

16. What is single-tier client-server architecture?

Single-tier client-server architecture is an environment in which **bulky mainframe back-ends server served the thin clients.**

17. List out the primary characteristics of the two tier client serverarchitecture?

- Application logic
- Application processing
- Technology
- Security
- Administration

18. What is multi-tier client-server architecture?

Multi-tier architecture is a client-server architecture in which the **presentation, the application processing, and the data management** are logically **separate** processes.

19. List out the types of communication of mainframe systems?

- **Synchronous communication**- It allows the **client and server to wait for each other** to transfer the message. That is, the client will not continue until the server has received the message.
- **Asynchronous communication**- It allows the server to **continuously receive messages from the client without waiting** for the server to respond.

20. What are the key benefits of service reuse?

- Accommodate future requirements with less development effort.
- Reduce the need for creating wrapper services
- Reduction of cost by not just avoiding duplication of code
- Reducing risks by reusing well-tested code and runtime environments.

21. State separation of concerns.

It is an established software engineering theory based on the **idea of breaking down a large problem** into a series of **individual concerns.**

22. What are the parts of automation logic? (May/Jun 2013)

The four identified parts of automation logic

- **Messages**=units of communication
- **Operations**=units of work
- **Services**=units of processing logic(collection of units of work
- **Processes**=units of automation logic(coordinated aggregation of units of works)

23. What are the issues that are raised in the client-server and the distributed internet architecture?

The issues that are raised in the client-server and the distributed internet architecture comparisons are discussed in a comparison between multi-tier client-server and SOA.

- Application logic
- Application processing
- Technology
- Security
- Administration

24. What is the use of RPC? Write down the advantages of RPC?

The remote procedure call (RPC) connections are used for **remote communication between** components residing on **client workstations and servers**.

- **Better load balancing:** more evenly distributed processing
- **More scalable:** only servers experiencing high demand need be upgraded.
- **Multiple concurrent requests** are processed.

25. Write down the disadvantages of RPC?

- Heavily loaded network
- More distributed processing necessitates more data exchange
- Difficult to program and test due to increased complexity

26. What are the proprietary communication protocols promotes by distributed internet architecture for usage?

- DCOM – Distributed Component Object Model
- CORBA – Common object request broker architecture

27. What's the difference between services and components?

- Services are **logical grouping of components** to achieve business functionality.
- Components are **implementation approaches** to make a service.

28. Define Client / Server architecture

- The **Mainframe back-ends served thin clients**, are considered an implementation of the single-tier client-server architecture.
- Mainframe systems natively supported both synchronous and asynchronous communication. Only upon certain conditions the server actually respond.

29. Define Distributed Internet architecture

- It distributes the **application logic among multiple components** so to reduce the server load

- Server-side components, now located on dedicated application servers, would then share and manage pools of database connections, to **reduce the burden of concurrent usage on the database server.**
- A single connection could easily facilitate multiple users.

30. Define SOA Characteristics (Apr/May – 2017)

- 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.

31. Define Coarse-Grained Services. (NOV/DEC 2012) (May/June 2015)

- A service-based system controls the network access to the objects within the service through a set of coarse-grained interfaces.
- A service may still be implemented as a set of fine-grained objects, but the objects themselves are not accessible over a network connection. A service implemented as objects has one or more coarse-grained objects that act as distributed look.

32. Define Service Component

- The Service Component is that **logical unit of code** which **implements** the functionality to support the Service, they are one or more Services.
- A Service Component is also usually associated with a data store of fundamental **data type, control data, process, data,** etc depending on the nature of the particular service component.

33. Define Service-component-level Testing (or) Unit Testing

- **Service-component-level testing** or **Unit testing,** is normally performed by the developers to test that the code successfully compiles, also check the basic functionality of the components and functions within a service
- The primary goal of Component testing is to **take a piece of code,** determine whether it behaves exactly required. Each Component is **tested separately**

before integrating it into services.

34. Define Process/Orchestration-level Testing

Process/Orchestration testing ensures **services are operating collectively** as specified. This phase of testing would **cover business logic, sequencing, exception handling and process decomposition** (including service and process reuse).

35. Define Security Testing

SOA combined with Government and Regulatory compliance, will require Security testing activities to be incorporated into the entire project life cycle.

36. Explain Legacy System Adapter

- The primary function of Legacy System Adapter is to **convert to/from data formats spoken by the legacy system** and the common data formats used by the Foundational Service Components.

37. Define Composite Service Component

- A Composite Service Component is a component which **combines the functionality of one or more other Foundational or Composite Service Components.**
- It may also encapsulate additional functional and data enrichment, business process.

38. Explain about QOS(or) Quality of Service.

Each SOA Services have QOS the key QoS elements are **security requirements**, such as **authentication and authorization**, **reliable messaging**, and policies regarding who can invoke services.

39. Define Service Proxy?

- The **service provider supplies** a service proxy to the **service consumer.**
- The **service consumer executes** the request by **calling an API function** on the proxy. It then formats the request message and executes the request on behalf of the consumer.

40. List the Principles of service orientation (Nov/Dec 2014)

- Services are reusable
- Services share a formal contract
- Services are loosely coupled
- Services abstract underlying logic
- Services are compos able

- Services are autonomous
- Services are stateless
- Services are discoverable

41. Define Service-Orientation and Interoperability

Service-oriented computing, is that **services must be interoperable**. Each of the eight SOA principles supports or contributes to interoperability in some manner.

42. Define service loose coupling

The Service Loose Coupling promotes the **independent design and evolution of a service's logic and implementation** while still guaranteeing baseline interoperability with consumers that have come to rely on the service's capabilities.

43. What are the logical components of web service frameworks? (MAY/JUN 2012)

- Web Services
- Web Service operations
- SOAP Messages
- Activities.

44. "Service are autonomous", comment. (MAY/JUN 2012)

- Represents the **ability of a service to carry out its logic independently** of outside influences.
- **Message-level autonomy**- Messages are "intelligence-heavy" and control the way they are processed by recipients
- Autonomy concept is expanded to solution environment and the enterprise i.e. applications.

45. Define Application Services layer.

- Application services layer holds **applications exposed as services**, newly added services, and legacy applications wrapped by standard Web services interface. Services at Application Services layer are set of stateless Web services that perform certain task(s).
- **Services** of **Application Services layer** are **reusable** among different business processes, can be integrated in new applications, and can be extended address new business processes.

46. Define Business layer.

- Business layer is responsible for supporting business process life cycle.

Business process lifecycle consists of **five stages: design, model, simulate, monitor, manage, and optimize business processes.**

- Business layer users are either business analysts, or business managers.

47. Define Business services layer.

- Business services layer holds orchestration and choreography engines under governance mechanisms to map business processes to composing Web services. Orchestration and choreography engines are the mapping enablers of business processes into executable services.
- Orchestration and choreography engines maintain business process workflow logic, performance requirements, and system/user state.
- Business services layer has access to business rules repository

48. What are the fundamental parts of SOA Framework?

The following fundamental parts of the framework:

- SOAP messages
- Web service operations
- Web services
- Activities

49. What is UDDI?

- Universal description discovers and integration**(UDDI)** provided for service registries.
- Some SOA systems used **UDDI service registry** or directory to **manage service descriptions.**

50. List any four pitfalls of SOA.

- SOA does not solve complexity automatically
- Big Design Upfront
- Incorrectly applied Canonical Data Model
- Unclear ownership/Project based funding
- Ignoring culture when introducing SOA

UNIT III WEB SERVICES (WS) AND STANDARDS

Web Services Platform – Service descriptions – WSDL – Messaging with SOAP – Service discovery – UDDI – Service-Level Interaction Patterns – Orchestration and Choreography

PART - A

1. What is a web service? (NOV/DEC 2013) (May/June 2015)

Definition 1:

A **web service** is used to **implement architecture** according to service-oriented architecture **(SOA) concepts**. The basic unit of **communication** is a **message.**

Definition 2:

- Web services are application components.
- Web services communicate using open protocols.
- Web services are self-contained and self-describing.
- Web services can be discovered using UDDI .
- Web services can be used by other applications XML is the basis for Web services.

2. Define service.

A **service** is a unit of **software capable of altering its role**, depending on its processing responsibility in a given scenario.

3. What are the fundamental roles of service?

- Service provider
- Service requestor
- Intermediaries
- Initial sender and ultimate receiver
- Service composition

4. What is referred to as service description documents?

The individual documents that comprise a service contract are referred to as service description documents.

5. What do service endpoints provide?

Service endpoint provides a formal definition of the endpoint interface and also establishment the physical location of the service.

6. What are service descriptions?

A WSDL endpoint description explains how the service description document itself is organized. It is also known as WSDL service definition or just WSDL definition.

7. What are the categories of service descriptions?

Service descriptions is divided into two categories

- Abstract description
- Concrete description

8. What does abstract description establish?

An abstract description establishes the interface characteristics of the web service without any reference to the technology used to host or enable a web service to transmit messages.

9. What are the parts that comprise an abstract description?

The three mail parts that comprise an abstract description are

1. Port type
2. Operation
3. Message

10. Define WSDL.

SOA services have **self-describing interfaces** in platform-independent XML documents. Web Services Description Language **(WSDL)** is the **standard** used to **describe the services.**

11. What does port type in abstract description provide?

Port type provides a high-level view of the service interface by sorting the messages a service can process into groups of functions.

12. Define concrete description.

The concrete description portion of the WSDL file defines the connection needed from the abstract web service interface to a physical transport protocol.

13. What are the parts that comprise concrete description?

The three main parts that comprise concrete description are

- Binding
- Port
- Service

14. What is metadata?

Metadata provides information about the service.

15. What is the use of SOAP? (MAY/JUN 2012)

The simple object access protocol (SOAP) is used to **define a standard message format** which is used **for communication** between services running on different operating systems.

16. List out some of the characteristics of SOAP messaging framework?

SOAP messaging framework has the following three characteristics that is

- Extensible
- Interoperable
- Independent

17. What are the parts of SOAP message? (MAY/JUN 2013)

SOAP message consists of three parts:

- SOAP envelope
- SOAP header
- SOAP Body
- SOAP Fault

18. List out messaging styles offered by SOAP.

- RPC(Remote procedure call)
- Document-style

19. Sketch the anatomy of a SOAP message (Nov/Dec 2016)

```
<? 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>
                  …….
      </soap:Header>
      <soap:Body>
        …….
            <soap:fault>
              …..
            </soap:fault>
       </soap:Body>
</soap:Envelope>
```

20. What is a SOAP node?

The programs that services use to transmit and receive SOAP messages are referred to as SOAP nodes.

21. List out the SOAP elements in Communication.

- SOAP sender
- SOAP receiver
- SOAP intermediary
- Initial SOAP sender
- Ultimate SOAP receiver

22. What is called the SOAP message path?

The **route taken by the message** is called the SOAP message path. The set of SOAP nodes through which the SOAP message passes, including the initial sender, the ultimate receiver and one or more intermediaries, are called the SOAP message path.

23. Define UDDI?

SOA services are **maintained** in the **enterprise by a registry** that acts as a directory listing. Applications can look up the services in the registry and invoke the service. Universal Description, Definition, and Integration **(UDDI)** is the **standard** used **for service registry.**

24. Define message exchange pattern.

Message exchange pattern (MEP) defines the **way that SOAP message are exchanged** between the web service requester and web service provider. It represents a set of templates.

25. List out some primitive MEPs. (Nov/Dec 2016)

A common set of primitive MEPs are listed below

- Request-response
- Fire-and-forgot
- Complex MEPs

26. What is publish-and-subscribe pattern?

Publish-and-subscribe pattern is an asynchronous MEP in which publisher's sends messages to all interested subscribers.

27. Write down some of the pattern supported by WSDL?

- In-out pattern
- Out-in pattern
- In-only pattern
- Out-only pattern

- Robust in-only pattern
- Robust out-only pattern
- In-optional-out pattern
- Out-optional-in pattern

28. List the Web services platform elements: (NOV/DEC 2012) (NOV/DEC 2013)

1) SOAP (Simple Object Access Protocol)
2) UDDI (Universal Description, Discovery and Integration)
3) WSDL (Web Services Description Language)

29. Define Web API.

- Web API is a development in Web services (in a movement called Web 2.0) where emphasis has been moving away from SOAP based services towards Representational State Transfer (REST) based communications.[3] REST services do not require XML, SOAP, or WSDL service-API definitions.
- Web APIs allow the combination of multiple Web services into new applications known as mashups.

30. Define Agents and Services.

- A **Web service** is an abstract notion that must be implemented by a concrete agent.
- The **agent** is the concrete piece of software or hardware that sends and receives messages, while the service is the resource characterized by the abstract set of functionality that is provided.
- Although the agent may have changed, the Web service remains the same.

31. Define Soap message.

A SOAP message is specified as an XML Information Set . While all SOAP message examples in this document are shown using XML 1.0 syntax, other representations MAY be used to transmit SOAP messages between nodes.

32. Define Request-Response.

Request-Response is a pattern in which the service consumer uses configured client software to issue an invocation request to a service provided by the service provider. The request results in an optional response.

33. Define Subscribe-Push.

- A third pattern for interaction is called Subscribe-Push. In this pattern, one or more clients register subscriptions with a service to receive messages based on

some criteria.

- Subscriptions may remain in effect over long periods before being canceled or revoked. A subscription may, in some cases, also register another service endpoint to receive notifications.

34. Define Atomic Service Transaction.

Atomic Service Transaction is the name of a design pattern authored by Thomas Erl and published as part of the SOA Design Patterns catalog. Within the catalog this pattern is further categorized as one of the Composition Implementation Patterns. The icon used to identify Atomic Service Transaction is:

35. List the Atomic Transaction Protocols.

- ✓ A Completion protocol which is typically used to initiate the commit or abort states of the transaction.
- ✓ ***The Durable 2PC protocol*** for which services representing permanent data repositories should register.
- ✓ ***The Volatile 2PC protocol*** to be used by services managing non-persistent (temporary) data.

36. Define atomic transaction coordinator.

The atomic transaction coordinator plays a key role in managing the participants of the transaction process, and in deciding the transaction's ultimate outcome.

37. List out the characteristics of ACID transactions

The acronym ACID refers to the four key properties of a transaction:

1. **Atomicity**

   It follows an all or nothing rule. Either all changes or no changes succeed.

2. **Consistency**

   System restored to a constant state after completion.

3. **Isolation**

   Multiple transaction don't interface

4. **Durability**

   Changes made as part of transaction survive subsequent failures.

38. Define Orchestration.

- Refers to an **executable business process** that may **interact** with both i**nternal**

and **external Web services.**

- Orchestration describes how Web services can interact at the message level, including the business logic and execution order of the interactions. These interactions may span applications and/or organizations, and result in a long-lived, transactional process.
- With orchestration, the process is always controlled from the perspective of one of the business parties.

39. Define Choreography.

More collaborative in nature, where **each party involved in the process describes the part they play in the interaction**. Choreography tracks the sequence of messages that may involve multiple parties and multiple sources. It is associated with the public message exchanges that occur between multiple Web services.

40. Define Orchestration service.(Apr/May – 2017)

- ✓ **Orchestration** services encapsulate the entire business process.
- ✓ **For example,** a service containing the entire flow of the Employer Registration" business process is an orchestration service. The complete registration process identifying the employer type, determining liability and registering the employer in the database and various other steps.

41. Compare Abstract and Concrete description.

Abstract Description	**Concrete Description**
It establishes the interface characteristics of the web service without any reference to the technology used to host or enable a web service to transmit messages.	It establishes the portion of the WSDL file defines the connection needed from the abstract web service interface to a physical transport protocol.
The three mail parts that comprise an abstract description are • Port type • Operation • Message	The three main parts that comprise concrete description are • Binding • Port • Service

42. Differentiate Orchestration form activities. (MAY/JUN 2013)

Orchestration services encapsulate the entire business process. For example, a service containing the entire flow of the Employer Registration" business process is an orchestration service.

WS-BPEL breaks down workflow logic into a series of predefined primitive activities

Basic activities represent fundamental workflow

- Invoke
- Receive
- Reply
- Throw
- Wait

A single orchestration scope can be classified as a complex, and most likely, long-running activity.

UNIT IV WEB SERVICES EXTENSIONS

WS-Addressing - WS-Reliable Messaging - WS-Policy – WS-Coordination – WS - Transactions - WS-Security – Examples

PART - A

1. What is the process element?

An <process> element is the **root element** and must have a **name attribute for** assigning the **name value**. It is used to establish the process definition-related namespace.

2. What does the partner links element define?

The partner Links define the services that are orchestration by the process. It contains a set of <partner link> element each **represent** the **communication exchange between two partners**-the process service being one partner and another service being the other.

3. What are the attributes in the partnerlink element?

- myRole
- partnerRole

4. What is the use of myRole attribute?

- Used when the process service is invoked by a partner client service.
- Process service acts as the service provider.

5. What is the use of partnerRole attribute?

- Identifies the partner service that the process service will be invoking.
- Partner service acts as the service provider.

6. What is the use of partnerLinkType element?

The partnetLinkType elements are used to identify the WSDL portType elements reference by the partnerLink elements within the process definition.

7. What does the variables element hold?

Variables hold the data that constitute the state of a BPEL business process during runtime.

8. Give the syntax for getvariableproperty function. (Nov/Dec 2014)

Syntax

Getvariableproperty (variable name, property name)

Example

Getvariableproperty (ticketapproval,'class')

9. List out the attributes of the variables element.

- Message type
- Element
- Type

10. What is the getvariableproperty function?(Nov/Dec 2014)

The get variable property function allows global property values to be retrieved from variables. It simply accepts the variable and property names as input and returns the requested value.

11. What is the getvariabledata function? (Nov/Dec 2014)

The getvariabledata function has a mandatory variable name parameter and two optional arguments that can be used to specify a part of the variable data.

12. Give the syntax for getvariabledata function. (Nov/Dec 2016)

Syntax

Getvariabledata (variable name, part name,location path)

Example

Getvariabledata ('input','payload','/tns:timesheetType/House/....')

13. What is the use of invoke statement? What are the five common attributes equipped with invoke element?

The <invoke> activity is used to involve the web service operation provide by partners.

- partnerLink
- portType
- operation
- inputVariable
- outputVariable

14. What is the use of the receive element?

A <receive> activity is used to receive requests in a BPEL business process to provide services to its partners. The process block until the message is received.

15. What is the reply element?

A <reply> activity is used to send a response to a request previously accepted through a <receive> activity. Responses are used for synchronous request/ reply interactions.

16. What is the purpose of the sequence element?

The sequence construct is to organize a series of activities so that they are executed in a predefined, sequential order. Nesting of sequence is allowed.

17. Give the structure of sequence element.

<sequence>
<receive>....</receive>
<assign>....</assign>
<invoke>...</invoke>
<reply>.....</reply>
</sequence>

18. Write down the syntax for switch, case and otherwise element. (NOV/DEC 2011)

<switch>
<case condition=
"getvariableData('EmployeeResponseMessage','Responseparameter')=0">
.....
</case>
<otherwise>
</otherwise>
</switch>

19. What is the use of assign activity?

The <assign> activity is used to:

- Copy data from one variable to another
- Construct and insert new data using expressions and literal values
- Copy partner link endpoint references

20. Give the overview of WS-coordination.

WS- coordination is a framework for coordinating distributed activities

- **Coordinator**
  - Activation service for creating coordination instance
  - Registration service for registering participating applications

- Additional protocol specific services

- **Set of coordination protocols**

21. What is the use of coordination context element?

The coordination context is used to carry information about active coordination to participants

- Information inside context is coordination protocol specific.
- Context format is not mandated by the standard.
- Typically passed in SOAP headers.

22. What is WS-policy? (NOV/DEC 2011) (NOV/DEC 2012)(May/June 2015)

WS-policy **defines a framework for allowing web services** to express their **constraints** and **requirements** in relation to security, processing, or message content.

23. What is the goal of WS-policy? (NOV/DEC 2011)

WS-policy provides the **mechanism** needed to **enable web services** applications **to specify policies.**

24. Give the specifications of WS-policy framework. (NOV/DEC 2011)

The WS-policy framework is comprised of the following three specifications:

- WS-policy
- WS-policyAssertions
- WS-policyAttachments

25. Define WS-Addressing.

The WS-Addressing specification implements these addressing features by providing two types of SOAP headers

26. What is WS-security?

WS-Security, also known as web services security or WSS, is a flexible and extensible framework to SOAP to apply security to web services.

27. Why this WS-security needed?

The WS-security is used to implement

- Message level security measures
  - Protect message contents during transport and during processing by service intermediaries.
- Authentication and authorization control
  - Protect service provides from malicious requestors.

28. Give the specifications of WS-security frame-work.

The WS-security framework is comprised of the following specifications:

- WS-security
- XML-encryption
- XML-signature

29. Give the syntax of WS-security element.

```
<Envelope>
<header>
....
<wsse:security actor="...."mustUnderstand="...'>
.....
</wsse:security>
.....
</header>
<body>
</body>
</envelope>
```

30. What does XML-signature elements provide?

The XML-signature elements provides message integrity and authentication information about the originator of the message.

31. Give the basic structure of the XML signature.

```
<Signature>
    <SignedInfo>
        <CanonicalizationMethod />
        <SignatureMethod />
        <Reference>
            <Transforms>
            <DigestMethod>
            <DigestValue>
        </Reference>
        <Reference /> etc.
    </SignedInfo>
  <SignatureValue />
  <KeyInfo />
```

```
    <Object />
</Signature>
```

32. Define WS-Coordination framework(Nov/Dec 2014)

The WS-Coordination framework exposes an Activation Service which supports the creation of coordinators for specific protocols and their associated contexts.

The process of invoking an activation service is done asynchronously, and so the specification defines both the interface of the activation service itself, and that of the invoking service, so that the activation service can call back to deliver the results of the activation - namely a context that identifies the protocol type and coordinator location.

33. List the stages of WS-Coordination

- **Instantiation (or activation)** of a new coordinator for the specific coordination protocol, for a particular application instance.
- **Registration** of participants with the coordinator, such that they will receive that coordinator's protocol messages during (some part of) the application's lifetime.
- **Propagation** of contextual information between Web services that comprise the application. An entity to drive the coordination protocol through to completion.

34. Define WS-Coordination context

- A coordination identifier with guaranteed global uniqueness for an individual coordinator in the form of a URI.
- An address of a registration service endpoint where parties receiving a context can register participants into the protocol.
- A TTL value which indicates for how long the context should be considered valid.
- Extensible protocol-specific information particular to the actual coordination protocol supported by the coordinator.

35. Define Compact Policy Expression.

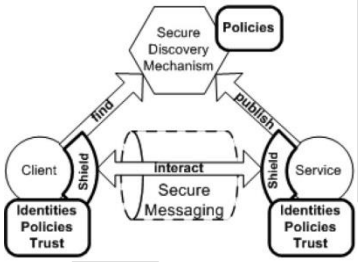
To express a policy in a more compact form while still using the XML Infoset, this specification defines three constructs: **an attribute** to decorate an assertion, **semantics** for recursively nested policy operators, and a **policy reference**/inclusion mechanism.

36. List the Web Services Security Requirements.

- Authorization
- Data Integrity and Data Confidentiality Integrity of Transactions and Communications Non-Repudiation
- End-to-End Integrity and Confidentiality of Messages Audit Trails
- Distributed Enforcement of Security Policies

37. Define Secure Messaging.

Secure Messaging ensures privacy, confidentiality and integrity of interactions. Digital signatures techniques can be used to help ensure non-repudiation.



UNIT V SERVICE ORIENTED ANALYSIS AND DESIGN

SOA delivery strategies – Service oriented analysis – Service Modelling – Service oriented design – Standards and composition guidelines -- Service design – Business process design – Case Study

PART - A

1. What is service oriented analysis? (MAY/JUN 2013)

The service-oriented analysis is the process of determining how business automation requirements can be represented through service-orientation.

2. What are the goals needed for performing a service-oriented analysis?

The overall goals of performing a service-oriented analysis are as follows:

- Define a preliminary set of service operation candidates.
- Group service operation candidates into logical contexts. These contexts represent service candidates.
- Define preliminary service boundaries so that they do not overlap with any existing or planned services.
- Identify encapsulated logic with reuse potential.
- Ensure that the context of encapsulated logic is appropriate for its intended use.
- Define any known preliminary composition models.

3. Give the step-by-step process in the service-oriented analysis.(Nov/Dec 2016)

Step 1: Define business automation requirements

Step 2: Identify existing automation systems

Step 3: Model candidate services.

4. What is service modeling?

Service modeling is a process of identifying candidate service operation and then grouping them into a logical context.

5. What is the use of service candidates? (Nov/Dec 2014)

The service candidates are used to distinguish a conceptualized service from an actual implemented service.

6. What is the key service-orientation principles applied to the service candidate? (Nov/Dec 2014)

- Reusability
- Autonomy
- Statelessness

- Discoverability

7. What is service oriented design?

Service-oriented design phase is a process that transforms previously modeled service candidates into physical service designs.

8. Give the overall goals for performing a service-oriented design.

The overall goals of performing a service-oriented design are as follows:

- Determine the core set of architectural extensions.
- Set the boundaries of the architecture.
- Identify required design standards.
- Define abstract service interface designs.
- Identify potential service compositions.
- Assess support for service-orientation principles.
- Explore support for characteristics of contemporary SOA

9. Write down the steps for composing SOA.

Step 1: choose service layers

Step 2: position core standards

Step 3: choose SOA extensions

10. What is Service-oriented analysis?

Service-oriented analysis establishes a formal analysis process completed jointly by business analysts and technology architects. Service modeling, a sub-process of service-oriented analysis produces conceptual service definitions called service candidates.

11. List out the component specification in Service modeling.

- Data Rules Services
- Configurable profile Variations

12. Expand WS-BPEL. What is WS-BPEL?

WS-BPEL stands for web services business process execution Language.

WS-BPEL is an XML based language enabling users to describe business process activities as web services and define how they can be connected to accomplish specific tasks.

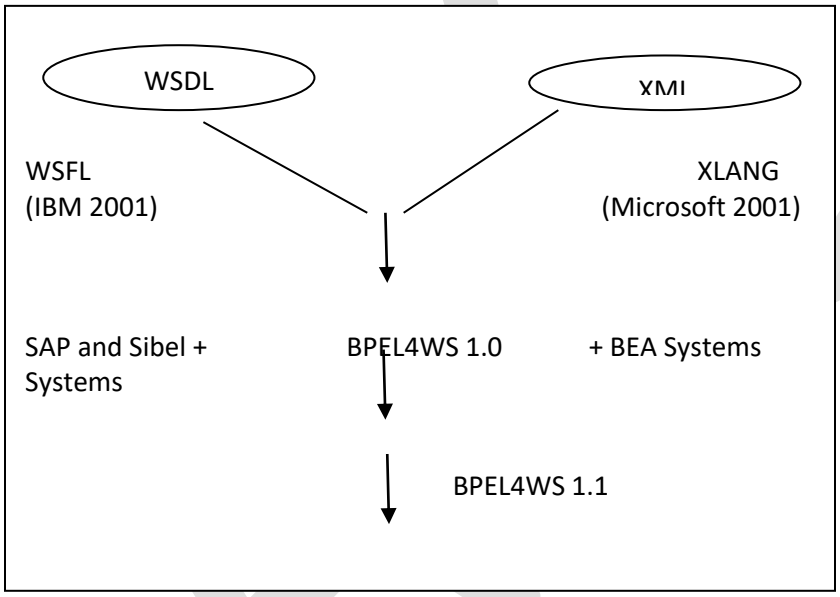
13. What is the process element?

An <process> element is the **root element** and must have a **name attribute for** assigning the **name value**. It is used to establish the process definition-related namespace.

14. What does the partner links element define?

The partner Links define the services that are orchestration by the process. It contains a set of <partner link> element each **represent** the **communication exchange between two partners**-the process service being one partner and another service being the other.

15. Draw the WS-BPEL family tree.



16. What are the attributes in the partnerlink element?

- myRole
- partnerRole

17. What is the use of myRole attribute?

- Used when the process service is invoked by a partner client service.
- Process service acts as the service provider.

18. What is the use of partnerRole attribute?

- Identifies the partner service that the process service will be invoking.
- Partner service acts as the service provider.

19. What is the use of partnerLinkType element?

The partnetLinkType elements are used to identify the WSDL portType elements reference by the partnerLink elements within the process definition.

20. What does the variables element hold?

Variables hold the data that constitute the state of a BPEL business process during runtime.

21. List out the attributes of the variables element.

- Message type
- Element
- Type

22. Define fault handlers.

Fault handlers are used to react to faults that occur while the business process activities are executing. The fault handles construct contain multiple catch element and (end with) a catchall child constructs.

23. Give the overview of WS-coordination.

WS- coordination is a framework for coordinating distributed activities

- **Coordinator**
  - Activation service for creating coordination instance
  - Registration service for registering participating applications
  - Additional protocol specific services
- **Set of coordination protocols**

24. What is the use of coordination context element?

The coordination context is used to carry information about active coordination to participants

- Information inside context is coordination protocol specific.
- Context format is not mandated by the standard.
- Typically passed in SOAP headers.

25. List out the stages to develop applications in WS-BPEL.

Denial - don't need it

Coercion - management says we have to use it

Elation - realization that it will solve all our enterprise application problems

Depression - realization that it will *not* solve all our enterprise application problems… yet

Enlightenment **-** understanding how - and when - to leverage the advantages of SOA, via BPEL (in this case, using your Java skills as a guide)

26. Define WS-Coordination framework(Nov/Dec 2014)

The WS-Coordination framework exposes an Activation Service which supports the creation of coordinators for specific protocols and their associated contexts.

The process of invoking an activation service is done asynchronously, and so the specification defines both the interface of the activation service itself, and that of the invoking service, so that the activation service can call back to deliver the results of the activation - namely a context that identifies the protocol type and coordinator location.

27. List out the SOA delivery lifecycle phases

- ✓ Basic phases of the SOA delivery lifecycle
- ✓ Service-oriented analysis
- ✓ Service-oriented design
- ✓ Service development
- ✓ Service testing
- ✓ Service deployment
- ✓ Service administration
- ✓ SOA delivery strategies

28. List out the SOA delivery three common strategies

- ✓ top-down
- ✓ bottom-up
- ✓ agile (or meet-in-the-middle)

29. What are the steps involved in top-down strategy of SOA Delivery Strategies

- ✓ Define relevant enterprise-wide ontology
- ✓ Align relevant business models (including entity models) with new or revised ontology
- ✓ Perform service-oriented analysis
- ✓ Perform service-oriented design
- ✓ Develop the required services
- ✓ Test the services and all service operations
- ✓ Deploy the services

30. What are the steps involved in bottom-up strategy of SOA Delivery Strategies

- ✓ Model required application services
- ✓ Design the required application services
- ✓ Develop the required application services
- ✓ Test the services
- ✓ Deploy the services

31. What are the steps involved in agile strategy of SOA Delivery Strategies

- ✓ Begin the top-down analysis, focusing first on key parts of the ontology and related business entities
- ✓ When the top-down analysis has sufficiently progressed, perform service-oriented analysis
- ✓ Perform service-oriented design
- ✓ Develop, test, and deploy the services
- ✓ As the top-down analysis continues to progress, revisit business services

32. List out goals of performing a service-oriented analysis

- ✓ Define a preliminary set of service operation candidates.
- ✓ Group service operation candidates into logical contexts. These contexts represent service candidates.
- ✓ Define preliminary service boundaries so that they do not overlap with any existing or planned services.
- ✓ Identify encapsulated logic with reuse potential.
- ✓ Ensure that the context of encapsulated logic is appropriate for its intended use.
- ✓ Define any known preliminary composition mode

33. Define Model candidate services

- ➢ A service-oriented analysis introduces the concept of service modeling a process by which service operation candidates are identified and then grouped into a logical context. These groups eventually take shape as service candidates that are then further assembled into a tentative composite model representing the combined logic of the planned service-oriented application

34. Compare Service oriented architecture vs Service oriented environment.

- ➢ Service-oriented architecture represents a technical view of a business automation solution based on service-orientation principles.

- Service-oriented environment to refer to the logical enterprise domain in which service-oriented principles are being applied.

35. List out the four types of service layers in service design.

- ✓ Entity-centric business services
- ✓ Application services
- ✓ Task-centric business services
- ✓ Process services