



TAGORE INSTITUTE OF ENGINEERING AND TECHNOLOGY

Deviyakurichi-636112, Attur (TK), Salem (DT). Website: www.tagoreiet.ac.in

Approved by AICTE, New Delhi and Affiliated to Anna University, Chennai

Accredited by NAAC

1

QUESTION BANK

Name of the Department : Computer Science Engineering
Subject Code & Name : IT8074 - SERVICE ORIENTED ARCHITECTURE
Year & Semester : IV & VII

UNIT I

INTRODUCTION TO XML

PART A

1. What is XML?

XML stands for Extensible Markup Language. It is a text-based markup language for describing the format for data exchanged between applications over the internet. XML permits document authors to create new markup languages for describing any type of data.

2. List the important characteristics of XML.

- ✓ **XML is extensible:** XML allows you to create your own self-descriptive tags, or language, that suits your application.
- ✓ **XML carries the data, does not present it:** XML allows you to store the data irrespective of how it will be presented.
- ✓ **XML is a public standard:** XML was developed by an organization called the World Wide Web Consortium (W3C) and is available as an open standard.

3. Mention the merits of XML.

- ✓ XML can work behind the scene to simplify the creation of HTML documents for large web sites.
- ✓ XML can be used to exchange the information between organizations and systems.
- ✓ XML can be used for offloading and reloading of databases.



- ✓ XML can be used to store and arrange the data, which can customize your data handling needs. 2
- ✓ XML can easily be merged with style sheets to create almost any desired output.
- ✓ Virtually, any type of data can be expressed as an XML document.

4. What are the XML Syntax Rules?

1. The XML declaration is case sensitive and must begin with "<?xml>" where "xml" is written in lower-case.
2. If document contains XML declaration, then it strictly needs to be the first statement of the XML document.
3. Attribute names in XML (unlike HTML) are case sensitive. That is, *HREF* and *href* are considered two different XML attributes.

5. What do you mean by XML declaration? Give an example.

The XML declaration is a *processing instruction* that identifies the document as being XML. All XML documents should begin with an XML declaration.

For example,

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
```

6. What is meant by a XML namespace?

An XML namespace is a collection of element and attribute names. XML namespace provide a means for document authors to unambiguously refer to elements with the same name (i.e., prevent collisions).

Example:

```
<subject>Geometry</subject> // data for student from  
school and  
<subject>Cardiology</subject> // data for student from medicine
```

Both the markup uses the element “subject” to markup data. Namespace can differentiate these two “subject” elements:

```
<highschool:subject>Geometry</highschool:subject>  
and <medicalschoo:subject>Cardiology</medicalschoo:subject>
```

Both **highschool** and **medicalschoo** are namespace prefixes.



7. How the namespace is declared?

A Namespace is declared using reserved attributes. Such an attribute name must either be **xmlns** or begin with **xmlns:** shown as below:

<element xmlns:name="URL">

Syntax:

- The Namespace starts with the keyword **xmlns**.
- The word **name** is the Namespace prefix.
- The **URL** is the Namespace identifier.

Example:

```
<root>
```

```
<highschool:subject
```

```
  xmlns:highschool="http://www.abcschool.edu/subjects"> Geometry
```

```
</highschool:subject>
```

```
<medicalschoo:subject
```

```
  xmlns:medicalshool="http://www.rmc.org/subjects"> Cardiology
```

```
</medicalschoo:subject>
```

```
</root>
```

8. What is meant by Document Type Definition?

Rules that define the legal elements and attributes for XML documents are called Document Type Definitions (DTD) or XML Schemas.

There are two different document type definitions that can be used with XML:

- DTD - The original Document Type Definition
- XML Schema - An XML-based alternative to DTD

9. What is the purpose of DTD?

The purpose of a DTD is to define the structure of an XML document. It defines the structure with a list of legal elements:



TAGORE INSTITUTE OF ENGINEERING AND TECHNOLOGY

Deviyakurichi-636112, Attur (TK), Salem (DT). Website: www.tagoreiet.ac.in

Approved by AICTE, New Delhi and Affiliated to Anna University, Chennai

Accredited by NAAC

4

Example DTD :

```
<!DOCTYPE  
PE note [  
<!ELEMENT note (to,from,heading,body)>  
<!ELEMENT to (#PCDATA)>  
<!ELEMENT from (#PCDATA)>  
<!ELEMENT heading (#PCDATA)>  
<!ELEMENT body (#PCDATA)>  
>  
<note>
```

```
<to>Tove</to>  
<from>Jani</from>  
<heading>Reminder</heading>  
<body>Don't forget to complete the work !</body>
```



</note>

The DTD above is interpreted like this:

- !DOCTYPE note defines that the root element of the document is note
- !ELEMENT note defines that the note element must contain four elements: "to, from, heading, body"
- !ELEMENT to defines the to element to be of type "#PCDATA"
- !ELEMENT from defines the from element to be of type "#PCDATA"
- !ELEMENT heading defines the heading element to be of type "#PCDATA"
- !ELEMENT body defines the body element to be of type "#PCDATA"

10. What is meant by Internal DTD?

A DTD is referred to as an internal DTD if elements are declared within the XML files. To refer it as internal DTD, *standalone* attribute in XML declaration must be set to **yes**. This means, the declaration works independent of external source.

Syntax

The syntax of internal DTD is as shown:

```
<!DOCTYPE root-element [element-declarations]>
```

where *root-element* is the name of root element and *element-declarations* is where you declare the elements.

Example

Following is a simple example of internal DTD:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE address [
  <!ELEMENT address (name,company,phone)>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT company (#PCDATA)>
  <!ELEMENT phone (#PCDATA)>]>
<address>
  <name>Tanmay Patil</name>
  <company>TutorialsPoint</company>
  <phone>(011) 123-4567</phone>
</address>
```



11. What is meant by XML validation?

XML validation is the process of checking a document written in XML (eXtensible Markup Language) against its specified DTD to confirm that it is both wellformed and also "valid". When a document fails to conform to a DTD or a schema, the validator displays an error message.

6

12. Give an example for External DTD.

In external DTD elements are declared outside the XML file. They are accessed by specifying the system attributes. To refer it as external DTD, *standalone* attribute in the XML declaration must be set as **no**. This means, declaration includes information from the external source.

Syntax:

Following is the syntax for external DTD:

```
<!DOCTYPE root-element SYSTEM "file-name">
```

where *file-name* is the file with *.dtd* extension.

Example:

The following example shows external DTD usage:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE address SYSTEM "address.dtd">
<address>
  <name>Tanmay Patil</name>
  <company>TutorialsPoint</company>
  <phone>(011) 123-4567</phone>
</address>
```

The content of the DTD file **address.dtd** are as shown:

```
<!ELEMENT address (name,company,phone)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT company (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
```

13. What is XML Schema?

XML Schema is commonly known as XML Schema Definition (XSD). It is used to describe



Accredited by NAAC

and validate the structure and the content of XML data. XML schema defines the elements, attributes and data types. Schema element supports Namespaces. It is similar to a database schema that describes the data in a database. 7

14. Give the syntax to declare XML schema.

Syntax:

The common syntax to declare a schema in our XML document as follows:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

Example: The following example shows how to use schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="contact">
<xs:complexType
  e>
  <xs:sequence>
    <xs:element name="name" type="xs:string" />
    <xs:element name="company" type="xs:string" />
    <xs:element name="phone" type="xs:int" />
  </xs:sequence>
</xs:complexType
  e>
</xs:el
ement
  >
</xs:sc
hema>
```

The basic idea behind XML Schemas is that they describe the legitimate format that an XML document can take.

15. What is CDATA?

The CDATA stands for Character Data. The character data will be parsed by the parser.

16. What is PCDATA?

It stands for Parsed Character Data (i.e. text). Any parsable character data should not contain the markup characters. The markup characters are < or > or &. If we want to use less than, greater than or ampersand characters then make use of <, > or &

17. What is meant by well-formed XML document?

An XML document with correct syntax is "Well Formed". A well-formed XML



document must follow the following XML syntax rules:

- XML documents must have a root element
- XML elements must have a closing tag
- XML tags are case sensitive
- XML elements must be properly nested
- XML attribute values must be

quote. Example well-formed

XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget to complete the work!</body>
</note>
```

18. What is meant by “Valid XML document”?

An XML document that contains the proper elements with the proper attributes in the proper sequence according to the DTD specification is called a “Valid” XML document. An XML document cannot be valid unless it is well-formed.

PART B

1. List and explain the XML syntax rules in detail. **(R)**
2. Explain how a XML document can be displayed on a browser. **(U)**
3. Explain the role of XML name spaces with examples. **(U)**
4. What are the features of elements and attributes in XML. **(Ap)**
5. Create a DTD for a catalog of four stroke motorbikes, where each motor bike has the following child elements – make, model, year, color, engine, chasis number and accessories. The engine element has the child elements engine number, number of cylinders, type of fuel. The accessories element has the attributes like disc brake, auto-start and radio, each of which is required and has the possible values yes and no. Entities must be declared for the names of the popular motorbike makes. **(C)**
6. What is DTD? Explain its types. **(U)**
7. What is XML schema? Give advantages and disadvantages of it. **(Ap)**
8. Explain the various data types used in XML Schema with suitable example. **(U)**
9. Write short notes on X-

Files. **(U) COURSE**



OUTCOME

- Be familiar with XML Fundamentals.

UNIT II

PART A

1. What is meant by XML DOM?

The W3C Document Object Model (DOM) is a platform and language-neutral Application Programming Interface (API) that allows programs and scripts to dynamically access and update the content, structure, and style of a HTML, XHTML & XML document.

2. What is meant by DOM Parser?

An XML parser that creates the DOM tree structure is known as **DOM parser**.

3. What is an XML Parser?

XML parser is a software library or a package that provides interface for client applications to work with XML documents. It checks for proper format of the XML document and may also validate the XML documents. Modern day browsers have built-in XML parsers.

Following diagram shows how XML parser interacts with XML document:



The goal of a parser is to transform XML into a readable code.

4. What are the two types of XML parsers?

Two types of XML Parsers:

1. Non-Validating XML Parsers
2. Validating XML Parsers

5. Compare DOM and SAX in XML processing.

- 1) DOM parser loads whole xml document in memory while SAX only loads small part of XML file in memory.



2) DOM parser is faster than SAX because it access whole XML document in memory.

3) SAX parser in Java is better suitable for large XML file than DOM Parser because it doesn't require much memory.

4) DOM parser works on Document Object Model while SAX is an event based xml parser.

10

6. How is XML parsing done with SAX? (or) What is event-oriented parsing?

Parsing can be done using two ways:

- Using DOM Parser
- Using SAX Parser

SAX Parser: it's an event based parsing it contains default handler for handling the events whenever SAX parser pareses the xml document and it finds the Start tag“<” and end tag”>” it calls corresponding handler method.

7. Define XSL.

The **Extensible Style sheet Language (XSL)** is an XML vocabulary that decides how an XML document data should look on the web browser.

XSL is group of three technoliges: i) **XSL Transformations (XSLT)**

ii) **XML Path Language (XPath)** iii) **XSL Formatting Objects (XSL-FO)**

8. Explain two types of XSL information.

An XSL document normally contains **two types of information**:

✓ **Template data** - Which is text that is copied to the output XML text document with little or no change? i.e. Everything in the body of the document that is not XSL markup is *template data*.

✓ **XSL markup** - which controls the transformation process always **start with “xsl:”**.

9. What are all the XSL components?

SL is constituted of three main components:

i) **XSL Transformations (XSLT)**: is technology for transforming the structure of the XML document data into another structure.



ii) **XML Path Language (XPath):** is a string-based language of expressions which defines the syntax and semantics for efficiently locating elements and attributes in XML documents.

11

iii) **XSL Formatting Objects (XSL-FO):** It is a separate XML vocabulary for defining document style properties of an XML document .(print-oriented)

10. Name any four XSL elements and Mentions its use.

- The **<xsl:variable>** XSL element creates a named variable.
- The **<xsl:value-of>** XSL element processes a value.
- The **<xsl:if>** XSL element invokes the enclosed code if the condition specified by the test attribute is true.
- The **<xsl:sort>** XSL element sorts a list of elements, such as within **<xsl:for-each>**.
- The **<xsl:template>** XSL element encapsulates a segment of XSL code, similar to a method, procedure, or function in other languages.

11. What is XSLT?

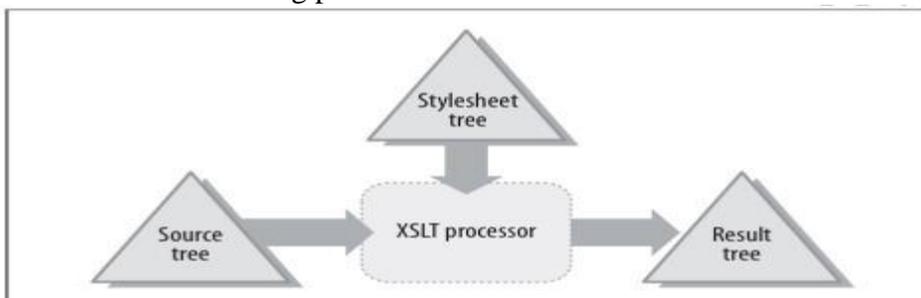
XSLT stands for XSL Transformations. XSLT is a language for transforming XML documents into XHTML documents or to other XML documents. XSLT uses XPath to navigate in XML documents. With XSLT user can add/remove elements and attributes to or from the output file.

12. Define XSL Formatting.

XSL-FO (XSL Formatting Objects) is a markup language for XML document **formatting** that is most often used to generate PDF files. **XSL-FO** is part of **XSL** (Extensible Stylesheet Language), a set of W3C technologies designed for the transformation and **formatting** of XML data. The other parts of **XSL** are **XSLT** and XPath.

13. How does XSLT works?

In the transformation process, XSLT uses XPath to define parts of the source document that should match one or more predefined templates. When a match is found, XSLT will transform the matching part of the source document into the result document.



14. What are all the events of Sax?

The SAX events include:-



- ✓ XML Text nodes
- ✓ XML Element nodes
- ✓ XML Processing Instructions (PI)
- ✓ XML Comments

15. Define XPATH.

XPath is a syntax for specifying a collection of elements or other information contained within an XML document. XPath is a language for finding information in an XML document. We can reach to any node of XML document using XPATH.

PART - B

1. Explains about DOM. **(U)**
2. Explains in detail about XSL. **(U)**
3. Write XML document for checking well formedness of XML document using SAX API. **(C)**
4. Give an XSLT document and a source XML document and explain the XSLT transformation process that produces a single result XML document. **(Ap)**
5. Explains in detail about XML parsers. **(U)**
6. Write short notes on XSL formatting. **(U)**
7. How to modeling databases in XML? Explain it with suitable examples. **(Ap)**
8. Distinguishes between DOM and SAX Parsers. **(An)**
9. Write XML document for checking well formedness of XML document using DOM API. **(C)**

COURSE OUTCOME

- Build applications based on XML.

UNIT III

PART A

1. What are the roots of SOA? (May 2012)

The following are building blocks of SOA

Service reusability, service abstraction, and service composability.



2. Define Service orientation. (Dec 2012)

Service-orientation is a design paradigm to build computer software in the form of services. Like other design paradigms (e.g. object-orientation), service-orientation provides a governing approach to automate business logic as distributed systems. What distinguishes service-orientation is its set of design principles to ensure the manner in which it carries out the separation of concerns in the software.

13

3. What are the characteristics of SOA? (Dec 2011)(May 2014)

- ✓ Loosely coupled: minimizes dependencies between services.
- ✓ Contractual: adhere to agreement on service descriptions.
- ✓ Autonomous: control the business logic they encapsulate.
- ✓ Abstract: hide the business logic from the service consumers.
- ✓ Reusable: divide business logic into reusable services.
- ✓ Composable: facilitate the assembly of composite services.
- ✓ Stateless: minimize retained information specific to an activity.
- ✓ Discoverable: self-described so that they can be found and assessed.

4. List the potential benefits of adhering SOA. (May 2012)

- ✓ Based on open standards.
- ✓ Supports vendor diversity.
- ✓ Fosters intrinsic interoperability.
- ✓ Promotes discovery.
- ✓ Promotes federation.
- ✓ Fosters inherent reusability.
- ✓ Emphasizes extensibility
- ✓ Promotes organizational agility.
- ✓ Supports incremental implementation.

5. How SOA differs from client server architecture?(Dec 2012)

The approach and the technology available are mature and stable. Systems of this type can be easily implemented and optimized for special application scenarios. On the other hand, once a client-server system is established it is difficult to change the functionality of the system. Therefore its flexibility is limited and best used, if the business process is expected to be stable and changes are seldom

6. State the difference between SOA and distributed Computing. (Dec 2011)

It has the ability to legacy language support to object oriented language. Message Interface is high in Distributed computing.

Middleware protocol is used in distributed computing Security is high



7. Give the 8 design principles to build SOA. (Nov 2011)

- ✓ increased consistency in how functionality and data is represented
- ✓ reduced dependencies between units of solution logic
- ✓ reduced awareness of underlying solution logic design and implementation details
- ✓ increased opportunities to use a piece of solution logic for multiple purposes
- ✓ increased opportunities to combine units of solution logic into different configurations
- ✓ increased behavioral predictability

8. What is meant by 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, etc.

9. What are the principles of service orientation? (Nov 2014)

- ✓ **Service Loose Coupling** – Services maintain a relationship that minimizes dependencies and only requires that they maintain an awareness of each other.
- ✓ **Service Abstraction** – Beyond descriptions in the service contract, services hide logic from the outside world.
- ✓ **Service Reusability** – Logic is divided into services with the intention of promoting reuse.
- ✓ **Service Autonomy** – Services have control over the logic they encapsulate.
- ✓ **Service Statelessness** - Services minimize resource consumption by deferring the management of state information when necessary
- ✓ **Service Discoverability** – Services are supplemented with communicative meta data by which they can be effectively discovered and interpreted.
- ✓ **Service Composability** – Services are effective composition participants, regardless of the size and complexity of the composition.

10. What is the need for design standards in building SOA?

Design standards can be healthy for an enterprise in that they "pre-solve" problems by making several decisions for architects and developers ahead of time, thereby increasing the consistency and compatibility of solution designs.

11. State Top-down requirement approach in SOA.

A preferred strategy to delivering services is to first conceptualize a service inventory by defining a blueprint of all planned services, their relationships, boundaries, and



12. Define counter agile delivery process. (may 2010)

It takes more effort to design and build service-oriented solution logic than it does to build a corresponding amount of logic that is not service-oriented, the process of delivering services in support of SOA can actually be counter-agile.

13. What are the additional principles for SOA?

Service Optimization – All else equal, high-quality services are generally preferable to low- quality ones.

Service Relevance – Functionality is presented at a granularity recognized by the user as a meaningful service.

Service Encapsulation – Many services are consolidated for use under the SOA. Often such services were not planned to be under SOA.

Service Location Transparency – This refers to the ability of a service consumer to invoke a service regardless of its actual location in the network.

14. Define service statelessness.

Services minimize resource consumption by deferring the management of state information when necessary

15. What is meant by service composability?

Services are effective composition participants, regardless of the size and complexity of the composition.

16. What is the meaning for service discoverability? (Nov 2011)

Services are supplemented with communicative meta data by which they can be effectively discovered and interpreted.

17. What are the other factors take into account when defining a SOA implementation?

- ✓ Efficient use of system resources
- ✓ Service maturity and performance
- ✓ EAI (Enterprise Application Integration)

18. What is Service Oriented Architecture (May 2013, Nov 2013)

Service oriented architecture is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity.

19. What is Architecture (Nov 2013)



TAGORE INSTITUTE OF ENGINEERING AND TECHNOLOGY

Deviyakurichi-636112, Attur (TK), Salem (DT). Website: www.tagoreiet.ac.in

Approved by AICTE, New Delhi and Affiliated to Anna University, Chennai

Accredited by NAAC

Architecture refers a systematic arrangement of computerized automation technological Solutions.

16

20. List out the logical components of automation logic (May 2013)

Web Service
operations Web
Services
Activities

21. Define Application architecture. (May 2014, Nov 2014)

Application Architecture is to an application development team what a blueprint is to a team of construction workers. It provides more detail such as common data models, communication flow diagrams, application-wide security requirements, and aspects of infrastructure.

a team of cons

PART B

1. Explain the basic building blocks of Service Oriented Architecture. (Nov 2011) **(U)**
2. List the characteristic feature of SOA and explain each in detail. (May 2012) **(R)**
3. Compare SOA with client server architecture and summarize the difference. (May 2013, Dec 2012) (May 2014) **(An)**
4. State and explain distributed Computing with SOA along with their difference. **(An)**
5. Summarizes the benefits of SOA. **(U)**
6. Explain briefly about web services as component wrappers. (Nov 2013) **(U)**
7. Explain the principles of Service orientation in detail. (Nov 2011, May 2013) **(U)**
8. Describe how SOA can be compared to distributed internet architectures? (Nov 2014) **(An)**
9. Illustrates the service layers of SOA in detail. (or) Discuss about different service layer in detail. (May 2013). **(An)**

COURSE OUTCOME

- Be exposed the key principles behind SOA.

UNIT IV

PART A

1. What is meant by Web Services? (May 2010, Nov 2013)



Accredited by NAAC

Web services constitute a distributed computer architecture made up of many different computers trying to communicate over the network to form one system. 17 They consist of a set of standards that allow developers to implement distributed applications

2. Define UDDI. (Dec 2012)

UDDI is a protocol for describing available Web services components. This standard allows businesses to register with an Internet directory that will help them advertise their services

3. What is meant by SOAP? (May 2012)

SOAP is a protocol for initiating conversations with a **UDDI Service**. **SOAP** makes object access simple by allowing applications to invoke object methods or functions, residing on remote servers.

4. What is WSDL? (Nov 2011)

WSDL is the proposed standard for how a Web service is described, is an XML-based service IDL (Interface Definition Language) that defines the service interface and its implementation characteristics.

5. How web services are implemented? (Dec 2012)

- A service provider creates a **Web service**
- The service provider uses **WSDL** to describe the service to a **UDDI registry**
- The service provider registers the service in a **UDDI registry and/or ebXML registry/repository**.
- Another service or consumer locates and requests the registered service by querying **UDDI and/or ebXML registries**.
- The requesting service or user writes an application to bind the registered service using **SOAP** in the case of **UDDI and/or ebXML**
- Data and messages are exchanged as **XML** over **HTTP**

6. How the Messages are processed using SOAP?

It should use the SOAP Envelope and SOAP Encoding namespaces and have the following form:

- An XML Declaration (optional), followed by
- A SOAP Envelope (the root element) which is made up of:
 - A SOAP Header (optional)
 - A SOAP Body

7. State the SOAP Processing Model. (May 2012)

SOAP provides a distributed processing model that assumes a SOAP message



originates at an initial SOAP sender and is sent to an ultimate SOAP receiver via zero or more SOAP intermediaries.

8. Define MEP. (Nov 2011)

A Message Exchange Pattern (MEP) is a template that establishes a pattern for the exchange of messages between SOAP nodes. MEPs are a type of feature, and unless otherwise stated, references in this specification to the term "feature" apply also to MEPs.

9. What is the specification needed in MEP?

An MEP specification MUST also include:

1. Any requirements to generate additional messages (such as responses to requests in a request/response MEP).
2. Rules for the delivery or other disposition of SOAP faults generated during the operation of the MEP.

The SOAP specification mentions a concept called "message exchange patterns"

10. Give the processing steps to coordinate web service.

The WS-Coordination specification defines an Activation protocol with the following two messages:

- The CreateCoordinationContext message targets an ActivationService, and provides an optional CurrentContext parameter that is a CoordinationContext for an existing activity. This parameter causes the activity to be imported instead of created. Import is described later.
- CreateCoordinationContextResponse message returns the newly created CoordinationContext.

11. Define Atomic Transaction in Web services.

Atomic transactions greatly simplify application programming by shielding the application from system-generated exceptions, especially when they do not recur on retry.

12. What is 2PC Protocol?

The atomic transaction 2PC protocol coordinates registered services to reach a commit or abort decision, and informs all services of the final result. The decision is the same for all the services in the transaction.

13. State three atomic protocols to complete transaction in web services.



- Completion
- Volatile2PC
- Durable2PC

14. What is the functionality of WS-Business Activity? (May 2014)

WS-BusinessActivity defines two coordination protocols that differ only in the knowledge of when a unit of work has been completed.

- BusinessAgreementWithParticipantCompletion:** The participant knows when the coordinator will ask no more work of it.
- BusinessAgreementWithCoordinatorCompletion:** The coordinator has to tell the participant that no more work will be asked of it.

15. How Business Activities Differ From Atomic Transactions?

- Type of exceptions
- Duration
- Isolation
- Durability
- Agreement Scope
- Diverse operational characteristics
- Combined services
- Fewer constraints in coordinator behavior
- Expires time

16. Define Orchestration.(Dec 2012)

Orchestration is fundamentally comprised of the co-existent application of Process Abstraction, State Repository, Process Centralization, and Compensating Service Transaction, and can be further extended via Atomic Service Transaction, Rules Centralization, and Data Model Transformation.

17. Give the strength of work flow system in Orchestration.

The strength of workflow systems is that they abstract the essence of the application flows into a set of processes that can be easily modified to account for differences in business processes.

18. Define Choreography. (Nov 2011, May 2014)

Generate a behavioral interface that conforms to a BPEL definition that describes the sequence and conditions in which one of the participants in a choreography sends and receives messages.

19. State the relationship between choreography types.



TAGORE INSTITUTE OF ENGINEERING AND TECHNOLOGY

Deviyakurichi-636112, Attur (TK), Salem (DT). Website: www.tagoreiet.ac.in

Approved by AICTE, New Delhi and Affiliated to Anna University, Chennai

Accredited by NAAC

	Abstract	Portable	Concrete
Types of Messages	Identified	Identified	Identified 20
Message Structure	Not Defined	Defined	Defined
Conditions	Identified	Identified	Identified
Condition evaluation rules	Not defined	Defined as far as possible	Defined as far as possible
Technology used	Not defined	Defined	Defined
Message Endpoint Data	Not defined	Not Defined	Defined

20. Give the features of Orchestrian servive Layer. (May 2010)

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

21. Explain the SOAP Message Format (May 2013)

```

<?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>
  
```

22. State 2 mechanism for WSDL basics.

Mechanism	Object	Meaning	Visibility of Schema Components
wsdl:import	WSDL 2.0 Namespace	Declare that WSDL 2.0 components refer to WSDL 2.0 components from a DIFFERENT targetNamespace.	XML Schema Components in the imported Description component are NOT visible to the containing description.



Accredited by NAAC

wSDL:include	WSDL 2.0 Document	Merge Interface, Binding and Service components from another WSDL 2.0 document that has the SAME targetNamespace.	XML Schema components in the included Description component's {element declarations} and {type definitions} properties are visible to the containing description.
--------------	-------------------	---	---

23. What are the basics of SOAP? (Nov 2011)

The listener process, which for simplicity is typically written in the same language as the service method, decodes the incoming SOAP request and transforms it into an invocation of the method.

24. Define Envelope element in SOAP message? (Nov 2014)

The SOAP envelope indicates the start and the end of the message so that the receiver knows when an entire message has been received. The SOAP envelope solves the problem of knowing when you are done receiving a message and are ready to process it. The SOAP envelope is therefore basically a packaging mechanism.

25. Differentiate orchestration from activities. (May 2013)

- Orchestration is a necessity of building a SOA, intra or inter organization
- It's the layer that creates business solution from the vast array of services and information flows found in new and existing systems.

- WS-BPEL breaks down workflow logic into a series of predefined primitive activities.
- Activities are Receive, invoke, reply, throw

26. List out Web service platform elements. (Nov 2013)

- SOAP
- WSDL
- UDDI

27. List out the characteristics of ACID transactions? (Nov 2014)

Atomic – Either all of the changes within the scope of the transaction succeed, or none of them succeed. This characteristic introduces the need for the rollback, feature that is responsible for restoring any changes completed as part of a failed transaction to their original state.

Consistent- None of the data changes made as a result of the transaction can violate the validity of any associated data models. Any violations result in a



rollback of the transaction.

Isolated-If multiple transactions occur concurrently, they may not interfere with each other. Each transaction must be guaranteed an isolated execution environment. 22

Durable- Upon the completion of a successful transaction, changes made as a result of the transaction can survive subsequent failures.

28. What is meant by WS-Coordination?

The procedure to establish coordination between the set of participants of an activity.

29. Give 2 stages of WS-Coordination. (May 2010)

In the first stage, the activity is configured; one may think of this as the participants being wired together. The second stage is that in which the participants exchange messages over the wires according to the protocols that have been established.

30. What is Coordination Context?

A Coordination Context includes the following information:-

- An activity identifier.
- The type of the coordination. (e.g., atomic transaction)
- The Endpoint Reference to the Registration service.
- An optional expiration time for the activity.
- Extensibility elements to allow other information to be communicated.

PART B

1. Explain the standards of Web services and its support to various technologies. (Nov 2011)

(U)

2. How the message transfer done in SOAP protocol? Give an Example. (U)
3. Explain in detail about Request and Response services in MEP. (May 2012) (U)
4. With an example give the salient features of WS-Transaction. (Ap)
5. Explain the role of Orchestration and illustrate with architecture. (U)
6. Write short notes on (May 2013) (U)
 - i. Service Description
 - ii. Atomic Transaction
 - iii. Choreography
7. Explain the role of Application service Layer in building Web services.(Dec 2012)



TAGORE INSTITUTE OF ENGINEERING AND TECHNOLOGY

Deviyakurichi-636112, Attur (TK), Salem (DT). Website: www.tagoreiet.ac.in

Approved by AICTE, New Delhi and Affiliated to Anna University, Chennai

Accredited by NAAC

(U)

8. How Business service layer utilize web services to communicate the application? **23**

Explain with an example. **(Ap)**

9. Describe Orchestration service Layer and its functionality in detail. (May 2010) **(R)**

10. Explain briefly about processing SOAP messages.(Nov 2013) **(U)**

11. Explain briefly about Technical Requirements for Orchestration and Choreography. (Nov 2013) **(U)**

12. Describe the process of messaging with SOAP and Atomic Transaction. May 2014) **(R)**

13. Discuss about Business service layer and Orchestration service layer in detail. (May 2014) **(U)**

14. Write short notes on: **(U)**

(i) Messaging with SOAP.(Nov 2014)

(ii) Business activities in web services. (Nov 2014)

15. Briefly explain about: **(U)**

(i) Service layer abstraction (Nov 2014)

(ii) Application Service Layer (Nov 2014)

16. Explain the features of SOAP in analyzing web services. (May 2012) **(U)**

COURSE OUTCOME

- Develop web services using technology elements.

UNIT V

PART A

1. **What is meant by WS-BPEL? (May 2010)**

Web Services Business Process Execution Language (WS-BPEL) is an XML based programming language to describe high level business processes. A 'business process' is a term used to describe the interaction between two businesses or two elements in some business.

2. **List the three basic components for WS-BPEL. (May 2012)**



Accredited by NAAC

- Programming logic
- Data types
- Input/Output (I/O)

3. Define Control flow. (Dec 2012)

The control flow is a nice hybrid model half-way between block structured and state transition control flow definitions. The model uses "links" to establish the dependencies between block definitions: "The links are defined inside the flow and are used to connect a source activity to a target activity".

4. Define Data Flow. (May 2010)

The information is passed between the different activities in an implicit way through the sharing of globally visible data containers. Containers may exchange specific data elements via the use of "Assign" statements.

5. What are the services provided by WS-Coordination? (Nov 2011)

The WS-Coordination specification defines a framework that allows different coordination protocols to be plugged in to coordinate work between clients, services, and participants.

6. What is the role of WS-Coordination with its requirement?

- 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 the Web services that comprise the application
- An entity to drive the coordination protocol** through to completion

7. What is the role of Registration process in WS-Coordination?

Once a coordinator has been instantiated and a corresponding context created by the activation service, a Registration Service is created and exposed. This service allows

participants to register to receive protocol messages associated with a particular coordinator.

8. What is the goal of WS-Policy?

The goal of WS-Policy is to provide the mechanisms needed to enable



Approved by AICTE, New Delhi and Affiliated to Anna University, Chennai

Accredited by NAAC

Web services applications to specify policy information. Specifically, this specification defines the following:

25

- An XML Infoset called a *policy expression* that contains domain-specific, Web Service policy information.
- A core set of constructs to indicate how choices and/or combinations of domain- specific policy assertions apply in a Web services environment.

9. What is meant by Policy Expression?

A *policy expression* is an XML Infoset representation of a policy, either in a normal form or in an equivalent compact form.

10. Define policy Assertion.

A policy assertion identifies a behavior that is a requirement (or capability) of a policy subject. Assertions indicate domain-specific (e.g., security, transactions) semantics and are expected to be defined in separate, domain-specific specifications.

11. What do you know about policy Alternative?

A policy alternative is a logical construct which represents a potentially empty collection of policy assertions. An alternative with zero assertions indicates no behaviors. An alternative with one or more assertions indicates behaviors implied by those, and only those assertions.

12. How the security will work in WS-Security? (nov 2011)

WS-Security provides means to secure your services above and beyond transport level protocols such as HTTPS. It allows you to:

- Pass authentication tokens between services
- Encrypt messages or parts of messages
- Sign messages
- Timestamp messages

13. Distinguish between the identifier and expires elements. (May 2014)

Ws-coordination uses the identifier element to associate a unique ID value with the current activity.

The Expires element sets an expiry date that establishes the extent of the activity possible life span

14. Differentiate getVariableProperty and getVariableData function?(Nov 2014)

getVariableProperty	getVariableData
----------------------------	------------------------



TAGORE INSTITUTE OF ENGINEERING AND TECHNOLOGY

Deviyakurichi-636112, Attur (TK), Salem (DT). Website: www.tagoreiet.ac.in

Approved by AICTE, New Delhi and Affiliated to Anna University, Chennai

Accredited by NAAC

The <code>getVariableProperty</code> 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.	The <code>getVariableData</code> function has a mandatory variable name parameter and two optional arguments that can be used to specify a part of the variable data.
Syntax: <code>getVariableProperty(variable name, property name)</code>	Syntax: <code>getVariableData(variable name, part name, location path)</code>
Example: <code>getVariableProperty("TicketApproval","class")</code>	Example: <code>getVariableData("input","payload","tns:TimesheetType/Hours/...")</code>

15. Write any four XML signature elements. (Nov 2014)

- CanonicalizationMethod
- DigestMethod
- DigestValue
- KeyInfo
- Signature
- SignatureMethod
- SignatureValue
- SignedInfo
- Reference

16. What is DigestMethod and DigestValue? (May 2014)

DigestMethod – Identifies the algorithm used to create the signature.

DigestValue – Contains a value that represents the document being signed. This value is generated by applying the DigestMethod algorithm to the XML document.

PART B

1. Explain in detail about basic architecture of WS-BPEL. (May 2012, May 2013, Nov 2013) **(U)**
2. Describe with an example to illustrate WS-Coordination. (Dec 2012) **(R)**
3. How the WS-Security will lead to SOA in accessing the web services? (Nov 2011) **(Ap)**
4. List the type of WS-Security with their salient features in detail. (May 2012, May 2013) **(R)**



TAGORE INSTITUTE OF ENGINEERING AND TECHNOLOGY

Deviyakurichi-636112, Attur (TK), Salem (DT). Website: www.tagoreiet.ac.in

Approved by AICTE, New Delhi and Affiliated to Anna University, Chennai

Accredited by NAAC

5. What are the different types of WS-Policy available? Explain them in detail. **(U)**
6. Discuss WS-BPEL language basics in detail.(16) (May 2014) **(U)**
7. Explain WS-security language basics in details.(16) (May 2014) **(U)**
8. Write short notes on WS-Coordination overview.(16) (Nov 2014) **(U)**
9. Briefly explain about WS-Policy language basics.(16) (Nov 2014) **(U)**
10. What are the services provided in WS- Coordination? Explain with an Example. (Dec 2012) **(U)**

27

COURSE OUTCOME

- Build SOA-based applications for intra-enterprise and inter-enterprise applications.