

Gestión y organización



Sistemas de Información Orientados a Servicios

RODRIGO SANTAMARÍA

BPM

BPM y SOA

Herramientas: BPEL y
BPMN

Estrategias de diseño

Roles y organizaciones

Modelos de financiación

Gestión y organización

Gestión

3

BUSINESS PROCESS MANAGEMENT (BPM)

BPM Y SOA

BPEL

ESTRATEGIAS DE DISEÑO

ORQUESTACIÓN VS COREOGRAFÍA

Business Process Management (BPM)

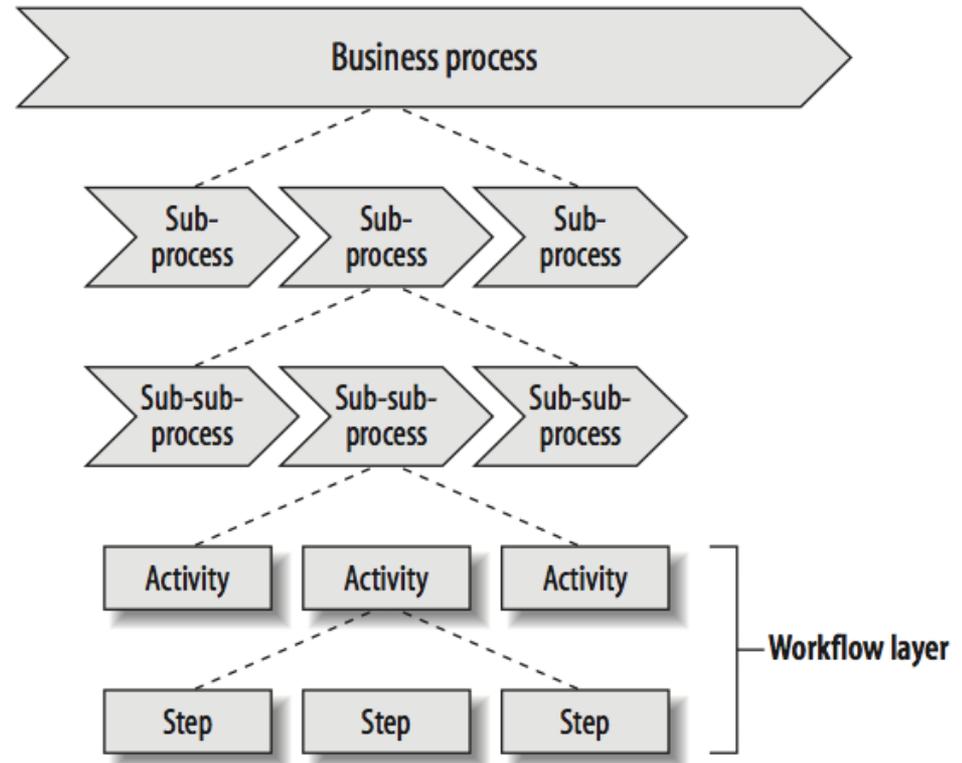
4

- **Management (gestión)**
 - Mejorar la visibilidad y control de los procesos de negocio en un amplio rango de sistemas, personas y organizaciones
- **Modeling (modelado)**
 - Descripción de los aspectos de los procesos de negocio: flujo, puntos de control, activadores, condiciones, contexto, etc.
- **Según estas definiciones, el modelado es parte de la gestión**
 - Por BPM, nos referiremos a gestión (management)

Business Process Management (BPM)

5

- **Proceso de negocio**
 - Descripción de *qué* hay que hacer, incluyendo E/S, posibles actividades manuales y cualquier tipo de recursos
- **Flujo de trabajo**
 - Descripción *cómo* hay que hacerlo, tratando en detalle los pasos o actividades



BPM y SOA



- Los servicios son aspectos de los procesos de negocio
- Sin embargo, BPM es un tópico muy amplio
 - Analizar el negocio (oferta, demanda, oportunidades)
 - Implementar e integrar estrategias de negocio
 - Monitorizar y optimizar los procesos de negocio
 - Etc.
- Aquí nos limitaremos a las siguientes preguntas
 - Cómo dividir una funcionalidad en sus partes (servicios)
 - Cómo diseñar esos servicios de manera que sean reutilizables

BPM y SOA

7

- Según Gartner (una de las mayores consultoras de EEUU) podemos considerar [Hill06]:
 - BPM → organizar *personas* para mejorar la agilidad
 - SOA → organizar *tecnologías* para mejorar la agilidad

De proceso de negocio a servicio

8

- Estrategia arriba-abajo
 - Peligro de ignorar aspectos técnicos → incremento de costes
- Estrategia abajo-arriba
 - Peligro de priorizar aspectos técnicos → no flexible, no intuitivo
- Estrategia 'ágil'
 - Mezcla de las dos anteriores

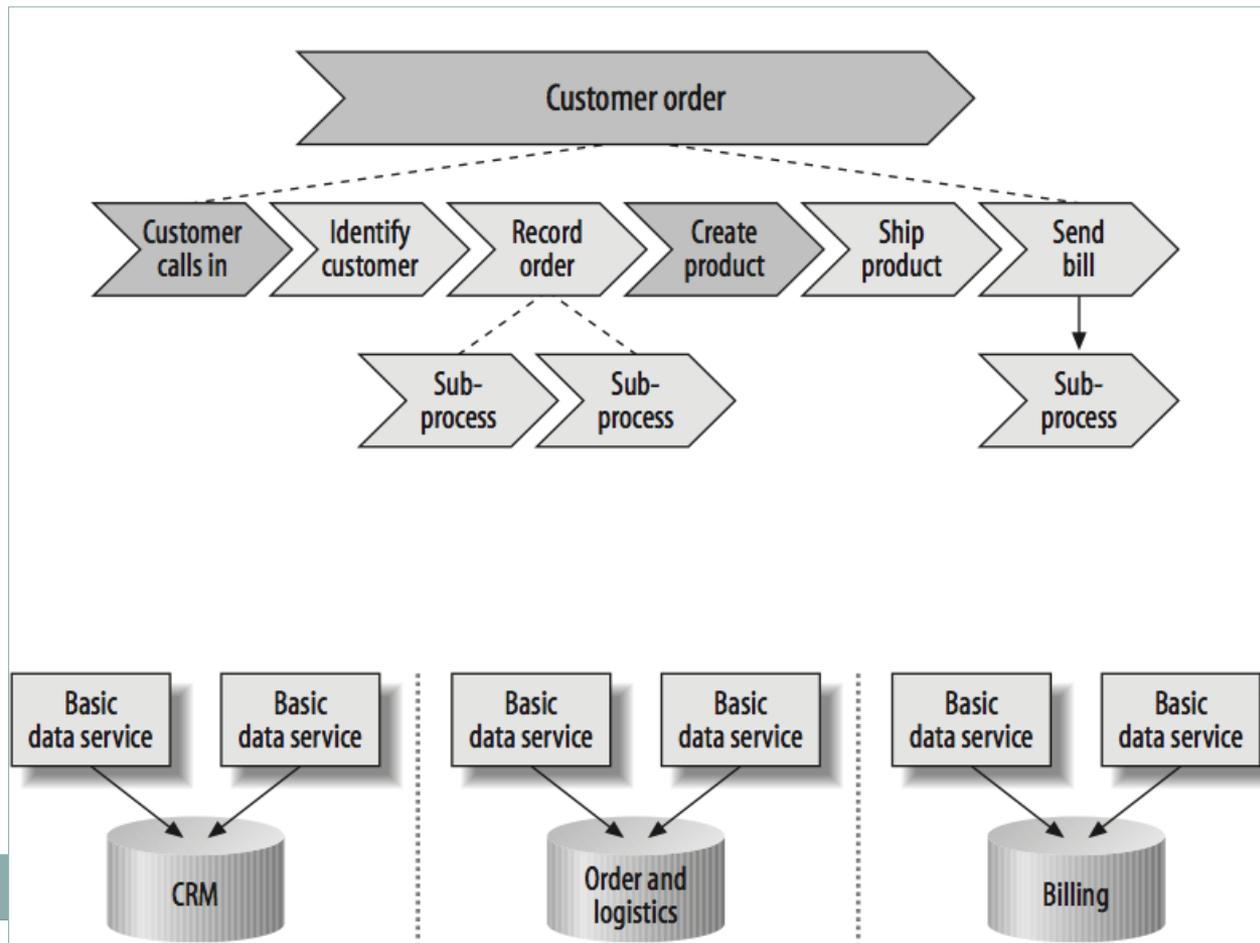
Your approach to SOA should be both top-down (through process decomposition) and bottom-up (exposing existing functionality as Services and composing them into processes). If you take only a top-down approach, you're likely to recommend building services that are technically difficult or complex to implement. Taking solely a bottom-up approach can yield services you don't need or, even worse, services that don't fulfill the business requirements.

[BloombergSchmelzer06]

Ejemplo

9

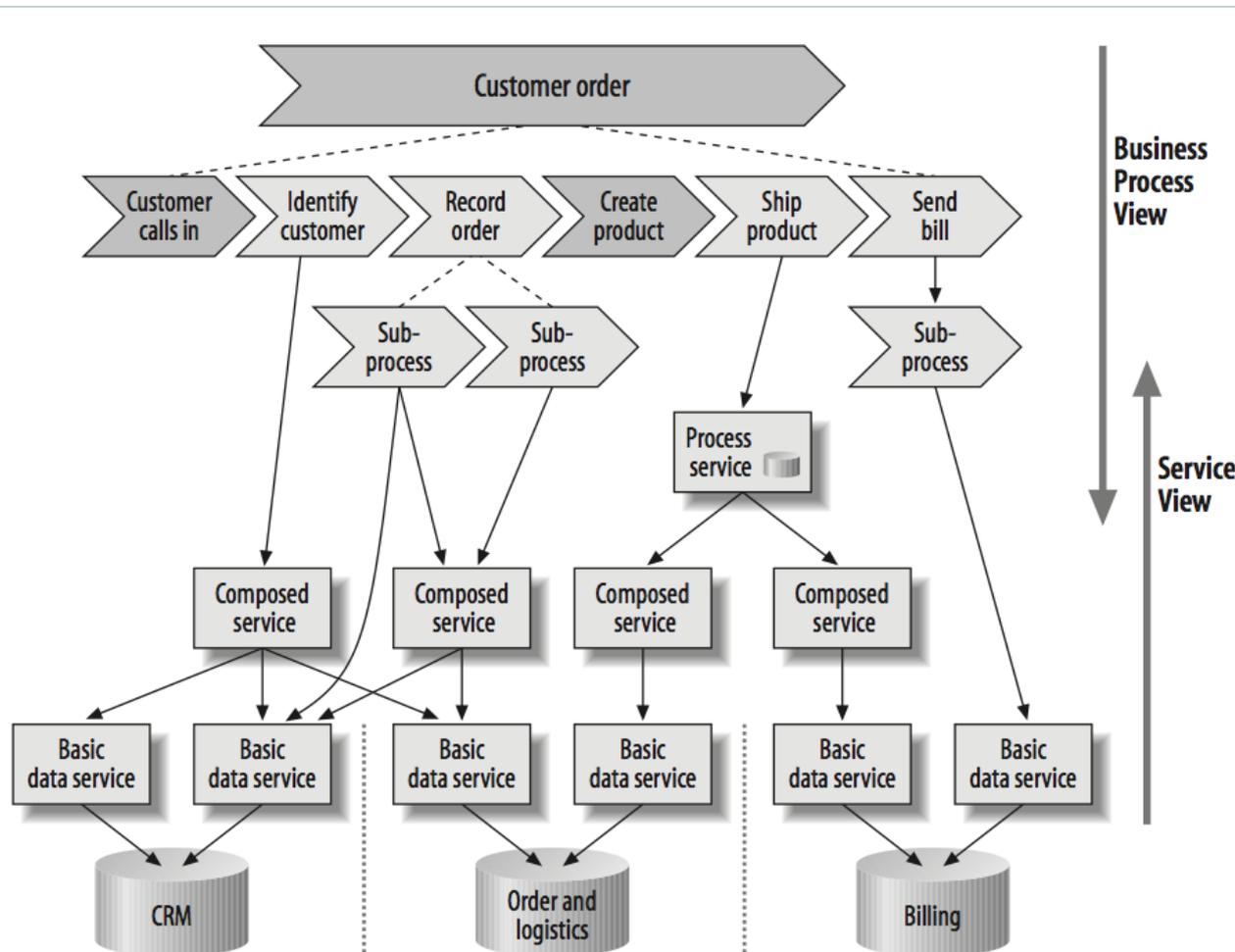
- Proceso: gestión de pedidos de clientes
 - Separar procesos en trozos más pequeños según *cuándo* y *dónde*



Ejemplo (II)

10

- Identificar qué subprocessos serán servicios



Qué es servicio, y de estos cuáles son básicos/compuestos dependerá de nuestras decisiones de diseño y del contexto

- *Existen ya servicios?*
- *Consideramos servicios técnicos?*
- *Reutilizables?*

¿Cómo sé que está bien hecho?

→ aprendizaje/experiencia

Empezar con una solución que parezca correcta y refinar

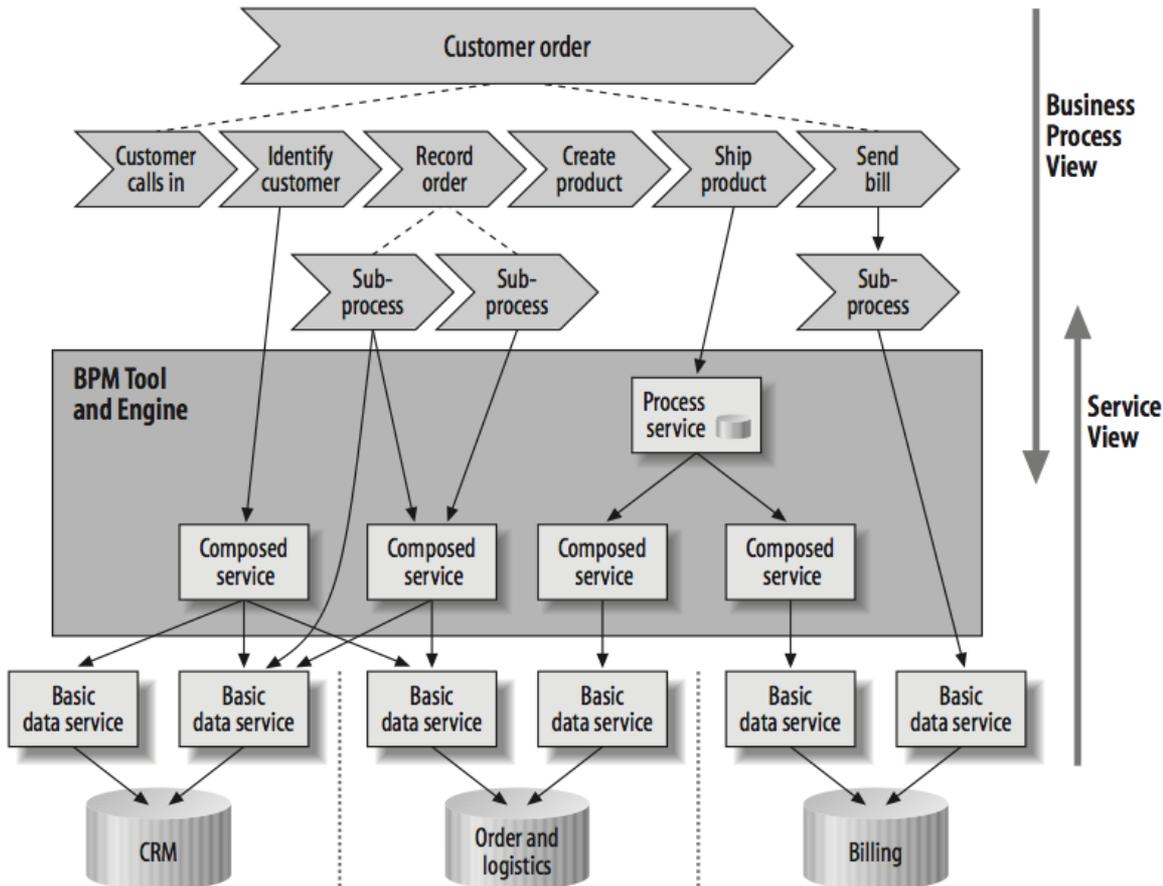
→ desarrollo iterativo

→ perfección = parálisis

Herramientas y motores BPM

11

- Apoyo informático a la gestión



1. Orquestación de servicios de proceso o servicios compuestos (alto nivel) a partir de servicios existentes
2. Ejecución y monitorización de dichos servicios

BPEL

12

- **Business Process Execution Language (BPEL)**
 - Estándar de definición de procesos basados en servicios
 - XML para definición de flujos de negocio
 - invocación de servicios
 - respuestas de proceso
 - variables de proceso
 - estructuras de control
 - tratamiento de errores y compensación
 - Desarrollado originalmente por IBM, ahora gestionado por OASIS (**O**pen **A**dvanced **S**tandards for **I**nformation **S**ociety)

BPEL (II)

13

```
<?xml version="1.0"?>
<process name="changeAddress" ...>
  <variables>
    <variable messageType="..." name="...">
  </variables>
  <flow>
    <receive .../> <!-- for this request (operation and input) -->
    <invoke .../> <!-- call other service -->
    <assign .../> <!-- map data -->
    <reply .../> <!-- return data -->
  </flow>
</process>
```

- **<process>** servicio web (en general, proceso) con distintas operaciones
- **<variables>** variables, manejadores de errores y enlaces a otros servicios
- **<flow>/<sequence>** una o más actividades a ejecutar **concurrente/secuencialmente**
 - **<receive>** actividad inicial, define el nombre de la operación y los datos de entrada
 - **<invoke>/<assign>** actividades que definen el proceso: invocación a otros servicios o mapeado/asignación de datos
 - **<reply>** actividad final normalmente, termina la operación y retorna datos de salida

BPEL (III)

14

- BPEL es legible pero tedioso de leer/escribir
 - Es fácil cometer errores
 - Normalmente se usa una herramienta
 - p.ej. BPEL Designer para Java/Eclipse
- Originalmente BPEL4WS (para servicios web)
 - Los 'procesos' que definimos se proveerán como servicios web
 - Aunque en teoría se pueden proveer como otros formatos
- Trata de ser un estándar automatizado, aunque hay otros más simples/menos automatizados:
 - BPMN (OMG)
 - Diagramas de actividad (UML)

BPEL: críticas

15

- Un analista utiliza constructos no estructurados
 - Básicamente, diagramas de cajas y flechas
- Un desarrollador utiliza constructos estructurados
 - Básicamente, lenguajes de programación
- Los BPEL se encuentran en una “tierra de nadie” intermedia [Vigneras08]
 - Lenguaje de Ejecución (EL)
 - Demasiado complejo y encorsetado para el analista
 - Demasiado genérico para el desarrollador
 - <http://www.infoq.com/articles/bpelbpm>

BPMN

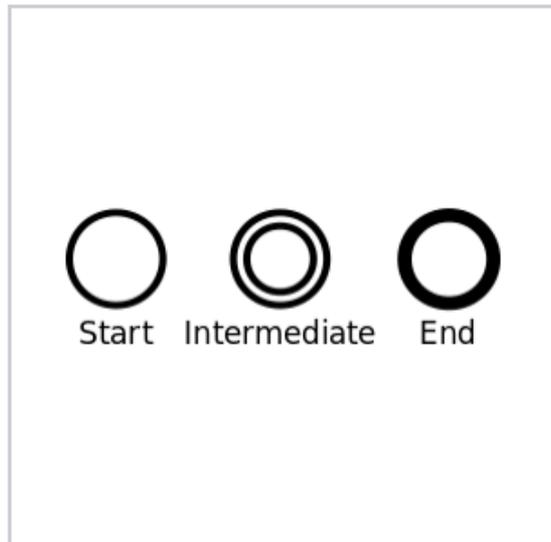
16

- La Notación BPM es una representación gráfica de procesos de negocio
 - Sin enlace al software que lo implementa
- Se utiliza en la industria actualmente*
- Múltiples herramientas:
 - <https://bpmtips.com/free-bpmn-modelling-tools-2020-edition/>
- Cuatro elementos:
 - Objetos
 - Conexiones
 - Pistas (Lanes)
 - Artefactos

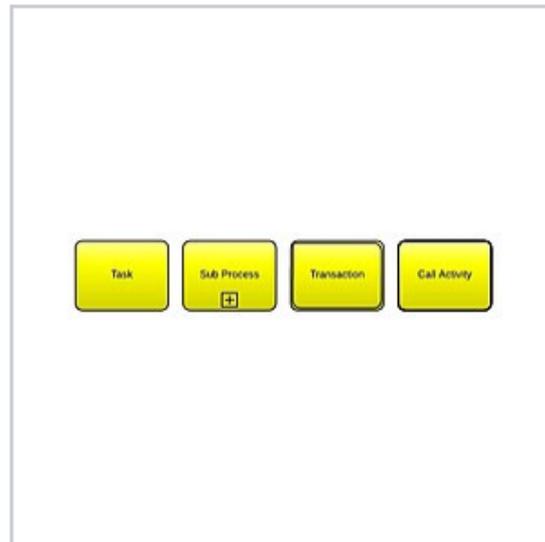
BPMN: Objetos

17

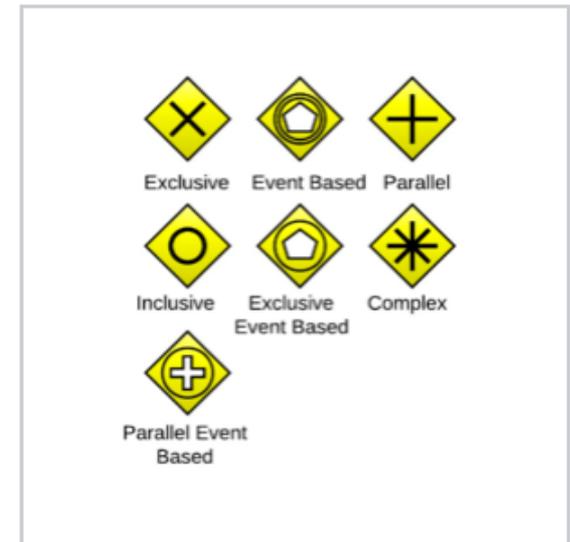
- **Evento:** algo que ocurre
 - Al inicio, final (resultado del flujo) o de manera intermedia
- **Actividad:** algo que se *hace* ~ servicios
 - Término genérico para un trabajo de la empresa
- **Puerta:** división o unión condicionada de rutas



Event



Activity



Gateway

BPMN: Puertas

18

- **Exclusiva:** redirige el flujo de manera única en función de una condición
- **Inclusiva:** redirige el flujo, que puede continuar por o venir de más de una ruta.
- **Paralela:** separa el flujo en varias líneas que se ejecutan todas de manera concurrente
- **Basada en eventos:** cualquiera de las anteriores, si es disparada por un evento



Exclusive



Inclusive



Parallel

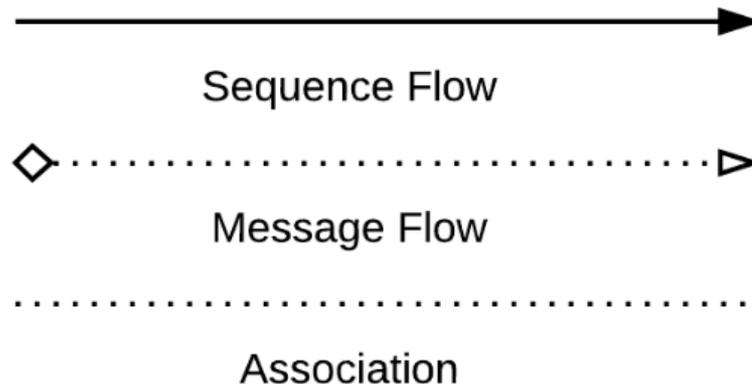


Event

BPMN: Conexiones

19

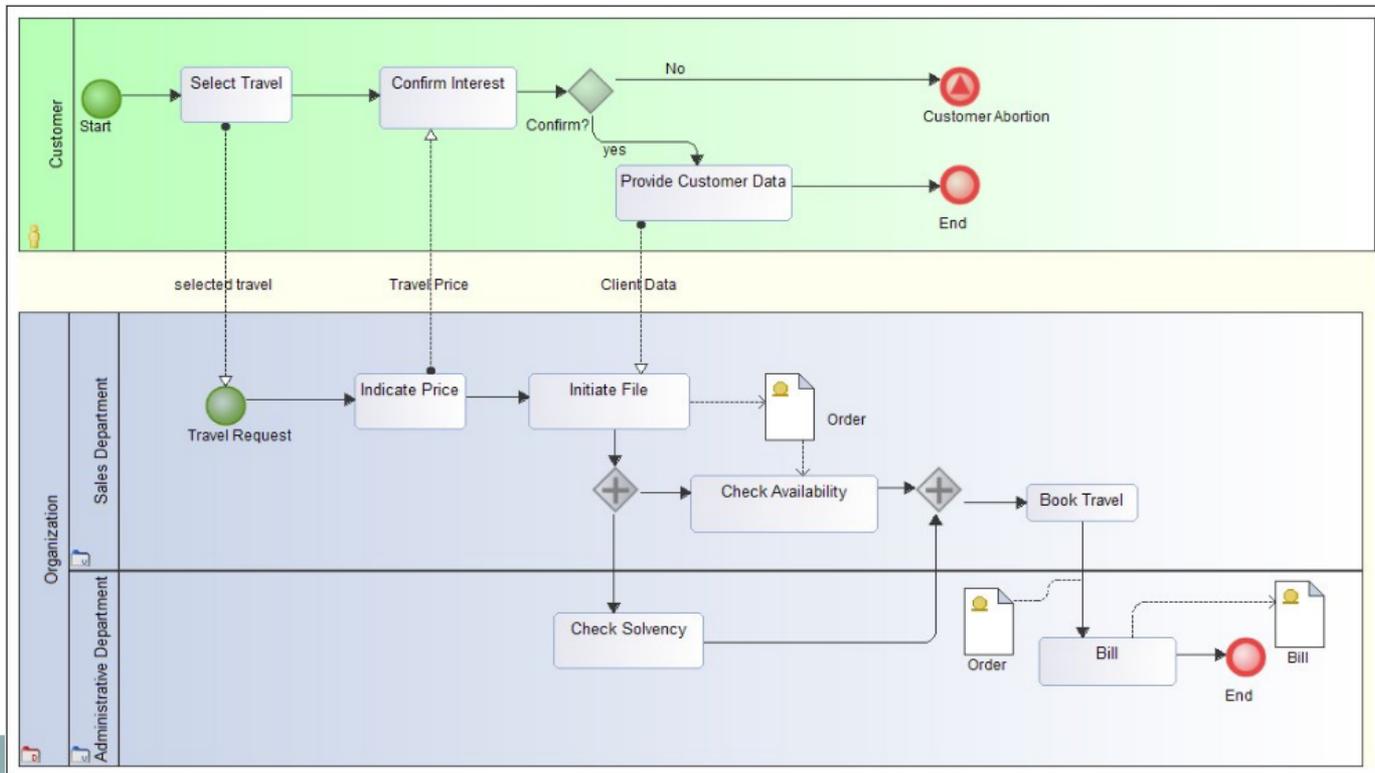
- Unen objetos de flujo de tres modos:
 - **Secuencia:** determina el orden de ejecución
 - **Mensaje:** indica el paso de información *entre organizaciones*
 - **Asociación:** unen objetos de flujo a otros componentes



BPMN: Pistas

20

- Representan áreas de influencia o responsabilidad.
- Metáfora de las piscinas
 - **Piscina (pool)**: cada organización
 - **Pista o carril (swimlane)**: cada departamento o rol en la organización



BPMN: Artefactos

21

- Entidades adicionales para hacer el modelo más legible:
 - **Datos:** requeridos o producidos por una actividad
 - **Grupos:** de actividades con algo en común
 - **Anotaciones:** cualquier otra información adicional



Data

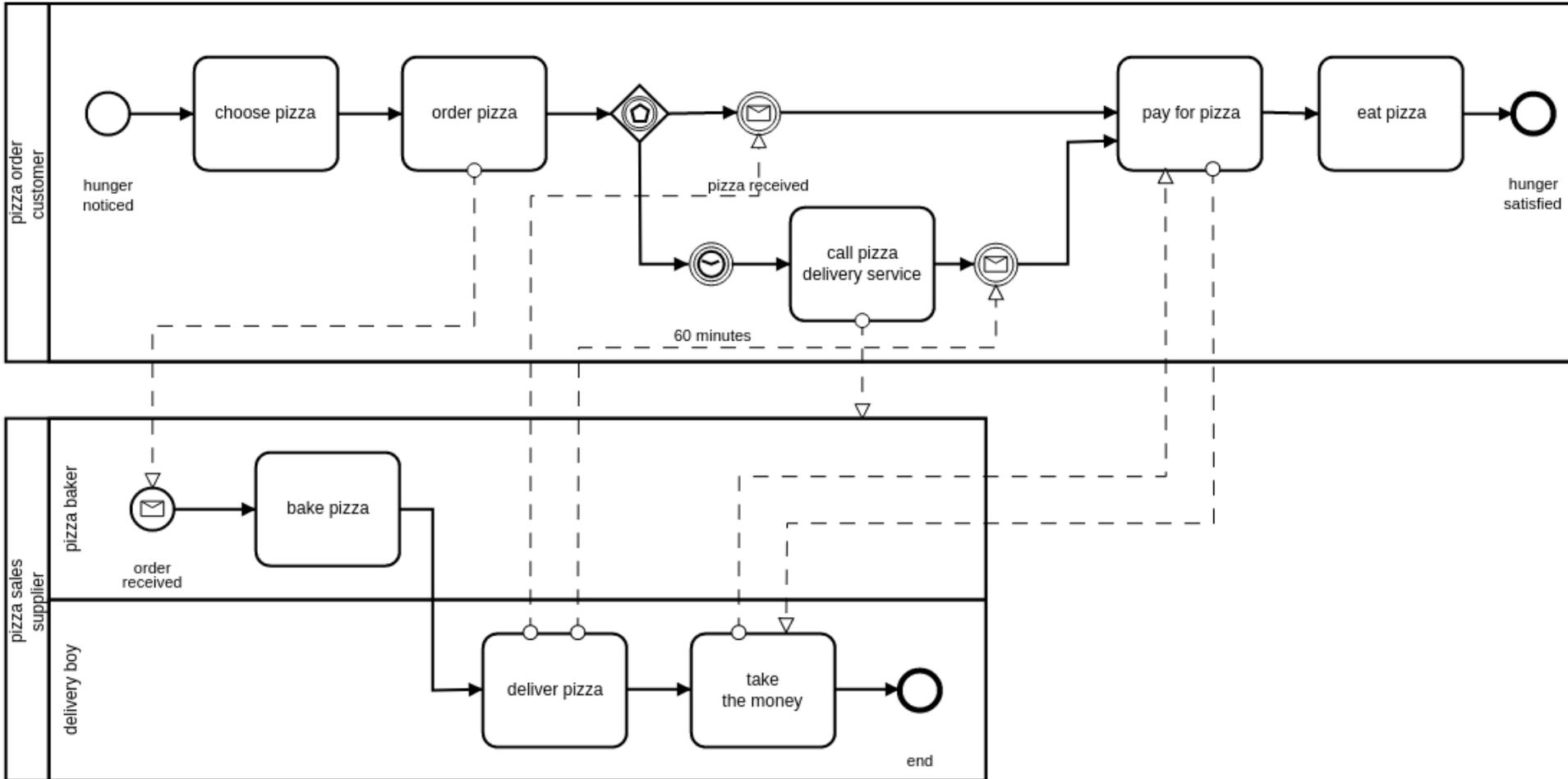


Group



Annotation

BPMN: Ejemplo



Estrategias de diseño

23

- **Gestión de portafolios**
 - Buscar cómo un servicio puede diseñarse para ajustarse a distintos requisitos → reutilización desde el principio
- **YAGNI (You ain't gonna need it)**
 - Programar lo estrictamente necesario → evita sobreesfuerzo
- **Usar el cerebro**
 - Entre las dos estrategias anteriores, decidir en función del contexto (tiempo, personal, sistema, etc.)

Estrategias de diseño: ejemplo

24

- Un cliente necesita un servicio que devuelva la dirección de un cliente (sólo calle y ciudad)
 - Parece sencillo retornar también el código postal
 - No requiere mucho sobreesfuerzo
 - Interesante si tenemos otros clientes que sí pueden pedírnoslo
 - Geolocalizar la dirección a través de una API externa
 - Puede ser difícil
 - Probablemente ningún cliente lo va a necesitar

Orquestación vs Coreografía

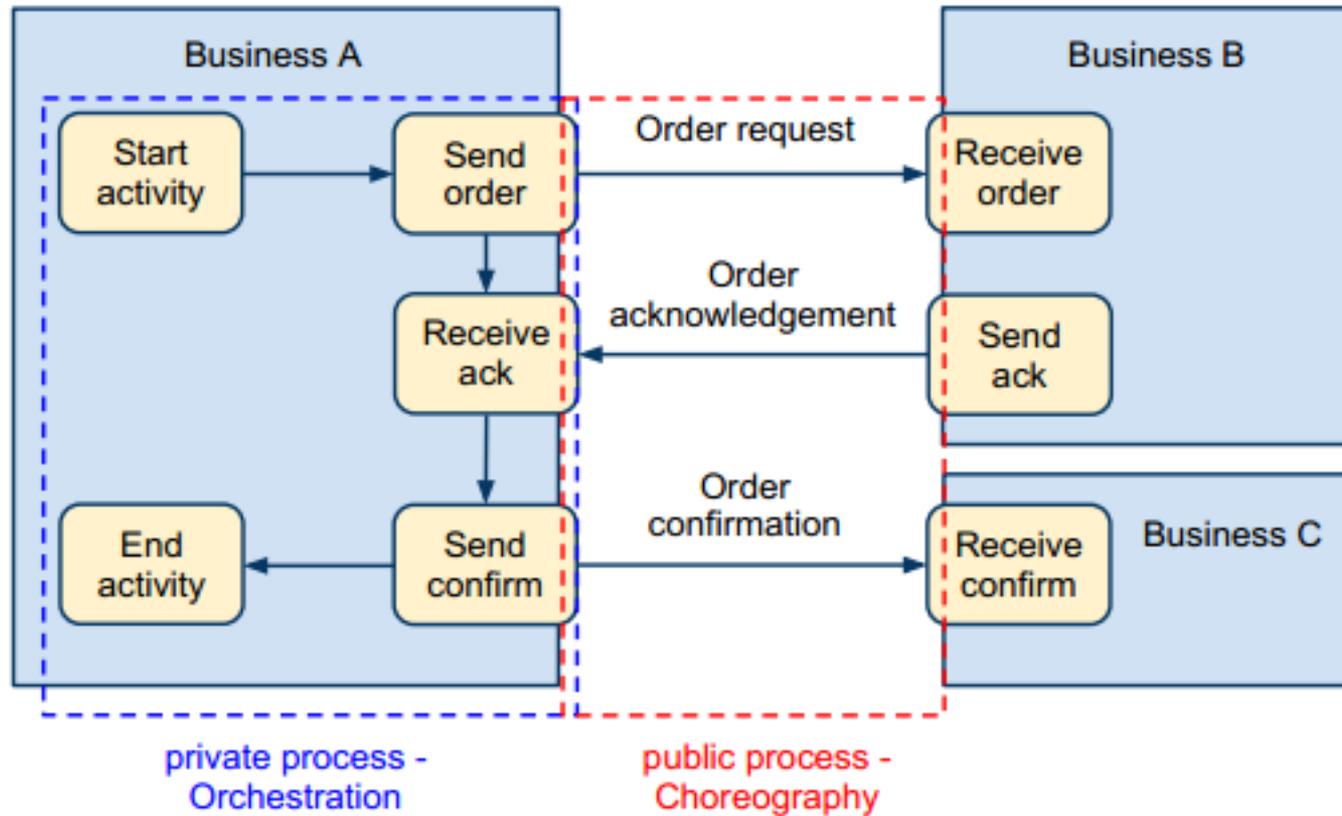
25

- Orquestación: diseñar servicios/procesos a partir de servicios existentes
 - Tenemos control central sobre el proceso
- Coreografía: colaboración entre distintas partes en el desarrollo del proceso
 - No tenemos el control o el conocimiento (!) total del proceso



Orquestación y Coreografía

26



Organización

27

**GESTIÓN DE LA SOLUCIÓN
COLABORACIÓN**

MODELOS DE FINANCIACIÓN

Organización

28

- SOA suele implicar una estrategia de negocio, además de tecnológica
 - Tiene un impacto en los sistemas de organización en general
- Los sistemas son distribuidos → las empresas no
 - Organización tradicional en departamentos
 - ‘Feudos’ buenos en lo suyo pero no acostumbrados a la cooperación con otros departamentos
 - Necesidad de nuevas estructuras como suplemento
 - **Gestión de la solución:** organización *‘orientada a procesos’* para nuevas funcionalidades
 - **Colaboración:** departamentos *inter-dominio* para mantenimiento

Gestión de la solución

29

- Añadir nueva funcionalidad puede tener impacto en varios departamentos (o compañías, equipos, etc.)
- Rol de ‘gestor del proyecto’
 - Diseño a alto nivel para ver el impacto en los sistemas existentes
 - Coordinar a los distintos dominios y niveles de dominio (departamento, compañía, equipo) responsables de los sistemas
 - Realizar estimaciones de coste a partir de la información experta de los dominios
 - Durante el desarrollo de la solución, estar alerta de cambios necesarios, errores, etc.
 - Cuando el proceso está terminado, si hay errores, trazar y determinar dónde ocurren

Colaboración

30

- Requisito clave de SOA
- Difícil entre departamentos aislados tipo *feudo*
- Necesidad de cambiar la cultura de la compañía
 - Delegar control en el gestor del proyecto
 - Asumir dependencias de su funcionalidad con otros dominios
 - Evitar el '*not invented here*' (desentenderse de la responsabilidad)
 - Eliminar la cultura de departamentos como 'centros rentables'
 - Buscan el beneficio propio por encima del sistema/compañía

*Any cultural model that has the potential to
undermine collaboration can jeopardize success*

Nicolai M. Josuttis

Colaboración (filosofía)

31

- Concepto de compañía como 'centro rentable'
 - Buscar el beneficio propio por encima de el del sistema/sociedad
- Tragedia de los comunes [Hardin68]
 - Un grupo de individuos explota un recurso común finito
 - El recurso termina destruido aunque esto no beneficie al grupo inicial ni al resto de la comunidad
 - Relación libertad - responsabilidad
 - Beneficio inmediato - beneficio sostenible
 - Interés egoísta/corto plazo - interés colectivo/largo plazo
 - Teoría de juegos: CC-PP (Commonize Costs - Privatize Profits)
 - The Evolution of Trust: <https://ncase.me/trust/>

SOLUTION MANAGEMENT BY EXAMPLE

One phone company I know of maintains most of its software in such a way that new releases get rolled out three or four times a year. Of course, this software has to match with existing marketing.

The process usually starts with marketing thinking about a new product or raising the need for a process optimization. A solution manager gets the task of making a high-level design and finding out how much the new feature or optimization will cost. Then, before the real development of a new release starts, the solution manager and the affected departments come together to make a release plan. The goal is to maximize the benefits for the company as a whole, at the lowest cost. The departments usually have budgets for each release, and the release plan should ideally use exactly 100 percent of each department's available resources. That is, the goal of the release planning is to give each team tasks that will occupy 100 percent of their development time, and to plan the tasks such that they will create the best possible benefit. Together, these multiple tasks will lead to the realization of complete solutions for the benefit of the company as a whole.

Of course, in practice things are a bit more complicated. For example, developers never have 100 percent of their time free for development (i.e., in agility terms, you need to know the “velocity” of the teams). In addition, you need time to verify that existing functionality is not broken. In fact, only the first half of the allotted development time is typically reserved for the development of new functionality. The second half is for integration and bug fixing. Last but not least, you must be aware that all your planning can become worthless if a competitor suddenly introduces a new feature. You’ll have to match this feature as soon as possible, which changes all existing plans.

However, experience shows that having the right project-oriented structure for such a case will help you to react fast. Distributed development also means distributed resources that help to develop solutions. If an organization has the appropriate structure, this can become a big market advantage.

Modelos de financiación

34

- El precio de implementar SOA y nuevos servicios sólo se ve recompensado cuando el sistema ‘escala’
 - P. ej. un nuevo servicio integrado y reutilizable no renta para la primera comunicación entre proveedor y consumidor, pero sí cuando tenemos múltiples proveedores y consumidores
- ¿Quién paga el desarrollo?
 1. El primer consumidor de una solución que requiera el servicio
 2. Todos los consumidores del nuevo servicio
 3. El proveedor, pues es una inversión
 4. Hay una bolsa de recursos para financiar nuevos servicios
- En la práctica: normalmente 1 ó 4

Resumen

36

- **BPM** nos permite identificar los servicios que forman parte de procesos de negocio distribuido
- En la práctica, se hace mediante una combinación de las aproximaciones **arriba/abajo** (descomponer procesos en trozos que darán lugar a servicios) y **abajo/arriba** (tener en cuenta los sistemas y servicios ya existentes)
- **BPEL** es el estándar más común para definición de 'procesos' que pueden convertirse en servicios compuestos o de proceso utilizando generalmente servicios web
- Componer servicios a partir de otros existentes se llama **orquestración** y es un modelo centralizado. Otra forma de organización es la **coreografía**, cuando los servicios no siempre están bajo control
- SOA afecta a la compañía u organización en su conjunto
- El procesamiento distribuido requiere distribuir la planificación, diseño, implementación, explotación y mantenimiento
- SOA requiere una combinación de estructuras orientadas a departamento y **orientadas a proyecto**
- El rol de un **gestor de proyectos** es coordinar los departamentos, crear el diseño a alto nivel y dar soporte durante la implementación y mantenimiento
- El modo de **financiación** suele ser cargar al primer cliente o tener un fondo de recursos

Referencias

- [Josuttis07] Nicolai M. Josuttis. *SOA in practice. The Art of Distributed System Design*. O'Reilly, **2007**. Ch 7/8.
- [BloombergSchmelzer06] Bloomberg, Jason and Ronald Schmelzer. *Service Orient of Be Doomed!: How Service Orientation Will Change Your Business*. John Wiley & Sons. **2006**
- [Hardin68] Garrett Hardin, *The Tragedy of the Commons*, Science, 162:3859, pp. 1243-1248. **1968**
- [Hill06] Hill J.B., Sinur J., Flint D. and Melenovsky M.J. *Gartner's position on business process management*. Business Issues, Gartner, Stamford, CT. **2006**
- [Pautasso09] Cesare Pautasso. *RESTful Web Service composition with BPEL for REST*. Data and Knowledge Engineering, 68, pp. 851-866. **2009**
 - <http://www.jopera.org/files/JDKE09-BPELforREST-Pautasso.pdf>
- [Vigneras08] Pierre Vigneras. *Why BPEL is not the holy grail for BPM*. InfoQ, **2008**
 - <http://www.infoq.com/articles/bpelbpm>

