

UNIT- IV

SOA & Business Process Management – P1

A Business process is a real world activity to achieve their goals. It consists of set of logically related tasks.

Ex: Deliver Products (or) Services

When performed in a appropriate sequence and correct business rules produces a business outcome.

Ex: Order-to-cash

Ranges from Short & Long-Lived

Short -> Minutes or hours

Ex: Website Freelancing

Long-> Weeks, months, years

Ex: Market shares, Gold Investment, Mutual Funds

BPM addresses the how organizations identify, model, develop, deploy & manage their business process.

The main goals & benefits of BPM including:-

- Minimize the misalignment (or) Inconsistency between business requirements
- Increases employee productivity & reduce operational costs i.e, ongoing business (Day to Day) by automating & streamlining (minimizing workflows) Business process.

Ex: **OC**: Employee related costs (salaries & wages), Rent & utilities.

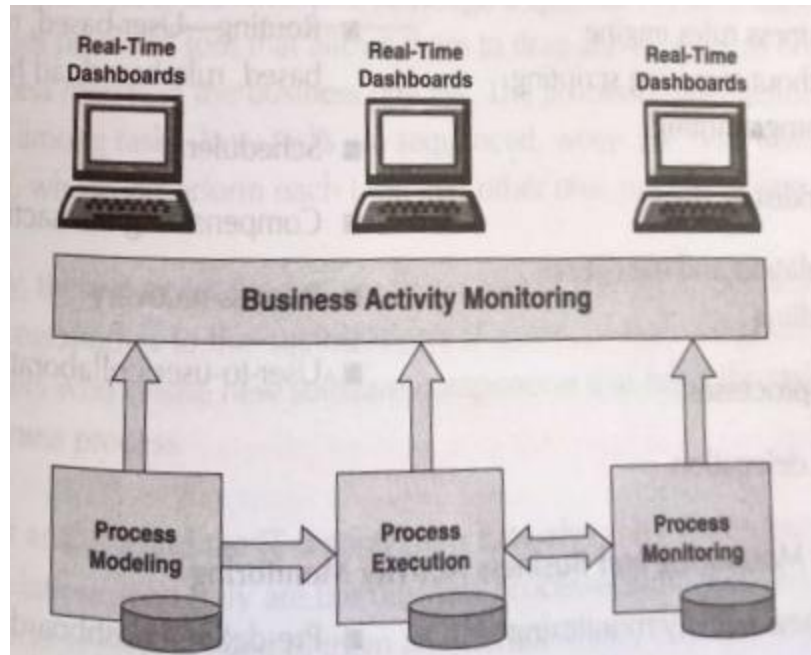
Business Process Management Systems

Associated with Defining, Managing & Executing business processes as a corporate asset,

BPM provide the technology that implements one (or) more core BPM Functions.

- ✓ **Workflow**: A service of tasks, processes (or) steps that need to be completed in a specific order to achieve a goal.

- ✓ **Process Automation:** Minimize workflows i.e, execute routine, manual tasks & processes without human interventions.
- ✓ **Enterprise Application Integration:** Connecting & coordinating different s/w applications within an organization to function as unified.
- ✓ **Business 2 Business:** Transactions (or) Interactions occur between 2 businesses rather than individuals.
Ex: Involves exchange of goods, services.
- ✓ **Service Composition:** Combining individual services (or) s/w components to create a composite service.
- ✓ **Orchestration:** Coordination & management of multiple tasks (or) activities to get a outcome.



BASIC COMPONENTS OF A BPMS

BPMS provides a process modeling tool,

Which allows processes (or) operations to be defined as “Graph”.

Where the node of the graph represents the tasks to be performed & the arcs of the graph represent control flow (or) data flow.

Ex: **ARCS**- Task Pre-conditions, Routine logic, Time delays, Deadlines.

Process Modeling

The goal of BPM is to capture business requirements at initial design stage and then make them available to the rest of the development process.

Here, Business Analysts plays a crucial (or) central role

In BPM Life Cycle

They satisfies their needs using a process modeling tool

These tools typically provide business analysts with a GUI interface (Although sometimes a spreadsheet –style interface provide them too) allows to drag & drop icons onto a graphical process model of the business process.

The process model defines relationships among tasks, how tasks are sequenced. When & how tasks are enabled, who can perform each task.

Relationships i.e,

Finish-to-start: Task B cannot start until Task A is completed

Start-to-start: Task B cannot start until Task A starts
(2 runs in parallel)

Finish-to-finish: Task B cannot finish until Task A finishes (P)

Start-to-finish: Task B cannot finish until Task A starts.

Next, These Process Models are given to **Technical Specialists** who map the PM to the organizations IT assets.

(or)

They given to s/w developers (who create new s/w components) that fulfill the tasks defined by BP.

Process Executions

After BP has been modeled & mapped to new & existing IT assets it is ready to be deployed.

BPM includes process execution engines that import the process models & then execute & manage as many business process (Necessary for supporting the organizations operational requirements)

The process execution engine is responsible for executing the process models & enforces the business roles associated with the process such as:-

- i) Invoking tasks (or) Executing tasks in the correct order.
- ii) Assigning & Routine tasks to authorized users.
- iii) Tracking the current state of the process.

Process Monitoring

Allow business users & IT administrators to monitor & control a business process.

This includes:-

- Seeing a summary of all executing processes

- Seeing a summary of all completed processes, including historical trends
- Suspending & Resuming Processes
- Altering the priority of processes
- Re-assigning Processes

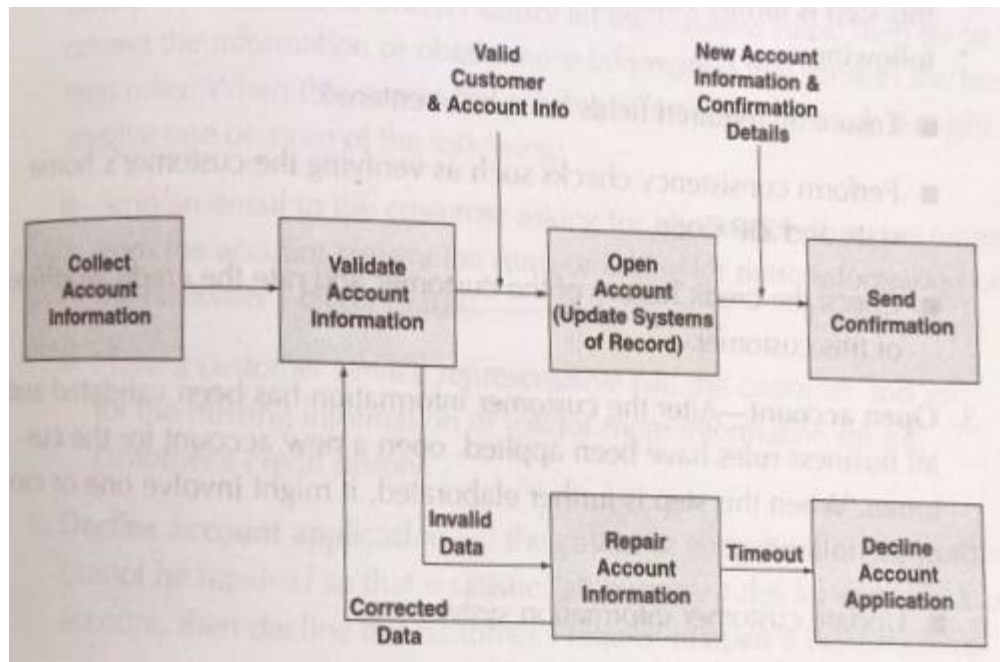
Example Business Process

Opening a Customer A/C

Activities to be performed are:-

Link between the activities (that determine the order in which activities can be performed & the data i.e, passed b/w the activities)

For, business rules it enables transistions (movements i.e, state (or) phase (1 to another)) b/w activities.



SIMPLE BUSINESS PROCESS

The steps in the business process are:-

1. Collect A/C Info:-

Gather all necessary customer & account info to open the a/c.

When this step is further elaborated it might involve one or more of the following:-

- Customer enter data via Website.
- Customer provides data-> Customer Service Representative over the phone.

- Customer provides data on a written form, & data enters it into a A/c system<- entry clerk
- Customer already has an a/c, customer info is extracted from Data Base.

2. Validate A/C info:-

After the info collected-> it needs to be validated
“business rules need to be checked”.

i.e,

- ✓ Ensure all required fields have been entered
- ✓ Perform consistency check (by verifying customer info home state & zipcode.
- ✓ Check the credit history & rate the worthiness of customer.

3. Open A/C:-

After validation, business rules have been applied,
Open a new a/c for the customer

- ✓ Update (Customer Info System), (Order Management System), (Billing System)

- ✓ Update Sale force Automative system & inform sales representative of new a/c.
- ✓ Inform business partners of New- Customer & the Customer Credentials, Such as “Credit Limit”.

4. Send Confirmation:-

After successful a/c created (or) opened, send confirmation to the customer along with details of new a/c.

- ✓ Send a letter to the customer thanking him/her for opening this a/c.
- ✓ Send a mail.

5. Repair a/c info:

If the customer & account info cannot be validated (or) doesn't satisfy all the business rules, then try to correct the information).

- ✓ Have a customer service- representative (for all the customer & ask missing info)

6. Decline A/C application:

If the customer & a/c info cannot be repaired (so that it satisfies all business rules for opening a new a/c) then decline the customer request to open a new a/c.

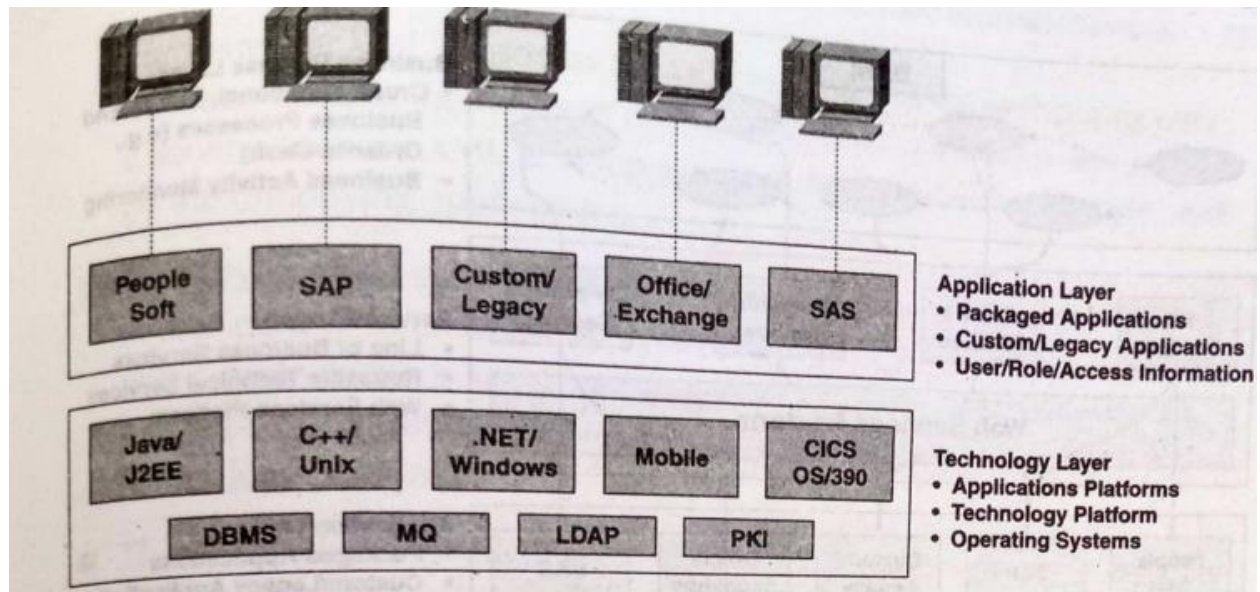
Combining BPM, SOA & Web Services

This section discuss about the benefits of using BPM, SOA & Web Services in combination.

The benefits include a more flexible and agile implementation of a BPM system and the ability to more easily create, manage, and maintain composite applications.

Benefits of BPM, SOA & WS

Most organizations have a huge applications & Technology Landscape (overview about all platforms, tools & solutions)



TYPICAL APPLICATION & TECHNOLOGY LANDSCAPE

Application Layer

Packaged applications-> Pre-Built (Design to address specific needs)

Custom/Legacy Applications->

Custom-Built -> From Scratch (Designed, Developed & meets the organization requirements)

LA-> (used outdated technologies)

User/role/access information-> Manages the user identifies, roles & controlling access to system, data & applications.

Technology Layer

Application Platforms-> Foundation of built, run & Deploy the application, offers a combination of development tools & infrastructure services)

Note: Used to create simple-> complex

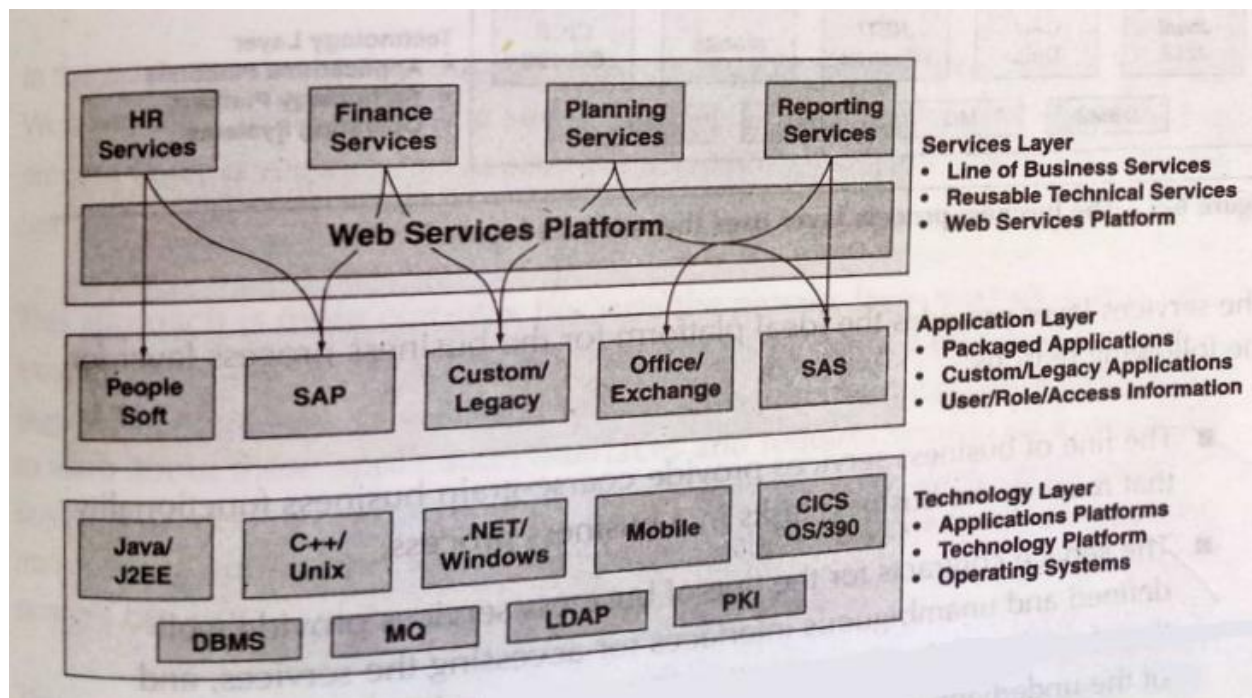
Ex: AWS

Technology Platforms-> H/w, S/w, development tools for to built the applications.

Operating Systems-> Intermediate b/w computer-hardware & user applications.

In the above figure, the applications are standalone in nature (independent) from (GUI to business logic to database) so sharing info is difficult due to difference in technology platform & data models.

So we need to overcome the communication gap between this applications for that moving to an SOA & web services introduces a service layer.



SERVICES LAYER BASED ON SOA & WS

Services Layer

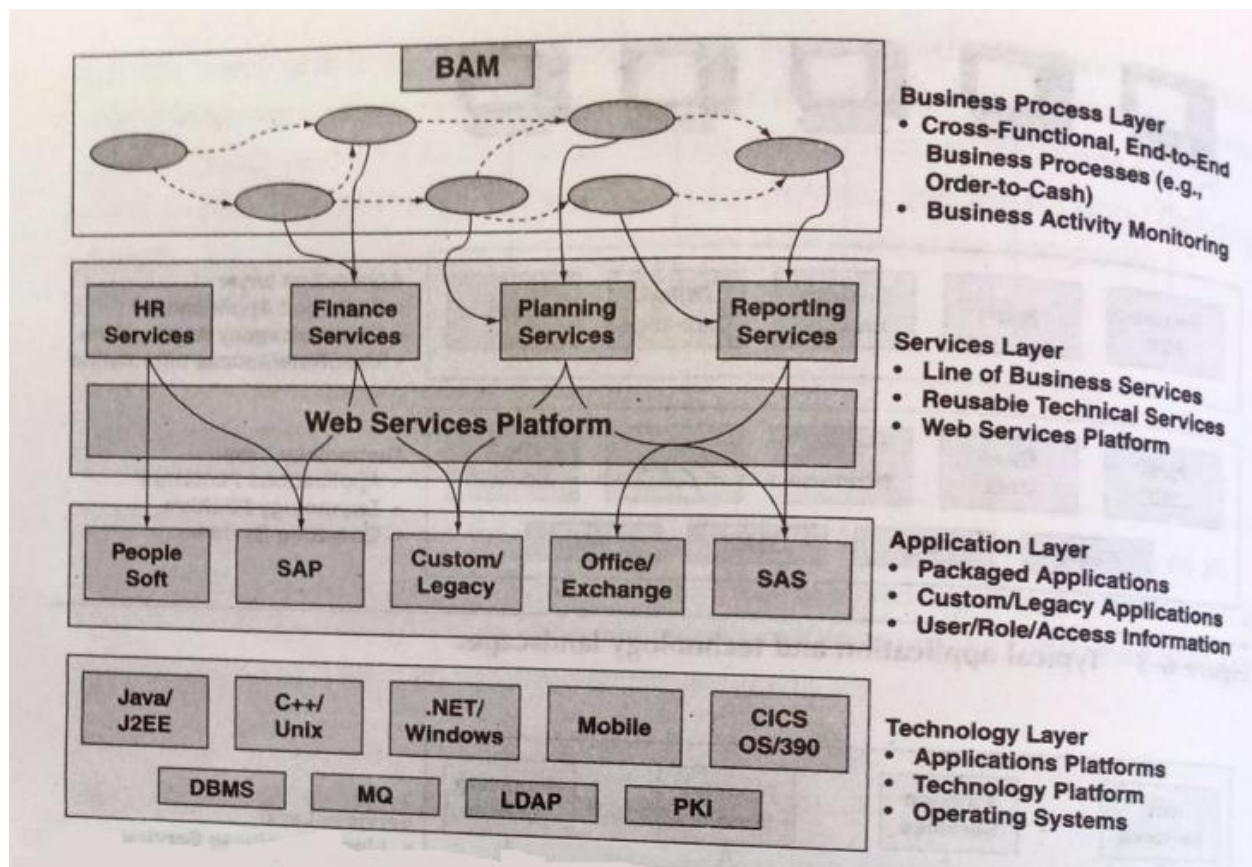
Line of Business Services-> Which aligns with the core business activities of an organization.

Reusable Technical Services-> Shares across multiple business domains.

Web Services Platform-> Which allows services to be defined & utilized in a manner. (i.e, Independent of application & Technology platform)

The next layer is the business process layer.

(BPL not responsible for knowing any details of underlying applications & technology platforms)



THE BUSINESS PROCESS LAYER USES THE SERVICES LAYER

Business Process Layer

Provides (or) describes level of details, B-functionality that maps to the business tasks in a business process.

Cross Functional-> Teams working together to achieve a common goal

End-to-End Business Processes-> Ex: Order-to-Cash

Business Activity Monitoring

The services layer provides the ideal platform for the business process layer for the following reasons:

- The line of business services provide coarse-grain business functionality that maps to the business tasks in a business process.
- The service contracts for the line of business services provide well-defined and unambiguous interfaces for accessing the services, and therefore the business process is not responsible for knowing any details of the underlying application and technology platforms.

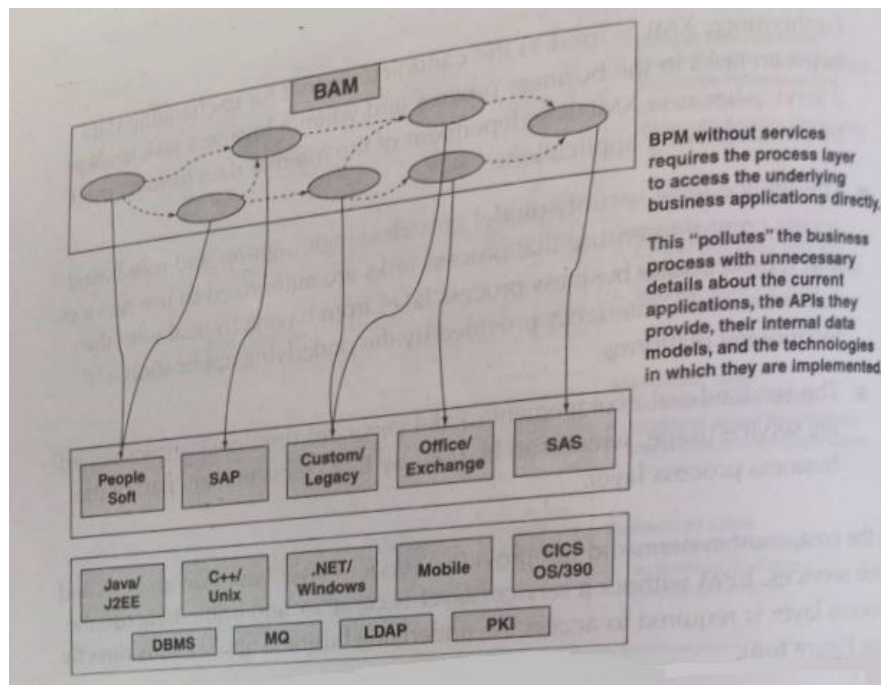
- The Service Registry & Service Discovery facilitates provided by the service layer ensure that the B-P Layer can dynamically locate & access services as necessary.
- The service-level data model is defined based on the business domain and is independent of the data model used by any particular application.
- Furthermore, XML, is used as the canonical format for exchanging data between tasks in the business process and when a business task invokes a service because XML is independent of the internal data formats used by the underlying applications.
- The service-level security model provides single sign-on and role-based access control to ensure that process tasks are authorized to use services, and it protects the business process layer from having to deal with the various security interfaces provided by the underlying application and technology platforms.

- The service-level management model generates run-time statistics regarding service usage, which can be used by BAM tools that are part of the business process layer.

In the past, most systems did not provide a services layer based on an SOA and Web services. BPM without a services layer is complex and brittle because the process layer is required to access the underlying business applications directly

This approach is more complex because the process layer must directly access existing applications using one or more interfaces defined by the application (e.g., APIs, messages, or database tables). This requires the process implementer to learn about these application interfaces and requires steps to be added to the business process to compensate for poorly defined application interfaces or for transforming application specific data into a canonical format that the business process can use.

This approach is more brittle (i.e., more likely to break) because the process is tightly coupled to specific applications and specific application interfaces. This means that something as simple as installing a newer version of an application that access it. This tight coupling also makes this approach harder to change. APIs, messages, or For instance, replacing an existing application with a new application from another that access the old application.



BPM WITHOUT SERVICES IS BRITTLE

ORCHESTRATION & CHOREOGRAPHY SPECIFICATIONS

Web services are foundation for creating the overall structure & implementing business process & working together within & across organizational boundaries.

2 Languages for WS Composition have emerged:-

They are:-

1. Business Process Execution Language (WS-BPEL)

Developed by BEA, IBM, Microsoft & Siebel & submitted to WS-BPEL Technical Committee at OASIS.

OASIS: “Organization for the advancement of structured information standard”

2. Choreography Description Language (WS-CDL)

Developed by the WS-Choreography working group at W3C, based on input specification written by Intalio, SUN, BEA & SAP.

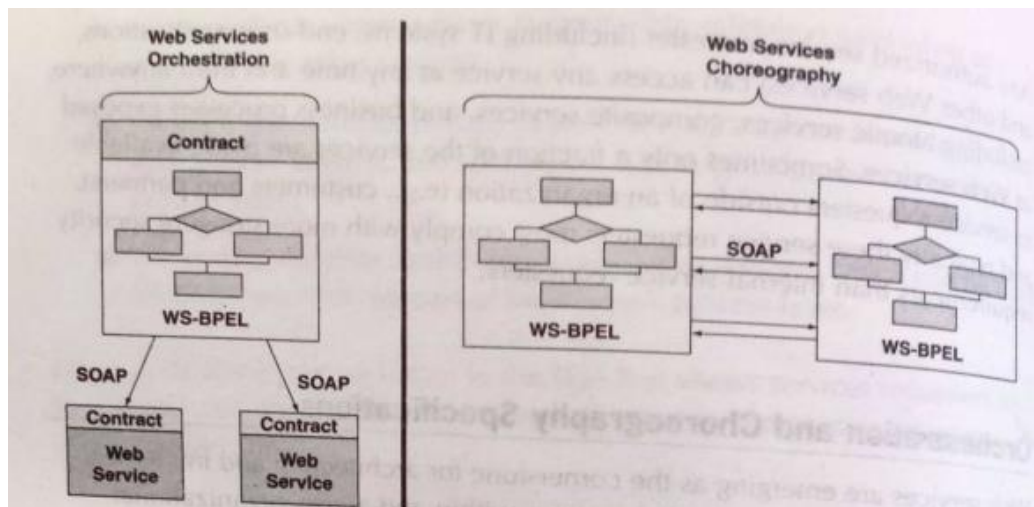
The goal of these Languages is to integrate WS together in a process oriented-way.

Comparing Web Services Orchestration & Choreography

The term Orchestration & Choreography are frequently used to describe 2 approaches to combine WS.

* WSO refers to compose web services for Business Process.

* WSC refers to compose web services for Business Collaborations.



COMPARING ORCHESTRATION & CHOREOGRAPHY

- **WSO** is used for defining composite services & internal processes that reuse existing WS. It also supports the preparation of information to exchange externally in WSC.
- WSC is used for defining how multiple parties collaborate in a peer-peer manner (Without need any central server can interact with each other)

EX: i.e, A->B

Can Both WSO & WSC be specified using a Common Language:

Yes, there's no doubt about the different characteristics of these interactions, because many WS-BPEL advocates says that WS-BPEL is the only language required for the interactions & executions.

However, disagreement remains over from others but both types of interactions can be specified using a single language. i.e, (A Complimentary Language is required that Single Language cannot handle everything)

♣ **WS-BPEL:** It is a process oriented composition language for web services.

WS-BPEL defines a set of basic tasks for creating WS Compositions:-

1. **Involve task**-> Allow the B-P to invoke a one-way (or) Request/Response operation.
2. **Receive task**-> Allows the B-P to do a blocking wait for a message to arrive.
3. **Reply task**-> Allows the B-P to send a message & reply to a received message.
4. **Wait task**-> Tells the process to wait for some time.
5. **Assign task**-> Copies data from one place to another.
6. **Throw task**-> Indicates that an error condition has occurred.
7. **Terminate task**-> Terminates the entire orchestration instance.

Structured tasks are used to combine the primitive tasks (basic) into more complex processes.

8. **Sequence task**-> Defines an ordered sequence of tasks.

9. **Switch task**-> Ability to select a particular branch based on conditional logic.

10. **Pick task**-> Block & wait for a suitable message to arrive (or) for a time-out alarm to go off.

11. **While task**-> Defines a group of tasks that should be repeated while condition is satisfied.

12. **Flow task**-> Collection of steps should be executed in parallel.



UNIT-IV

METADATA MANAGEMENT- P2

Metadata is data about data

(OR)

It refers to data that provides information about other data.

For example, a file is an entity and a file record layout is metadata.

* **File Record Layout** refers to structure (It describes how individual records are formatted & arranged, specifying the position & size of data each field)

Structure of the data such as:-

1. **Field Names:** Individual data fields within each record.
2. **Field Datatypes:** The type of data (text, numeric, date) that field can store.

3.Field Length: The no. of characters (or) bytes allocated for each field.

A web service may have a variety of M-D associated with it, including Datatypes & Structures for messages, message exchange patterns for exchanging messages, the n/w addresses of the endpoints that exchange messages and any requirements for extended features such as security, reliability or transactions.

W-S Metadata is important part of basic & SOA-based web services solutions. WS Metadata & M-D Management technologies includes the following:-

- **UDDI** -> A Registry & Repository for storing & retrieving WS.

It is a specification for distributed directory which allows businesses to list themselves on internet & discover each other services.

- **XML Schema** -> for defining datatypes (integer, string) & structures (format)
- **WSDL** -> for defining messages, message exchange patterns, interfaces & endpoints.
- **WS-Policy** -> declaring assertions for various qualities of service requirements, such as Reliability, Security & Transactions.
- **WS-Addressing** -> for defining WS endpoint references (identify the communication where it ends) & associated message properties.
- **WS- Metadata Exchange** -> for dynamically accessing XML, WSDL & WS-Policy Metadata when required.

These different kinds of metadata will work together to define the characteristics of any WS, from simple to complex.

Also, the metadata items are contained in XML files of changing definition and stores into a directory such as UDDI (or) LDAP for easy retrieval.

All of the metadata items benefit from the use of Naming Conventions.

Naming Conventions & Easy Retrieval

When thinking about “Metadata & Management”

It is important to develop a quality of storing & retrieving the metadata as it is to define it in the 1st place.

One of the reason UDDI failed to gain acceptance (Broad adoption) is doesn't provide sufficient methods for effectively categorizing & identifying metadata for easy search & retrieval.

It's an old saying in database world i.e,

That you have to know how you're going to retrieve something before you figure out, how you're going to store it & its definitely holds true for metadata.

Ex: How you name your service is important.

Because, you don't want to call the "Customer Lookup" service The customer lookup service is "Redundant" but you might want to call it the "Customer ID Lookup".

Similarly, for "Customer Name Lookup"

"it is important to use good names for data items so whoever ends up requesting the service can easily understand the data that the service provider.

These conventions are a part of any large project, also important for services designed for "Reuse" & "Interoperability".

Metadata Specification

Metadata specification refers to the guidelines and standards that define how metadata should be structured, represented, and used to describe, manage, and exchange information about data. Metadata is essentially "data about data," providing contextual information that makes the data more understandable and easier to work with. A metadata specification outlines the following key aspects:

Structure and Format:

Defines how metadata should be organized (e.g., hierarchical or flat structure).

Specifies the formats (Ex: XML, JSON, RDF, or proprietary formats).

Data Elements:

Describes the individual pieces of information required (e.g., title, author, date, version, keywords).

Includes standardized terms or fields for t

he metadata elements, such as Dublin Core, METS (Metadata Encoding and Transmission Standard), or schema.org for web data.

Vocabulary or Ontology:

Specifies any controlled vocabularies, taxonomies, or ontologies that should be used to standardize terminology in the metadata. This ensures consistency in the metadata descriptions.

Types of Metadata:

Descriptive: Information used for identification, discovery, and selection (e.g., title, description).

Structural: Details about the organization of the data (e.g., how pages in a book relate to each other).

Administrative: Information related to the management of the data (e.g., rights, provenance, and technical details).

Statistical/Quantitative: Often found in datasets, explaining measures like sampling methods or units of measurement.

Metadata Standards:

Specifies relevant industry or domain standards to be followed

Common metadata standards include:

Dublin Core for general resource description.

MARC (Machine-Readable Cataloging) for library resources.

EXIF (Exchangeable Image File Format) for image metadata.

Schema.org for web data

ISO 19115 for geographic information

Encoding Rules:

Defines how values for each metadata element should be encoded. For instance, a date might need to be formatted in ISO 8601, and a location might need to follow a specific coordinate system.

Interoperability:

Outlines how metadata should be exchanged across different systems or platforms, ensuring that metadata can be shared and interpreted across diverse environments.

Compliance and Validation:

Specifies rules for ensuring that the metadata conforms to the expected structure and content, including validation mechanisms and tools

Versioning and Updates:

Guidance on handling updates, revisions, and changes to metadata over time, including version control and history tracking.

A well-defined metadata specification helps organizations ensure consistency, improve data discoverability, facilitate sharing, and enhance long-term data management.

POLICY

A metadata policy is a formal document or set of guidelines that defines how metadata should be created, managed, used, and maintained within an organization or for a specific project. It establishes the rules and expectations around metadata management to ensure consistency, quality, and interoperability of data across systems. A good metadata policy helps improve data discoverability, usability, and long-term preservation.

Key Elements of a Metadata Policy:

Purpose and Scope:

Purpose: Why the metadata policy is needed (e.g., to improve data management, ensure consistency, enhance search and retrieval).

Scope: The range of data, systems, or departments the policy applies to (e.g., all digital assets, specific datasets, or specific domains like research or publications).

Roles and Responsibilities:

Data Stewards/Managers: Responsible for overseeing metadata quality and consistency.

Creators: Individuals or teams responsible for adding or updating metadata for datasets, documents, or other assets.

End Users: Those who access, search, or use metadata, and who may be responsible for ensuring metadata meets their needs.

Technical Support: Those responsible for implementing and maintaining the metadata infrastructure, tools, and standards.

Metadata Standards:

Defines the standards and schemas that should be followed (e.g., Dublin Core, ISO 19115 for geospatial data, Schema.org for web data).

Ensures consistency in the metadata format and terminology used across systems or datasets

Specifies any controlled vocabularies, taxonomies, or ontologies to standardize terms used in metadata.

Metadata Creation and Maintenance:

Creation: Guidelines on how metadata should be created when new data is produced (e.g., which fields need to be filled in, who should create it, and when it should be done).

Update and Review: Instructions for how metadata should be updated over time, especially for ongoing or evolving datasets, including a review process to ensure metadata remains relevant and accurate.

Quality Control and Validation:

Defines processes for ensuring metadata is complete, accurate, and follows the prescribed standards.

Automated Validation: Tools or scripts that automatically check the metadata for errors or missing fields

Manual Review: Procedures for manual quality assurance (QA) of metadata, especially for complex or critical datasets.

Access and Security:

Access Control: Defines who can create, edit, and access metadata, including any restrictions or permissions related to data sensitivity or privacy.

Security and Confidentiality: Guidelines for ensuring sensitive or private information in metadata is handled appropriately and complies with legal or regulatory requirements (e.g., GDPR).

Interoperability and Data Sharing:

Data Exchange: Outlines how metadata should be shared or exchanged between different systems, ensuring it is machine-readable and compatible with external systems.

APIs and Protocols: Guidelines for using standardized protocols (e.g., OAI-PMH, RESTful APIs) for metadata exchange.

Compliance and Legal Considerations:

Ensures metadata adheres to relevant laws, regulations, and standards (e.g., intellectual property, privacy laws, and industry-specific regulations).

Defines any requirements for long-term preservation of metadata to support compliance with archival standards

Training and Awareness:

Provides guidance on training users (e.g., data creators, managers, and end users) to ensure they understand the importance of metadata and how to create and maintain it correctly.

Promotes ongoing awareness within the organization about best practices and new developments in metadata standards

Monitoring and Auditing:

Defines how the implementation of the metadata policy will be monitored, including audits or regular checks to ensure compliance with the policy.

Specifies metrics for measuring the effectiveness of the metadata management processes

Versioning and Updates:

Outlines how the metadata policy itself should be versioned and updated, especially as new technologies, standards, or business needs arise.

Ensures that changes to the policy are documented and communicated to all relevant stakeholders

Benefits of a Metadata Policy:

Consistency: Standardizes metadata creation and management across the organization.

Efficiency: Saves time by setting clear guidelines and reducing confusion around metadata practices.

Data Discoverability: Helps users locate and use data more easily by ensuring metadata is accurate, comprehensive, and consistently applied.

Compliance: Ensures metadata management meets legal and regulatory requirements, such as data privacy laws or archiving standards.

Data Quality: Improves the overall quality of data by encouraging the use of structured, standardized metadata.

A well-defined metadata policy is essential for managing data assets efficiently, supporting data discovery, and ensuring compliance with internal and external requirements.

WS Metadata Exchange

Web Services (WS) Metadata Exchange refers to the exchange of metadata between different systems or services over the web. Metadata in this context generally describes the structure, operations, and capabilities of a web service, facilitating better integration and interoperability between different systems. The concept is often implemented using web standards like SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), and other related technologies.

Key Aspects of Web Services Metadata Exchange:

Web Services Description Language (WSDL):

WSDL is an XML-based language used to describe the functionality of a web service. It specifies the available operations, the input/output message formats, the protocols for communication (e.g., HTTP, SOAP), and where the service is located.

Web service clients use WSDL to understand how to interact with the web service (e.g., what parameters to pass, what responses to expect).

Web Services Metadata Exchange (WS MetadataExchange):

WS-MetadataExchange (WS-MEX) is a specification that defines a standard mechanism for the exchange of metadata about web services. It enables a service to provide its WSDL and other metadata documents in a standardized way.

WS-MEX typically operates over SOAP and allows clients to retrieve metadata (like WSDL) dynamically by making a request to a specific endpoint.

WS-MEX Endpoint: Typically, a WS-MEX endpoint is configured to serve metadata about web services. A service can expose its metadata at a well-known URL (like /mex) where clients can request metadata.

SOAP Request for Metadata: A WS-MEX client sends a SOAP message to a WS-MEX endpoint to request metadata, such as the WSDL or other related documents.

Benefits of Web Services Metadata Exchange:

Discoverability: By exposing metadata in a standardized format, web services make themselves discoverable, which simplifies integration with other services.

Interoperability: Standard metadata exchange enables different systems, possibly built on different technologies, to communicate seamlessly. WSDL and WS-MEX help bridge the gap between heterogeneous systems.

Dynamic Interaction: Clients can dynamically retrieve updated metadata (such as WSDL) from services, which is especially useful when the service definition changes.

Self-Describing Services: Services that provide metadata (e.g., WSDL) are self-describing, making it easier for

developers to understand how to interact with the service without needing in-depth knowledge of its internals.

WS-MEX Use Cases:

Service Discovery: When a client needs to find out about a service's operations, data types, and bindings, it can query the WS-MEX endpoint for the WSDL or other metadata.

Service Registration: Services may register their metadata with a central registry using WS-MEX so that clients can discover and use them dynamically.

Dynamic Binding: Clients can use WS-MEX to dynamically bind to a service at runtime based on the metadata retrieved from the service's WSDL.

How WS-MEX Works:

A service exposes metadata through a WS-MEX endpoint, typically at a URL like /mex.

A client sends a request (usually via SOAP) to the WS-MEX endpoint to retrieve the service's metadata, such as WSDL, schema, or other relevant documents.

The WS-MEX endpoint responds with the requested metadata, which the client can then use to interact with the service.

Example of WS-MEX Interaction:

Requesting Metadata: A client requests metadata from a service by sending a SOAP request to a WS-MEX endpoint

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:mex="http://schemas.xmlsoap.org/ws/2004/09/mex"
">

<soap:Body>

  <mex:GetMetadata>
```

```
<mex:MetadataReference
URI="http://example.com/service"/>

</mex:GetMetadata>

</soap:Body>

</soap:Envelope>
```

Service Responds with WSDL: The WS-MEX endpoint responds with the WSDL, which includes the operations, data types, and endpoint information needed by the client to interact with the service.

Example response (simplified):

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:mex="http://schemas.xmlsoap.org/ws/2004/09/mex
">

<soap:Body>
```

```
<mex:Metadata>  
  
  <wsdl:definitions  
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"  
  
  xmlns:tns="http://example.com/service">  
  
    <!-- WSDL content describing the service -->  
  
  </wsdl:definitions>  
  
</mex:Metadata>  
  
</soap:Body>  
  
</soap:Envelope>
```

WS-MEX vs WSDL:

WSDL is the most common metadata format for web services, providing detailed descriptions of the service's operations, types, and bindings. However, WS-MEX offers a more flexible, standardized way to exchange that metadata dynamically, often using SOAP.

While WSDL is a static document that describes the service, WS-MEX allows clients to request metadata in a more dynamic, real-time fashion.