SOA
(Service Oriented Architecture)

Reda Bendraou
Plan

• IS & Architecture: Historical evolutions
• IT Planning
• SOA (Software Oriented Architecture)
• SOA: Key Concepts
  – Reference Model
  – Business Processes
  – Services
  – ESB
  – BPMN/BPEL
• SOA: Transversal Aspects (methodology, security, monitoring, reporting, etc.)
• SOA: Feedbacks from the industry
• Wrap-up!
IS & Architecture: Historical evolutions
IS & Architecture: Historical evolutions

• **Mainframes**
  – Unique Server, all the application aspects/layers are deployed on the same server (Monolithic)
    • Plus: Reliability and consistency
    • flaws: costly, hard to maintain

• **Client/Server**
  – GUI deported on the customer’s machines
  – Introduced because of the decrease in PC prices
  – Modules on the client side=> needed to be updated for each IS evolution
    • Plus: Server side get less solicited
    • flaws: costly to maintain the client side

• **Web Applications / Internet**
  – Client side: only need a browser!
  – E-commerce emergence
    • Plus: client side is generated on the server side, no need to update clients
    • flaws: Load on the server side 😞
IS & Architecture: Historical evolutions

• **N-Tiers**
  – Processors and Memory getting cheaper and cheaper
  – Application’s modules are deployed on different servers for more scalability
    • **Plus:** more robustness, performance,
    • **Flaws:** the IS is now scattered in different locations; becomes harder to get the global state of the system, to maintain consistency, **Harder to have a global image of the IS**

• **Cloud Computing**
  – The notion of “Pay as you Go”, Processors, Memory, HD, Apps
  – Notion of SaaS (Software As A Service, ex. *Customer Relationship Management* (CRM)) & PaaS (Plateform As A Service, Server + DB, etc.)
    • **Plus:** High Performance and robustness, economical model
    • **Flaws:** juridical gaps, Data outsourcing, security, dependability to the Internet, **Single Cloud of Failure**, Web services calls are very costly!!!
IS & Architecture: Historical evolutions

• Current IS reflect quite well the technological evolutions
  – Many different modules, technologically heterogeneous, multiplication of data sources, notion of Silos, layers (we build on top of existing layers) => erosion of the IS
Challenges

• Homogenize this notion of Silos (Repositories of data)

• Constant need for Agility
  – More productivity
  – More reactivity in integrating and absorbing new IS (acquisitions)
Challenges

- QoS : quality of service
- System’s global Performance
- Creating Value !!!
IT Planning
IT Planning: Definition

- IT Planning is a process to establish clear objectives for IT organizations that link directly back to the enterprise's strategic business goals.

- Allows to continuously drive the IT progression/ Evolution

- Quite similar to the Urbanism domain
  - processes
  - Layers
  - Notion of districts
IT Planning : 2 principal rules

• Application’s modules must be focusing on one and only one aspect/ functionality of the system
  – A road map is needed to reach each module’s goal

• Loosely coupling / High consistency
  – Between applications
  – Between application’s modules
  – Between module's classes/services
IT Planning: the reference model

- Business Processes and documents
- Functions
- Softs and Data
- Physical infrastructure
IT Planning: Challenges

• To maintain horizontal and vertical coherence
  – To be able to concretize Enterprise’s new objectives
    • At least to say if it is feasible or not
  – IT Planning could have as a main goal just to know about the existent system!
IT Planning Vs. Enterprise Architecture

• **Enterprise Architecture**
  – Set of processes and objects
  – Softs and infrastructure that supports the IS system

=> **IT Planning**
  – Tools, methodologies and concepts used to concretize the Enterprise’s objectives
  – EA = the target
  – IT Planning= the process
IT Planning: Process

- To map the existent system (cartography)
- Define the target (EA)
  - Enterprise Architecture
    - Objects (data)
    - Processes (and events)
    - Services
  - Physical Infrastructure
- Technological choices
- Progression roadmap
- Project management
Two ways

- **Big-Bang or Progressive**
  
  Big-bang not realistic for complex systems
IT Requirements

- **Business Requirements**
  - *Agility*: quickly integrate new processes
  - Global vision of the system
  - Reduce costs

- **Technical Requirements**
  - More reusability
  - Loosely coupling (Event-Driven Architecture, Asynchronous calls, etc.)
  - Security, transactions, QoS
  - Maintaining coherence of Data repositories

=>Next, how SOA could be the answer to these requirements
SOA
(Service Oriented Architecture)
SOA: Service Oriented Architecture

• A software architectural style in which the enterprise’s business processes play a primary role. They orchestrate the execution of services provided by application’s components
A Good Architecture is...

- **Modular** => Minimizing interactions / dependencies
- **Standard** => use of standards (best practices)
- **Evolvable** => Agile
- **Survivable** => Back-up / Security aspects
SOA is not...

- A technology
  - An approach, a vision, architectural style...

- SOA is not Web Services

- SOA doesn’t necessarily imply the use of EAI/ESB

- SOA is not the solution for all your performance issues
SOA: Key Concepts
SOA: Reference Architecture

Credit to IBM
SOA: Design Principles
Thomas Erl

- Service description in a standard language
- Services loosely coupled
- The right abstraction of services (granularity)
- Reuse of services
- Stateless
- A directory of web services (search)
- Services composition
Combining SOA + EDA

- EDA (Event-Driven Architecture): An architectural style where software components communicate in an asynchronous way using the publish/subscribe paradigm
SOA Components

• XML
  – Standard for data exchange

• Business Processes
  – BPMN/BPEL
    • Modeling/Execution languages for business processes

• Services
  – Reusable and autonomous functionalities
  – Can be of different kind (authentication, functional, security, etc.)
  – Web Services (SOAP, WDSL, UDDI, etc.)

• EAI=>ESB
  – Data integration in the IS
  – Very helpful in SOA but not mandatory

• Methodology
  – No consensus so far
Business Processes
"a structured, measured set of activities designed to produce a specific output for a particular customer or market. It implies a strong emphasis on how work is done within an organization, in contrast to a product focus’s emphasis on what. A process is thus a specific ordering of work activities across time and space, with a beginning and an end, and clearly defined inputs and outputs: a structure for process actions. Processes are the structure by which an organization does what is necessary to produce value for its customers."

Business process

• Describes the business concerns and not the system itself

• A process can be transversal to many applications, organization's departments, etc.
Business process categories

• **Customer processes**: directly bring value to the customer
  – ex: online shopping, delivery etc.

• **Sustaining processes**: indirectly bring value to the customer
  – ex: Updating the online catalogue, delivery

• **Enabling processes**: for managing internal services (not related to the customer)
  – Employee salaries, printing, monitoring etc.
Business Process Modeling / Execution languages

• BPMN (Business Process Modeling Notation): OMG standard for business process modeling (the UML equivalent to BP)

• WS4BPEL (Business Process Execution Language for Web Services): XML-based dialect for web service orchestration

• Automatic generation from BPMN towards BPEL
  – Attention: difference in abstraction levels
  – One necessarily a one to one mapping between BPMN and BPEL

• For Human interaction activities => The BPEL4People solution
Choreography Vs. Orchestration

• **Choreography**
  – Coordinating and synchronizing the execution of many business processes running concurrently and exchanging messages (ex. WS-CDL language)

• **Orchestration**
  – Coordinating the execution of many services of the same business process (using BPEL for instance)
Choreography Vs. Orchestration

Choreography

Orchestration
SOA/BPM offer

• **Commercial tools**
  – IBM WPS (WebSphere Process Server) => IBM Business Process Manager
  – Oracle SOA Product line
  – TIBCO ActiveMatrix
  – Jboss (jBPM)
  – Enterprise Architecte
  – Modelio
  – MagicDraw

• **Open Source**
  – Intalio BPMS
  – JBOSS BPM
  – Apache ODE
  – Active BPEL
  – Sun Glassfish
Notion of Service
Service

- A processing unit which provides an interface described in a standard and neutral language (technology independent) and which is physically deployed on a machine
Service

- **Interface**: contains 1 to N methods
- **Reusable unit**
- **Must have a QoS assigned to it**
- **Can be a provider/consumer of service**
Categories of Services

• **Business Entity Services**
  – Provides methods to manipulate Business entities
  – Make sure that business constraints are enforced

• **Business Services**
  – Provides methods that realize a specific business functionality.
    Usually it needs to call business entity services for that

• **Technical Services**
  – Provides technical services
    • Authentication services
    • Mailing, printing, etc,
Service Contract

• *Defines the service’s methods: signature, protocols, and QoS*
  – WSDL (SOAP) ans WADL (REST)
How to identify Services

• One of the most important aspects in SOA

• **Service Granularity is fundamental**
  – Determines the service reusability

• SOA’s success depends on % of service reusability

• Too fine grained:
  – Many interactions => Performance issues

• Too coarse grained
  – No reusable

=>Need to find a good balance
How to identify Services?

- For a better identification of services, we need to conciliate 2 approaches
  - Top-down and bottom-up
How to identify Services

• No need to publish all the identified services:
  – Each a service implies a cost and a risk
  – Avoid services proliferation

• Example the “Service Litmus Test” by IBM
Orchestration Vs. Propagation of Services

- **Promote Orchestration**
Enterprise Service Bus (ESB)
ESB in SOA

• A medium for integrating the different components of the enterprise’s IS (applications, resources, directories, etc.)

• Use standard protocols
  – SOAP, WSDL, binding HTTP and JMS
  – WS-ReliableMessaging, WS-Transaction, etc.
  – Orchestration avec BPEL

• It’s not mandatory in a SOA architecture but can be of a great help
ESB Vs EAI

• **EAI** : Hub and Spoke based solution
  – Do not scale (centralized)
  – SPOF (single point of failure)
  – Use of proprietary protocols

• **ESB**: next generation of EAI
  – Distributed
  – Use only standards

• **ETL et EII**, have different objectives
ESB: Why use them?

- **Use of standards**
  - XML, JMS, JCA, JMX.

- **A faster integration of new modules and services**

- **Use of directories for discovering and using services**
  - Eases the routing of messages between the modules

- **Service based architecture**
The ESB offer

- **Commercial Products**
  - IBM Websphere ESB
  - Oracle Enterprise Service
  - TIBCO Business Works

- **Open Source**
  - Apache ServiceMix, Synapse
  - Mule
  - JBoss ESB
  - Glassfish (sun open ESB)
  - Spring Integration
ETL (Extract / Transform / Load)

- Used for handling the transfer of a big amount of data between different applications
  - Routing, extraction, transformation
BPMN
(Business Process Modeling Notation)
BPMN?

• A graphical language for business process modeling

• An OMG standard, Ver2.0 Since 2011

• BPMN models can used to generate BPEL (Business Process Execution Language) code

• It’s not a methodology or a framework
BPMN: Notations

- Key Concepts
BPMN: Notations

Flow Objects
- Events
- Activities
- Gateways

Connectors
- Sequence Flow
- Message Flow
- Association

Artifacts
- Data Object
- Text Annotation
- «Group»
BPMN: Activity

- Represent a work to be done in the process

- Can be:
  - Atomic (a task)
    - Ex. Send Invoice
  - Composite (Sub-process)
    - Composed of other activities
  - Repetitive
BPMN: Event

- Event: something that may happen during the process execution

- An event can start, interrupt or end an execution flow

- Notation

[Diagram showing start, intermediate, and end symbols]
BPMN: Events

• **Start Events** say when the process should start

• **Intermediate Events** are triggered after the start of a process and before it finishes

• **End Events** designate the end of the process
Intermediate Events: Examples

- When receiving the « Voting Response » message, start « Increment Tally » activity.

- When the event is attached to the activity this means that triggering the event will interrupt its execution.
BPMN: Gateways

• Control the execution flow of the process (convergence or divergence points)
BPMN: Gatways, Examples

- Exclusive choice

- Inclusive choice, many options possible at the same time

- Concurrent execution
BPMN: Connectors

- Sequence is used to determine the execution order between activities

- Message Flow: to show how messages flow in the process

- To link artifacts to object flow
BPMN: Connectors, Example

- Example of connectors: Association and Sequence
BPMN: Conclusion

• A modeling language for communicating and reasoning around business processes

• A source for generating BPEL code

• Not always implemented correctly by tools
BPMN: References

• BPMN Official Page
  http://www.bpmn.org/
• BPMN Wikipedia
• BPMN Specification (v2.0 Beta 2)
  http://www.omg.org/spec/BPMN/Current/
• BPMN Specification (v1.2 Formal)
  http://www.omg.org/spec/BPMN/1.2/
• BPMN Examples (v2.0 Beta 2)
  http://www.omg.org/spec/BPMN/2.0/examples/PDF
WS-BPEL ou BPEL
(WS-Business Process Execution Language)
BPEL Historical

- WSFL, May 2001 (IBM) The Web Services Flow Language
- XLANG, May 2001 (Microsoft)
- BPEL 1.0, July 2002 (BEA, IBM, Microsoft) A merger of WSFL and XLANG
- BPEL4WS 1.1, March 2003 (BEA, IBM, Microsoft, SAP, Siebel) The specification submitted to OASIS
- WS-BPEL 2.0, March 2007 (OASIS: 39 companies as members of the technical committee) The first version of the "standard" blessed by a standards organization
WS-BPEL

• A standard for describing the orchestration (execution) of web services

• Comes with traditional programming languages constructs
  – sequence, alternative, iteration
  – variable, affectation, scoping variables
  – exceptions

• It’s a reusable definition in form of WDSL
  – The process (BPEL code) is considered as a service its self and can be part of a more complex process (other BPEL code)
WS-BPEL: How it Works?

BPEL Interpreter

Lecture

Requête

S1

S2

S3

S4

BPEL file

WSDL S1

WSDL S2

WSDL S3

WSDL S4
Structure of a BPEL code: Example

```xml
<process>
  <!-- web services participating in the process-->
  <partnerLinks> ... </partnerLinks>
  <!-- Variables used by the process-->
  <variables> ... </variables>
  <!-- used for asynchronous calls-->
  <correlationSets> ... </correlationSets>
  <!-- exceptions handlers -->
  <faultHandlers> ... </faultHandlers>
  <!-- handlers for transactions in “recovery” mode -->
  <compensationHandlers> ... </compensationHandlers>
  <!-- event handlers-->
  <eventHandlers> ... </eventHandlers>
  <!-- Flux d’activités du processus -->
  (activities)*
</process>
```
WS-BPEL: Liste of d’activity kinds

“Basic activities”

<invoke> sends a message to a port of a partner
<receive> blocking wait of a message
<reply> sends a message replying to a received message (by <receive>)
<wait> blocks the execution for a given duration or until an instant
<throw>, <rethrow> throws an exception
<assign> assigns a value to a variable
<exit> terminates the process
<compensate> executes the compensate field
<link> expresses dependencies between activities (may have a transition condition)
<empty> nop

“Structured activities”

<sequence> sequential execution
<flow> parallel execution
<if> conditional execution (else branch is mandatory)
<while>, <repeatUntil>, <forEach> iteration
<scope> defines an activity with its own variables, handlers, …
<pick> blocks the execution until a message/timeout occurs
Example using BPEL with a graphical editor
Mapping BPMN vers BPEL

Possible mapping
BPEL: Advantages

• Ensure interoperability

• A clear separation between the business logic (BPMN) and the process execution
  – More agility to enterprises

• The entire process can be viewed as Service
BPEL: Limits

• BPEL still not that mature concerning security issues and Tasks requiring human interactions

• But some standard has been issued:
  – WS-Policy
  – WS-HumanTask
  – WS-Reliability, WS-Security, ...
  – ...


- Transversal Aspects
SOA: Methodology

- Many proprietary solutions
- Exp. Praxème Method, or IBM’s SOA (SOMA) method
- Always combine a mix between bottom-up and top-down
SOA: Methodology

• **Top-Down**

  – Model business processes and business entities

  – Decompose the system into functional blocks around the business entities

  – Refine the business process into a more detailed version with a clear distinction between human tasks and automated tasks (future services)

  – Align the newly discovered services with the existing blocks (software components)

  – Explicitly model what we call Pivot Objects (objects exchanged between the different application of the information system)
SOA: Methodology

• Bottom-Up

  – Identify the different services already implemented by the system’s components

  – Façade these services with the new process activities

  – Create connectors and adaptors if needed
QoS Quality of Service

- **Aim:** define for each service a probe to monitor the following information
  - **Service Info**
    - Service name
    - Version
    - Semantics Description
  - **Service State**
    - Inactive, busy, stopped, in error state, over solicited
  - **Service Metrics**
    - Number of requests, time to answer, requests that resulted to error state
  - **Service’s interdependency with other services**
QoS

• These indicators will be used to:
  – Count the number of web service requests for reporting matters
  – Anticipate scalability issues and switch the flow to other servers if needed
  – Watch SLA constraints (Service Level Agreement)
  – Check services versions (date end of service)
  – Follow the process from end to end
BAM: Business Activity Monitoring

- Live process execution
- Allows a better reactivity in case of issues
- A good way to anticipate a blocking state
BAM: Business Activity Monitoring
BAM Offer

• **Commercial tools**
  – IBM Websphere Business Monitor
  – BEA Aqualogic BPM
  – ORACLE BAM
  – TIBCO BusinessFactor & OpsFactor

• **Open-source**
  • Pentaho BI
  • Spago BI
Security

- **XML Encryption** (W3C): Message level confidentiality
- **XML Signature** (W3C): Message level integrity, non repudiation
- **XML Key Management System** or XKMS (W3C): XML based PKI
- **Security Assertions Markup Language** or SAML (OASIS): SSO (Single SignOn), Authentication and Authorization
- **WS-Security** (OASIS): SOAP message security
- **XACML** (OASIS): Access control and policy management
- **WS-Trust** (OASIS): Trust management
- **WS-Policy** (W3C): Policy management
- **WS-SecureConversation** (OASIS): Secure session management
- **XBCF** (OASIS): Biometrics
- **SPML** (OASIS): Service provisioning
- **Project Liberty** (Sun etc): Federated identity
- **WS-Federation** (MS, IBM): Federated identity
SOA: Success & Failure Stories
# Some Projects

<table>
<thead>
<tr>
<th>Domain</th>
<th>Project Type</th>
<th>Infrastructure</th>
<th>Nbr of Services</th>
<th>QoS</th>
<th>Charges J/H</th>
<th>state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank</td>
<td>Redesign</td>
<td>JEE, JSF, XML, + Frameworks Open source</td>
<td>&lt;20</td>
<td>Critical</td>
<td>4500</td>
<td>Prod</td>
</tr>
<tr>
<td>Stock Exchange</td>
<td>Project</td>
<td>.NET, smart Client</td>
<td>&lt;20</td>
<td>Critical</td>
<td>1200</td>
<td>Prod</td>
</tr>
<tr>
<td>Insurance</td>
<td>Redesign</td>
<td>JEE, + Frameworks Open source</td>
<td>&gt;500</td>
<td>Critical</td>
<td>&gt;20000</td>
<td>Prod</td>
</tr>
<tr>
<td>Services</td>
<td>Project</td>
<td>JEE, + Frameworks Open source</td>
<td>&lt;20</td>
<td>Critical</td>
<td>2000</td>
<td>Prod</td>
</tr>
</tbody>
</table>
Main Mistakes to avoid?

It’s to see SOA only from the technological angle and to focus only on Web Services. SOA is, first and before all, about business processes, activities and human interactions.
What kind of Design errors are you facing in SOA projects?

‘I observed an accumulation (anarchic) of web services as well as the difficulty to link them to the business processes and enterprise’s objectives. They should be aligned on the business’’
Why SOA projects fail?

• Lack of team’s adhesion to the SOA Vision
• Tools are not mature enough
• No dissociation between SOA and WS
• Abusing of Web Services instead of focusing on Business Processes
• Scalability issues (the Service model is quite costly)
Success Factors

• Use of Components (loosely coupled)

• Putting in place a committee for watching that SOA Principles are always applied and followed by the development teams

• Use of best practices and document them inside the organization

• In earlier development stages, define KPI (Key Performance Indicators) and implement them in the future system. This will help watching continuously the system’s performance and QoS.
Advantages et drawbaeks

Advantages
• adaptive Architecture
• Reuse of code (components)
• Use of standards
• Better activity but in the long run (don’t expect results in the short term)

Drawbacks
• Tools are not mature enough
• Latency
• Hard to implement sometimes
• Qos hard to check sometimes
Next Step?
Références

• La série de livres sur SOA de Thomas Erl
  – www.thomaserl.com

• Livres
  – Open source SOA in Action, de Jeff Davis, édition Manning, 2009
  – Open source ESB in Action, de Tijs Rademakers, édition Manning, 2009
  – SOA (édition Dunod)
  – Essential Business Process Modeling (M/ Havey -O’Reilly)
  – SOA, le guide de l’architecte d’un SI agile, Dunod

• Sites:
  – http://www.soa-consortium.org/

• Les articles / Présos de Willy Goldgewitch, expert ESB - Valtech