Contents

fundamental terminology and concepts

- design characteristic
  - its a specific **aspect** or quality of a solution logic
  - eg:vendor-neutral,loosely-coupled,distributable
  - soa characteristics: vendor-neutral,composition-centric,enterprise-centric,business-centric
  - object-oriented characteristics: encapsulation,abstraction,polymorphism,inheritance
- design principle
  - its a highly **recommended guidelines** to shape solution logic to realize specific design characteristics
  - eg:standardized service contract(1 of 8 soa principles),encapsulate what varies,open class for extension but closed for modification(related to decorator pattern),design loosely coupled components that interact(related to observer pattern),program to interfaces not implementation
- design paradigm
  - its an **approach** or technique to shaping solution logic to meet specific goals
  - its a set of **complementary design principles**
  - eg:service-orientation(to design distributable solution logic),object-orientation(to design componentized solution logic)
- design pattern
  - it describes a common problem,and corresponding solution
  - eg:proxy pattern,observer pattern,decorator pattern
- design pattern language
  - a sequence of related design patterns form the basis of design pattern language
- design standards,industry standards
  - **custom** standards apply to design of solution logic for a particular **enterprise**
  - **industry** standard refers to **open technologies** standards like those related to xml,web services
- best practice

- guidelines as a result of past experience
- guidelines in the form of general lessons learnt
- approach to solving or preventing certain problems
- eg:reusable logic to be maintained by a separate custodian
- **note**:design principle is related to design only whereas best practice can relate to anything

elements of service-oriented computing

- service-oriented architecture
  - its a distinct form of **distributed technology architecture** in support of services,service compositions,and service inventories
  - supports designing service oriented solutions which comprise of services,and service compositions adhering to service orientation(paradigm)
  - **types** or scopes
    i. service architecture
    ii. inventory
    iii. composition
    iv. enterprise
- service-orientation
  - its an approach to shaping solution logic to meet specific goals
  - its a **paradigm** which comprises of set of service-oriented design principles
  - service interoperability is a natural byproduct of application of service oriented design principles
- service-oriented computing
  - its a new generation **distributed computing platform** which is like an umbrella term comprising of many elements like SOA,service orientation
- service
  - its a **fundamental** unit of service oriented solution logic
  - exist as a physically independent software programs
  - a single service can provide a **collection of capabilities**
  - when a service is implemented as a component,capabilities are referred as methods,and when expressed as a part of service contract,they are called operations
- service composition
  - a coordinated aggregate of services
- service inventory
  - an independently standardized and governed collection of complementary services within a boundary
  - 2 **types**:**domain**,**enterprise**
  - **an enterprise service inventory is not comprised of domain service inventories**
  - a service inventory is considered to have **normalized** services when service boundaries within the inventory do not overlap with each other

- service inventory blueprint
    - conceptual blueprint of all the planned services of an inventory
    - also known as **service enterprise model** or **service inventory model**

strategic goals & benefits of service-oriented computing

strategy goals

1. increased intrinsic interoperability
    - interoperability refers to **sharing of data**,aml could be used as data format
2. increased federation
    - a federated IT environment is where applications are **united** by standardized service contracts while allowing individual service implementations to remain disparate and **independently governed**
3. increased vendor diversification options
    - achieved using standards based and vendor neutral implementations like web services platform
4. increased business & technology alignment
    - achieved through functional abstraction on many levels

benefits

5. increased ROI
6. increased business agility
7. reduced IT burden

service models & service layers

- service model is a **classification** to indicate to which predefined type(**utility**,**entity**,**task**) a service belongs to
- service model provides templates for common types of services
- service **models**,and service **layers** are used to classify and organize services within a service inventory
- **entity** services(agnostic) also known as **business entity services** ,represents business models like customer,invoices
- **task** services(non-agnostic) also known as **business process services**
- **utility** services(agnostic) are non-business centric,also known as application services,technology services,infrastructure services

analysis,service modeling,design

- **service modeling**
    - its a subprocess of service oriented analysis that produces conceptual service definitions called **service candidates**
- service modeling gradually results in defining **service inventory blueprint**
- service oriented **design** process uses a set of service candidates from service inventory blueprint as a starting point from which they are shaped into actual physical service contracts

SOA delivery approaches

1. top down
    - more time,effort,cost needed on analysis of service to be implemented
2. bottom up
    - quick analysis related to business requirements,its a tactical approach,could result in governance burden when business requirements change
3. agile delivery(meet in the middle)

SOA characteristics

- vendor neutral
    - related to 'increased vendor diversification options' goal of SOA
- business centric
    - related to 'increased business & technology alignment' goal of SOA
- enterprise centric(agnostic services)
    - related to 'increased intrinsic interoperability' goal of SOA
    - enterprise centric resources have following primary characteristics
        a. available beyond a specific implementation boundary
        b. designed according to established design principles,and standards
- composition centric
    - related to 'increased federation' goal of SOA

service-orientation design principles

1. Standardized service contract
    a. services within the same service inventory are in **compliance** with the same **contract design standards**
2. service loose coupling
    a. service contracts impose **low** consumer dependency requirements,and themselves **decoupled** from their surrounding environment
    b. this principle emphasizes loosening dependencies between service **contract**,service **implementation**,and service **consumer**
    c. it spans both inter,& intra service designs
3. service abstraction
    a. service contract only contain **essential** information,and information about services is limited to what is expressed in the service contracts
    b. emphasize is to hide as much of the underlying details as possible,this **supports** service loose coupling principle
4. service reusability
    a. services contain and express **agnostic** logic,and position themselves as reusable enterprise resources
5. service autonomy
    a. services exercise a **high level of control** over their underlying runtime execution environment
    b. this fosters increasing service **reliability**,and **behavioural predictability**

6. service statelessness
    a. services **minimize** resource consumption by deferring management of state information when necessary
    b. this fosters design characteristics like **availability**,and **scalability**
7. service discoverability
    a. services supplement **communicative** meta data to effectively discover,& interpret it
8. service composability
    a. services are effective participants of composition regardless of size & complexity of composition

Note

- principles that **regulate**
    a. service loose coupling
        - minimizes dependencies
    b. service abstraction
        - minimizes the availability of meta information
    c. service composability
        - maximizes composability
- principles that **implement**
    a. standardized service contract
        - implements standardized contract
    b. service autonomy
        - implements independent functional boundary & runtime environment
    c. service reusability
        - implements generic & reusable logic & contract
    d. service statelessness
        - implements management of statelessness logic
    e. service discovery
        - implements communicative meta information

common SOA technologies

1. cloud computing
2. web service
    a. comprises of the following
        i. technical service contract consisting of a wsdl definition,an xml schema definition
        ii. a body of programming logic
        iii. message processing logic

Cloud computing and SOA connection points

- cloud computing
    - its a style of distributed computing in which services,softwares,infrastructure are delivered to external customers using internet technologies

- https://www.youtube.com/watch?v=iMJCa4QoU8k&list=PL4EF34F7A4FC0B00F

notes

- when planning a transition toward SOA,we are usually required to balance the **strategic** goals(long term) with **tactical**(short term) requirements
- **service registry** is a product or technology that is key to facilitating service discovery and service governance in general
- a primary focus of **service modeling** is the abstraction and encapsulation of business logic in support of defining business service candidates
- **grid**,and **virtualization** in simple explanation
  - each software needs a container
  - virtualization can create multiple logical containers within a single physical system
  - if service is smaller than container,virtualization could be used
  - if service is larger than container,grid could be used
  - grid can uses multiple physical systems together