

Choreography and Orchestration

UNIT-IV

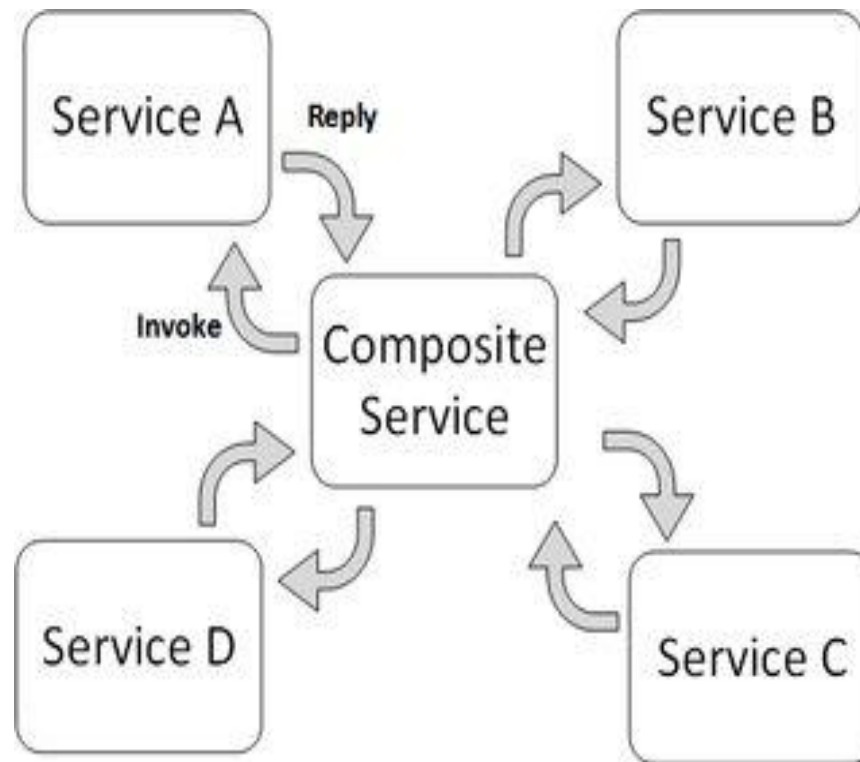
Introduction

- Basic technologies like XML, SOAP, WSDL enables to describe, locate, and invoke services as an entity
- But do not give a rich behavioral detail about the role of the service in more complex collaboration
- This collaboration includes a sequence of activities and relationships between activities, which build the business process
- There are two ways to build this process: service orchestration and service choreography

Orchestration

- *It is a centrally* controlled set of workflow logic facilitates interoperability between two or more different applications
- Ex. Enterprise application integration (EAI) middleware (integrate various legacy environments)
- A common implementation of orchestration is hub-and-spoke model
- It allows multiple external participants to interface with a central orchestration engine

Hub and Spoke Model



Advantages of Orchestration

- It accommodate merging of large business processes
- Different processes can be connected without having to redevelop
- Orchestration bridges this gap by introducing new workflow logic
- It can significantly reduce complexity of solution environments
- **Workflow logic is abstracted and more easily maintained than when embedded within individual solution components**

Contd...

- In SOA, extensions of business process logic can be express in a standardized way using orchestration
- Building services upon orchestration logic standardizes process representation across an organization
- Allows enterprise federation and promoting service-orientation
- **Web services Business Process Execution Language (WS-BPEL) - a primary industry specification that standardizes orchestration**

Business protocols and Process definition

- Workflow logic of an orchestration consists of numerous business rules, conditions, and events collectively
- **Business protocol** defines how participants can interoperate to achieve the completion of a business task based on workflow logic
- **Process definition** contains the details of the workflow logic expressed by an orchestration

Process services and Partner services

- Process participants are identified and described within a process definition
- The process itself is represented as a service, resulting in a *process service*
- Other services allowed to interact with the process service are called as *partner services or partner links*
- *Depending on the workflow* logic, a process service can be invoked by an external partner service, or it can invoke other partner services

Basic activities and structured activities

- WS-BPEL breaks down workflow logic into a series of predefined primitive activities
- Basic activities - receive, invoke, reply, throw, wait
- structured activities - sequence, switch, while, flow, pick
- Basic activities can be assembled by structured activities

Sequence

- Basic and structured activities can be organized to execute in predefined way
- A *sequence aligns groups of* related activities into a list of sequential execution order
- Sequences is used when output of one application logic is dependent on the outcome of another

Flows

- *Flow has groups of related activities with different execution way*
- *Pieces of application logic can execute concurrently within a flow*
(set of activities not needed to wait before another finishes)
- But flow itself does not finish until all encapsulated activities have completed processing
- It ensures synchronization among application logic residing in individual flows

Links

- *Links establish dependencies between activities that are part of flows*
- *Before an activity completes, it ensures that requirements of outgoing links first are met*
- *Similarly, before any linked activity begin requirements within any incoming links are met first*
- *Rules provided by links are referred as *synchronization dependencies**

Orchestration and SOA

- Orchestration enables service-oriented solution environments to be extensible and adaptive
- Orchestration is a key integration enabler in SOA

Choreography

- All organizations internal processes should be structured, to interoperate and realize an automation solutions
- Interoperation extends to collaboration, where multiple services from different organizations need to work together to achieve a common goal
- Web Services Choreography Description Language (WS-CDL) specification organize information exchange between multiple organizations on public collaboration

Contd...

- **Collaboration**
- Choreographies are intended for public message exchanges
- Goal is to establish organized collaboration between services representing different service entities
- Choreographies enables to establish universal interoperability patterns for common inter-organization business tasks

Contd...

- **Roles and participants**
- A Web service assumes one of a number of predefined *roles*
- *This represent the capability of the service* within the context of a particular business task
- Roles can be bound to WSDL definitions and are categorized as *participants (services)*.

Contd...

- **Relationships and channels**
- Each message exchange between two roles in a choreography is defined individually as a *relationship*
- *Every relationship* consists of exactly two roles
- *Channels* defines the characteristics (nature) of the message exchange between two specific roles
- Channel information can actually be passed around in a message
- It facilitates more complex exchanges involving multiple participants

Contd...

- **Interactions and work units**
- Actual logic behind a message exchange is encapsulated within an interaction
- Interactions are the fundamental building blocks of choreographies
- Completion of an interaction represents actual progress within a choreography
- Work units - impose rules and constraints that must be adhered to for an interaction to successfully complete

Contd...

- **Reusability, composability, and modularity**
- Each choreography can be designed in a reusable manner to be applied to different business tasks comprised of the same fundamental actions
- Using an import facility, a choreography can be assembled from independent *modules*
- *Modules* represent distinct sub-tasks and can be reused by numerous different choreographies

Difference between Orchestration and Choreography

- Service orchestration represents a single centralized executable business process (the orchestrator)
- Orchestrator coordinates the interaction among different services
- Orchestrator is responsible for invoking and combining the services
- The relationship between all the participating services are described by a single endpoint (i.e., the composite service)
- Orchestration includes the management of transactions between individual services
- Orchestration employs a centralized approach for service composition

Contd...

- Service choreography is a global description of the participating services
- It is defined by exchange of messages, rules of interaction and agreements between two or more endpoints
- Choreography employs a decentralized approach for service composition
- Choreography describes the interactions between multiple services, whereas orchestration represents control from one party's perspective
- i.e. **choreography** *differs* from an **orchestration** with respect to where the logic that controls the interactions between the services involved should reside