Software System Architecture and Design

Introduction to Service-Oriented Architecture
Let's begin with business conceptual model

Let's focus on business functions and business behaviour

- **Business functions:**
  - Organizations are established to perform specialized functions in the business
  - Frequent reorganization within any business is a recognized fact of life
  - Functions are idealized conceptual structures that are stable over time and in the face of managerial reorganization whims

- **Business behaviour:**
  - Business behaviour is an ordering of tasks or activities that accomplish business goals and satisfy business commitments
  - It may include manual or automated operations that complete units of work
  - **Business behaviour IS NOT STABLE**
Business functions
Business behaviour

- Also known as *business process*
Business and IT alignment problem

- Since business behaviour (business processes) is constantly changing, IT needs to look for something more stable to base their development on

- Business functions
  - They are more stable – they should be basement for IT development

- In short:
  - BAD if system development is based on business behaviour (i.e. business process is hardcoded within software system) – IT department won't keep up with business changes
  - BETTER if development is based on business functions (also known as business services) – create software services, that will be grouped into processes by some system that will automate business processes (workflow engine)
Ways to design software system

Enterprise Business Process Management – Architecture, Technology and Standards

BAD (two-layered):
1. services (S1-S4)
2. applications

GOOD (three-layered):
1. process
2. services
3. applications

This is SOA

Workflow engine, executing business process
Service-Oriented Architecture

- SOA is not a product (!)
  - You can not buy a SOA, though most companies sell SOA products (SOA enablers)
- It’s about **bridging the gap** between business and IT through a set of business-aligned IT services using a set of design principles, patterns, and techniques
- There is a huge amount of theory behind SOA: business process management (BMP), IT management, IT alignment with business
  - SOA is not Web Services!
Service-oriented architecture

Book: "Service Oriented Architecture For Dummies"

- Service oriented architecture is more than a bunch of new software products strung together to allow technology companies to have something else to sell.
- SOA represents a dramatic change in the relationship between business and IT
  - SOA makes technology a true business enabler and empowers business and technology leaders alike
- The software industry has been on a journey toward a service oriented approach to software for more than 20 years. Smart people have known for a long time that if software can be created in such a way that it can be reused, life will be a lot better
- If software can be designed to reflect the way business operates, business and technology can align themselves for success
- Finding good ways to reuse the years of investment in software means money spent wisely
Service-oriented architecture

Book: "Service Oriented Architecture For Dummies"

- Service-oriented architecture is
  - a software *architectural style*
  - for building applications that implement *business processes or services*
  - by using a set of *loosely coupled* black-box *components*
  - *orchestrated* (or taking part in *choreography*)
  - to deliver a well-defined *level of service*.

- Related terms:
  - QoS - Quality of Service
  - SLA - Service Level Agreement
Service-oriented architecture

Book: "Service Oriented Architecture For Dummies"

- **SOA is for building business applications**
  - Many legitimate approaches to software architecture exist, and SOA is not intended for building every kind of software. It is intended explicitly for building business applications.

- **SOA is a black-box component architectural style**
  - SOA deliberately hides complexity wherever possible, and the idea of the black box is integral to SOA. The black box enables the reuse of existing business applications by adding an adapter to them, no matter how they were built.

- **SOA components are loosely coupled** (mažai sukibę)
  - Components (black-boxes) talk to each other via interfaces. They are NOT dependent on particular component implementation.

- **SOA components are orchestrated to link together through business processes to deliver a well-defined level of service**
  - SOA creates an arrangement of components that can, collectively, deliver a very complex business service. Simultaneously, SOA must provide acceptable service levels (SLA – service level agreement). Service level is directly tied into the best practices of conducting business — commonly referred to as *business process management*.
SOA architectural style
(trivial conceptual model)

Consumer Obtains Service description from Broker or directly from Provider (Broker is optional)

Service Consumer - Service Description - Service provider

«use»

«contains»

«realize»

«described in»

Service Broker

+ invokeService ( )
+ bindToService ( )
+ findService ( )
SOA architectural style (conceptual model)

Book "SOA Patterns"
Web Services are only one of technologies that implement Software Services
Other technologies you could use: HTTP, RMI, CORBA, DCOM, ...
Business service

- The logical encapsulation of business function
  - In simple terms, we mean that you wrap up everything you have to do to make a particular business function happen and give that rolled-up something a name and call it a business service
- Let's take a restaurant as an example:
  - Everything the kitchen has to do to prepare the meal, from chopping vegetables to cooking to plating, could be called the meal-preparation service
  - Everything the server does to extract the order from us (elucidate menu items, tell us what isn’t available right now, suggest appetizers and side dishes, write down our order) could be rolled up into the order-taking service
Fundamental SOA components
Fundamental SOA components

- The **ESB** makes sure that messages get passed back and forth between the components of a SOA implementation.
- The **SOA registry** contains important reference information about where the components of a SOA are located.
- The **workflow engine** provides the technology to connect people to people, to connect people to processes and processes to processes.
- The **service broker** connects services to services, which in the end enables the flow of business process.
  - ESB usually acts as a service broker.
- The **SOA supervisor** makes sure that the platform underneath the SOA environment works in a consistent and predictable way.
  - Service level agreements (SLA), Quality-of-service (QoS)
ESB: Typical Enterprise systems integration
ESB: Typical Enterprise systems integration

- Each department builds its own systems.
- As people use the systems they find they need information from other systems and point-to-point integration emerges.
- The diagram shows 4 types of point-to-point integrations:
  - ETL (Extract, Transform Load) which is a DB to DB relation,
  - On-line and File Based which are application to application relations
  - direct connection to a DB which is an Application to database relation
- There are additional relation types like replication, message based etc.
- One attempt to solve these problems was EAI (Enterprise Application Integration) but it (alone) never really succeeded mostly because EAI is data centric and not process centric which means it cannot keep up with business process change – not to mention that EAI solutions are expensive to build and maintain
  - ESB is the same EAI, just not for data integration, but for service integration.
Web Services – “how simple they are...”

Diagram:
- Service Broker
- UDDI Inquiry: find_xxx
- UDDI Publish: save_xxx
- WSDL
- Service Provider
- Service Consumer
- Service
- Client
- XML
- SOAP
Really so simple? (URL)
Some WS-* standards

- WS-Addressing
- WS-Atomic Transaction
- WS-Business Activity
- WS-Coordination
- WS-Eventing
- WS-Notification
- WS-Reliable Messaging
- WS-Management

- WS-Policy
- WS-Secure Conversation
- WS-Security Policy
- WS-Security
- WS-Trust
- WS-Metadata Exchange
- WS-Transfer
SOA is not (only) Web Services!

If all my components communicate via Web Services, all I have is a bunch of Web Services! I do not yet have SOA!
SOA and Distributed Objects

Book "SOA Patterns"

- One of the ideas behind SOA is to set **explicit boundaries** between larger chunks of logic, where each chunk represents a **high cohesion** business area.
- This is an improvement on the more traditional (distributed objects) approach, which more often than not results in an unintelligible/unmanageable object soup.
Service-oriented approach to building

**Business Process**
- Receive Order
- Fulfill Order
- Restock
- Bill Customer

**Defined Services**
- Receive Order Service
- Fulfill Order Service
- Restock Service
- Customer Billing Service

**IT Systems**
- Web Application
- Retail System
- Warehouse System
- CRM
The OMG and Service Oriented Architecture

- Business Processes
- Interfaces defined by enterprise context
- Business Services
- Interfaces defined by enterprise semantics and requirements
- Components
- Operational Resources

- BPMN
- BPDM
- SVBR
- MDA-BPEL Profile
- XMI
- UML
- MDA-J2EE Profile
- MDA-Eclipse Profile
- MDA-CORBA Profile
SOA layers

(IBM SOMA – Service Oriented Modeling and Architecture)

1. Operational Systems
   - Packaged Application
   - Custom Application
   - OO Application

2. Service Components
   - Service Provider
   - Service Consumer

3. Services
   - Atomic and composite

4. Business Process
   - Orchestration; Composition

5. Consumers
   - B2C Portal
   - B2B

6. Integration (Enterprise Service Bus)

7. QoS Layer
   - Security
   - Management & Monitoring Infrastructure Services

8. Data Architecture (meta-data) & Business Intelligence

9. Governance

Atomic Service
Composite Service
Registry
Layer 1: Operational systems layer

- This consists of:
  - existing custom built applications – *legacy systems*, including existing CRM and ERP packaged applications,
  - and older object-oriented system implementations, as well as business intelligence applications.
- The composite layered architecture of an SOA can leverage existing systems and integrate them using service-oriented integration techniques.
Layer 2: Enterprise components layer

- This is the layer of enterprise components that are responsible for realizing functionality and maintaining the QoS of the exposed services.
  - These special components are a managed, governed set of enterprise assets that are funded at the enterprise or the business unit level.
  - As enterprise-scale assets, they are responsible for ensuring conformance to SLAs through the application of architectural best practices.
- This layer typically uses container-based technologies such as application servers to implement the components, workload management, high-availability, and load balancing.
Layer 3: Services layer

- The services the business chooses to fund and expose reside in this layer.
  - They can be *discovered* or be statically bound and then invoked, or possibly, choreographed into a composite service.
- This service exposure layer also provides for the mechanism to take enterprise scale components, business unit specific components, and in some cases, project-specific components, and externalizes a subset of their interfaces in the form of service descriptions.
- Thus, the enterprise components provide service realization at runtime using the functionality provided by their interfaces.
- The interfaces get exported out as service descriptions in this layer, where they are exposed for use.
  - They can exist in isolation or as a composite service.
Layer 4: Business process composition layer

- Compositions and choreographies of services exposed in Layer 3 are defined in this layer.
- Services are bundled into a flow through orchestration or choreography, and thus act together as a single application.
- These applications support specific use cases and business processes.
- Visual service flow composition tools, provided by many vendors, can be used for the design of application flow.
Layer 5: Access or presentation layer

- Although this layer is usually out of scope for discussions around a SOA, it is gradually becoming more relevant.
  - There is an increasing convergence of standards, such as Web Services for Remote Portlets Version 2.0 and other technologies, that seek to leverage Web services at the application interface or presentation level.
- You can think of it as a future layer that you need to take into account for future solutions.
- It is also important to note that:
  - SOA decouples the user interface from the components,
  - and that you ultimately need to provide an end-to-end solution from an access channel to a service or composition of services.
Layer 6: Integration (ESB – Enterprise Services Bus)

- This layer enables the integration of services through the introduction of a reliable set of capabilities, such as:
  - intelligent routing, protocol mediation, and other transformation mechanisms, often described as the ESB.
- Web Services Description Language (WSDL) specifies a binding, which implies a location where the service is provided.
- On the other hand, an ESB provides a location independent mechanism for integration.
Layer 7: QoS (Quality of Service)

- This layer provides the capabilities required to monitor, manage, and maintain QoS such as security, performance, and availability.
- This is a background process through sense-and-respond mechanisms and tools that monitor the health of SOA applications including:
  - the all important standards implementations of WS-Management,
  - other relevant protocols and standards that implement quality of service for a SOA.
Layer 8: Information architecture and business intelligence layer

- The information architecture and business intelligence layer ensures the inclusion of key considerations pertaining to data architecture and information architectures that can also be used as the basis for the creation of business intelligence through data marts and data warehouses
  - This includes metadata content, which is stored in this layer, as well as information architecture and business intelligence considerations
- Especially applicable to industry-specific SOA solutions, this layer captures cross-industry and industry-specific data structures, XML-based metadata architectures, and business protocols of exchanging business data
  - Some discovery, data mining, and analytic modeling of data is also covered in this layer
Layer 9: Governance layer

- The governance layer covers all aspects of business operational life-cycle management in SOA.
  - It provides *guidance* and *policies* for making decisions about an SOA and managing all aspects of an SOA solution, including *capacity, performance, security, and monitoring*.
- This layer can be applied to all the other layers in the SOA solution stack.
  - Since it helps enforce QoS and make appropriate application of performance metrics, it is well connected with layer 7.
- The architectural decisions in this layer are encapsulated in consulting practices, frameworks, architectural artefacts, documentation of SOA capacity planning, any SOA-solution SLAs, SOA performance-monitoring policies, and SOA solution-level security-enablement guidelines.
Registries

- Layers 1, 6, 7, and 8 have registries associated with them.
- In layer 1 the registry tracks the existing services that can be found in the operational layer.
  - SOA will be built incrementally, project by project, in most cases.
  - After one project is complete, another project can leverage the existing assets provided by the previous project.
- The QoS layer (layer 7) will store access control lists (ACLs) or Lightweight Directory Access Protocol (LDAP) entries in registries for its operational and non-functional concerns.
- The data architecture layer (layer 8) includes databases to store metadata and other data structures.
SOA layers - example
Facts about SOA 😊

- In the last year, SOA increased Turkey's GDP by a factor of 10.
- Guns don't kill people, the SOA WS-* stack kills people.
- SOA can write and compile itself.
- SOA is the mistress to all CIOs.
- SOA is just one letter away from SOB. On purpose.
- If a tree falls in the forest, SOA knows about it.
- SOA is being used in the developing world to solve hunger. Entire populations will be fed on future business value.
- SOA can always win at TicTacToe. Even if you go first.
- Java EE can sometimes make a diamond from a lump of coal. SOA can make diamonds from air.
- SOA knows what you did last summer, and is disappointed that it wasn't SOA.
- SOA violates the first and third laws of thermodynamics. But not the second, as all energy flows from SOA.
- On the eight day, God created SOA, then SOA created Rock and Roll.
- For years Theoretical Physicists have searched for a grand unified theory that explains the architecture of everything ... DUH ... SOA
- SOA can't be named BOA (Business Oriented Architecture) since that would be too constricting for SOA.
- SOA is also a yoga posture that consists of performing all other yoga postures simultaneously.
- Dante has a special level in hell for consultants whose resumes do not say SOA.
- SOA is a power source more efficient than nuclear, cleaner than solar/wind, more available than coal, and more geopolitically stable than oil. Its too bad you can't afford it.
- SOA can do it in one line.
- "It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of SOA." - Jane Austen, opening line to Pride and Prejudice.
- The first rule of SOA is you do not talk about SOA.
- SOA in a Nutshell is 7,351 pages spread over 10 volumes.
- SOA actually stands for SOA Oriented Architecture. Let THAT bake your noodle.
- If you plug SOA into the back of your head, you'll know Kung Fu.
- Neo didn't bring down the Matrix. SOA did.
- Implementing SOA for the first time is the triumph of imagination over intelligence.
- Implementing SOA for the second time is the triumph of hope over experience.
- Unlike its predecessors, SOA actually is a real silver bullet. Capable of slaying legacy apps and werewolves. And your career if you really mess up the implementation.
- **Software architects don't use SOA. SOA uses software architects.**
Reading

- Highly recommended:
  - Design an SOA solution using a reference architecture

- Other articles:
  - BPM and SOA: Where does one end and other begin?
  - The OMG and Service Oriented Architecture

- Books
  - Service Oriented Architecture For Dummies, 2007
  - Enterprise Service Oriented Architectures; Concepts, Challenges, Recommendations, 2006