

Introducción



Sistemas de Información Orientados a Servicios

RODRIGO SANTAMARÍA

Organización
distribuida

SD a gran escala

Un poco de historia

Definiciones

Introducción

Organización distribuida

3

- n sistemas individuales \rightarrow 1 sistema distribuido
 - Flexibilidad
 - Mantenimiento
 - Escala
 - ▮ Los sistemas centralizados no son fácilmente escalables
 - ▮ Nos movemos hacia sistemas descentralizados
 - ▮ $m < n^*$ sistemas individuales pueden modificarse o caerse sin afectar al sistema distribuido

Organización distribuida

4

- Evolución del concepto de organización
 - En computación
 - En neurología
 - En economía
 - En política

Cerebro como concepto distribuido

5

“There is no central processing unit in the brain; rather the whole functions as a kind of *distributed* computer. Also, there is no central conductor coordinating all of the distributed parts. Each component coordinates interactions with the other components to which it is connected”

Colin Ware. *Visual thinking for design*

Cerebro como concepto distribuido



- Hay neuronas por todo el cuerpo, algunas, como las del sistema digestivo, modulan muchas emociones*.
- El 90% de las fibras del nervio vago (el nervio visceral más importante) llevan información al cerebro, y no al revés*.

Economía como concepto distribuido

7

- Blabla car (transporte P2P)
- Airbnb (alojamiento P2P)
- Comunitae, Zopa (préstamos P2P)
- Kickstarter (financiación P2P)
- ¿Conoces algún otro ejemplo?

Política como concepto distribuido

8

- Evolución de sistemas centralizados (absolutismo, dictadura) a sistemas cada vez más descentralizados (democracias representativa, participativa, directa)
- Nuevos
 - Movimientos: 15M, Democracia 2.0
 - Modelos: Voto preferencial¹
 - Herramientas: Doodle, Kuorum, Appgree²

¹ <http://www.resilience.org/stories/2014-01-08/majority-voting-is-inadequate>

² <http://vimeo.com/80561797>

Computación como concepto distribuido

9

- **Evolución de los sistemas distribuidos (SD)**
 - Soluciones para sincronización y coordinación.
 - Tiempos lógicos
 - Algoritmos distribuidos
 - Soluciones técnicas a problemas de heterogeneidad y comunicación
 - Middleware
 - Servicios web

Service-Oriented Architecture (SOA)

10

- Arquitectura centrada en servicios discretos en vez de un diseño monolítico.
- Es una aproximación distribuida que mantiene el sistema construido sobre ella **escalable** y **flexible**
- Tres conceptos principales
 - **Servicios** que representan funcionalidades independientes, combinables e implementables por cualquier tecnología sobre cualquier plataforma
 - **Enterprise Service Bus (ESB)**: Infraestructura para la combinación y distribución de servicios.
 - **Acoplamiento bajo (loose coupling)**: dependencias reducidas dentro del sistema

SOA vs Servicio Web

11

- Un servicio web *no* es SOA
 - Sería simplemente la infraestructura sobre la que se puede apoyar la arquitectura SOA

Although Web Services do not necessarily translate to SOA, and not all SOA is based on Web Services, the relationship between the two technology directions is important and they are mutually influential: Web Services momentum will bring SOA to mainstream users, and the best-practice architecture of SOA will help make Web Services initiatives successful.

Gartner, 2003. Service-Oriented Architecture Scenario.
<https://www.gartner.com/doc/391595/serviceoriented-architecture-scenario>

SOA vs SD

12

- Un sistema distribuido *no* es SOA
 - SOA es un paradigma aplicable para la construcción de un SD
 - Especialmente recomendado para un SD a gran escala
 - ¿Qué SD conoces que no sea SOA?

Características de un SD a gran escala

13

- **Legado:** ciclo de vida prolongado
 - Mantenimiento como característica importante
- **Heterogeneidad** en distintos aspectos:
 - Objetivos
 - Tiempos de implementación
 - Lenguajes/paradigmas de programación
 - Plataformas/middleware
- **Complejidad:** no vale un 'arreglo rápido'
 - Cualquier cambio en un componente puede afectar a otros componentes

Características de un SD a gran escala (II)

14

- **Propiedad:** distintos propietarios
 - Lo que implica distintos equipos, presupuestos, agendas, etc.
- **Imperfección:** el perfeccionismo es demasiado caro
 - Perfeccionismo=parálisis
 - La mayoría de los sistemas tienen un 99% de corrección, un 100% requeriría demasiado esfuerzo/coste
- **Redundancia:** accidental o 'controlada'
 - En la práctica, tener todos los datos normalizados es difícil y puede penalizar el rendimiento considerablemente.

Historia

15

- A. Pasik, ingeniero en la consultora Gartner, acuñó el término SOA en 1994
 - Antes de que se generalizara XML o los servicios web
 - Razón: ‘cliente/servidor’ había perdido su sentido
 - El ‘cliente’ estaba ejecutando la representación y la mayoría de la lógica de negocio
 - El ‘servidor’ ejecutaba el manejo de la base de datos y parte de la lógica de negocio, y almacenaba los datos
 - Es decir, ‘cliente’ y ‘servidor’ sólo definían el hardware, pues el software estaba repartido entre ambos.
 - Para evitar la confusión con el viejo concepto de cliente/servidor, Pasik llama al nuevo modelo ‘orientado a servicios’.

Historia (II)

16

- SOA toma masa crítica con los Servicios Web (2000)
- Un servicio web no necesariamente cumple con SOA
- No todo SOA está basado en servicios web
- Pero ambos se influyen mutuamente
 - Los servicios web llevan el concepto de SOA a más usuarios
 - SOA ayuda a realizar servicios web más exitosos

Historia (III)

17

- Para 2005, SOA se ha convertido de alguna manera en un icono poco cuestionado

*There is way too much **hype** about it. The idea of services is not a means of abstraction. It is simply a mechanism for reaching into systems.*

*You see organizations rushing to [implement] services, but they are really **missing the fundamental engineering principles.***

In about 18 months, they will complain SOA doesn't work. They'll be blaming the wrong thing. They should be blaming their architectures and best practices.

Grady Booch, 2006¹

- ¿Se te ocurren otras críticas al modelo?

¹ <http://gcn.com/Articles/2006/07/12/Grady-Booch--Avoid-the-stupid-SOA-approach.aspx?Page=1>

Historia (IV)

18

- En 2011, se introduce el término **microservicio***
 - SOA de 'grano fino'
 - Llevado a la escala web por Netflix
 - Tiene también sus críticas
 - No entraremos en mucho detalle de esta arquitectura, para más información:
 - Martin Fowler, **2014**. *Microservices*.
<https://martinfowler.com/articles/microservices.html>
 - Repositorio de conocimiento:
 - <https://microservices.io>

Definición

19

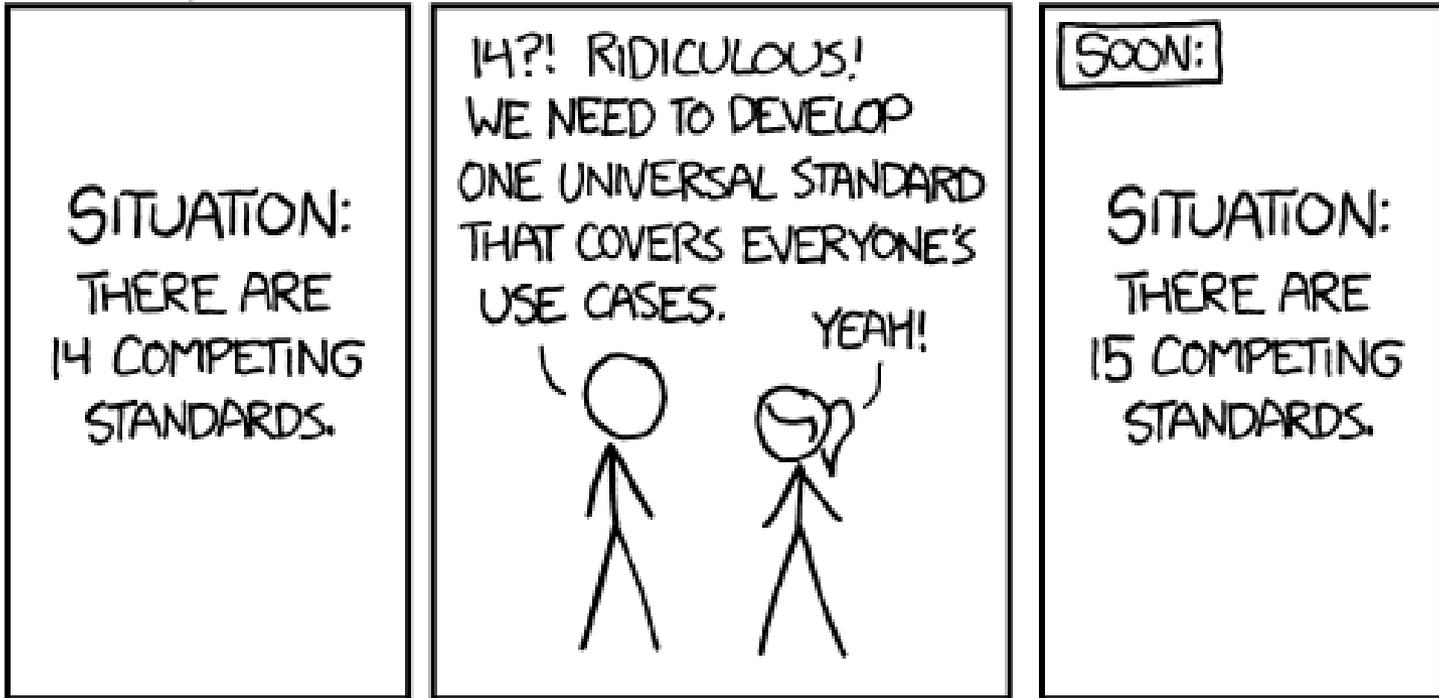
- SOA es un **paradigma** (o concepto/filosofía) que busca mejorar la **flexibilidad** de un sistema
- Las definiciones varían pero estos dos aspectos (paradigma/flexibilidad) siempre están presentes
- Esta definición es muy vaga, se complementa con los distintos **actores** a los que va dirigida, los **roles** que toman y los **conceptos** técnicos asociados

Actores

20

- Actuamos en **sistemas distribuidos complejos**
 - Localización y uso de recursos distribuidos
- Con **distintos propietarios**
 - Dichos SD pueden estar bajo distintos dominios de propiedad
 - Este es un punto importante que suele ignorarse y que es una de las razones por las que SOA no es sólo un concepto técnico
- Y **heterogéneos**
 - La heterogeneidad es intrínseca a todo SD complejo moderno
 - En vez de luchar contra ella (estándares), SOA la acepta como algo inherente e inevitable

HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)

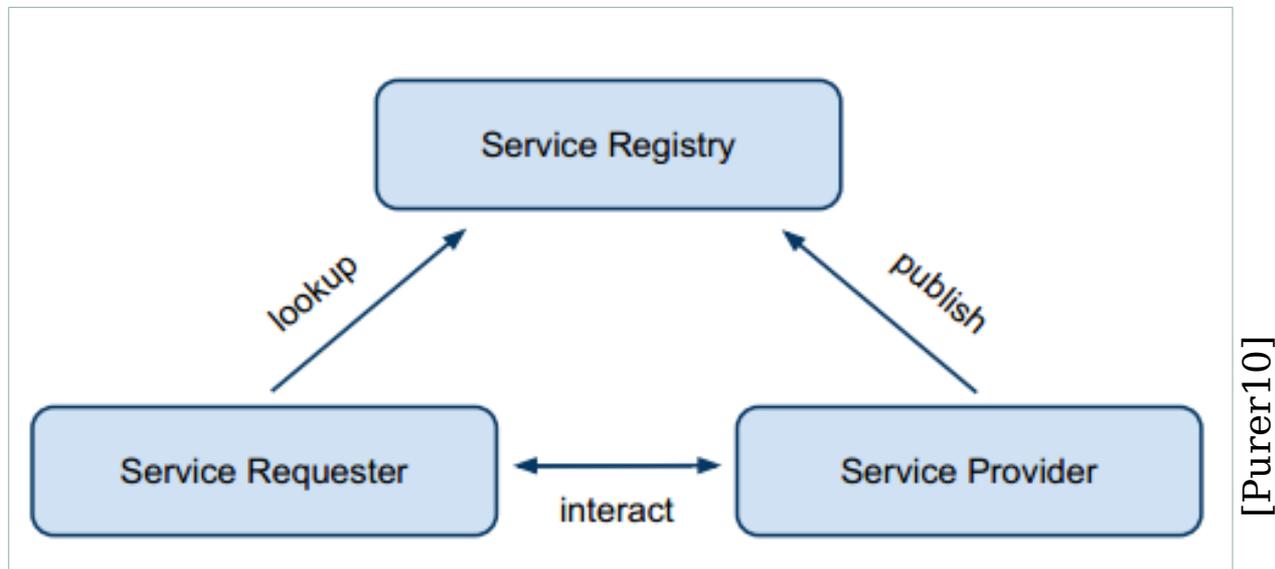


<https://xkcd.com/927/>

Roles

22

- **Proveedor**: sistema que implementa un servicio
- **Consumidor**: sistema que invoca un servicio
- **Participante**: un proveedor o consumidor
 - A veces también se identifica un *registro* como el participante que publica el servicio [Purer10]



Conceptos técnicos

23

- Hay tres conceptos técnicos con los que trabaja SOA
 - Servicio → funcionalidades
 - Interoperabilidad → con buena conectividad
 - Acoplamiento → pero baja dependencia

Servicios

24

- Representación informática de una funcionalidad
 - Suficientemente abstracta para ser comprendida por el cliente y para ignorar detalles técnicos como la plataforma
- Más allá de esta definición, hay poco consenso, aunque cualquier funcionalidad independiente de un negocio puede considerarse un servicio
 - “Crear un cliente”
 - “Transferir dinero”
 - “Encender la radio”
 - “Calcular la mejor ruta para un viaje”

Interoperabilidad y acoplamiento

25

- **Alta interoperabilidad**
 - Capacidad para conectar sistemas fácilmente
- **Bajo acoplamiento***
 - Capacidad para minimizar dependencias entre sistemas
 - Ayuda a mejorar la tolerancia a fallos y la flexibilidad
 - Nos dirige hacia la escalabilidad
 - Un modo de 'bajar' el acoplamiento es la descentralización

* En inglés *loose coupling*. *Loose* literalmente significa 'flojo, suelto'

Ingredientes

26

- Los conceptos anteriores no se pueden ‘comprar’ o implementar directamente
- Es necesario hacer el trabajo duro de ocuparse de
 - Infraestructura
 - Arquitectura
 - Procesos
 - Gobierno (metaprosesos)

Infraestructura

27

- Parte técnica de SOA que permite la alta interoperabilidad
- Se conoce como Enterprise Service Bus (*ESB*)
- Responsabilidades
 - Representación de datos
 - Enrutamiento
 - Seguridad y fiabilidad
 - Monitorización y autenticación

Arquitectura

28

- La A en SOA debe ser definida de manera específica para cada sistema
 - Tipos de servicios
 - Grado de desacoplamiento
 - Interfaces de servicios
 - Tecnología para la infraestructura
 - Estándares utilizados

Procesos

29

- **Business process modeling (BPM)**
 - Proceso de división las tareas de negocio en actividades independientes/atómicas, que serán los servicios
- **Ciclos de vida de los servicios**
 - Proceso de definición de los distintos pasos que llevan a la implementación de un servicio
- **Model-driven software development (MDSD)**
 - Proceso de generación de código para implementar servicios
- **Gobierno: metaproceso de control de procesos y estrategia SOA**
 - Buscar el grupo de personas que reúna los ingredientes SOA
 - A este equipo se le conoce como SOA competence center

SOA

sólo es un paradigma

alta interoperabilidad

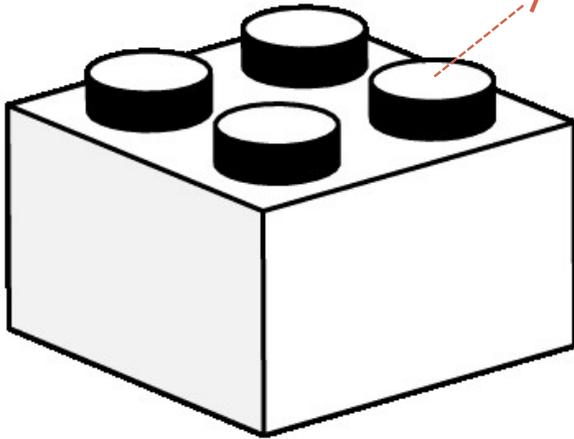
capacidad para conectar sistemas fácilmente

bajo acoplamiento

minimiza la dependencia entre sistemas

servicios

que se ofertan



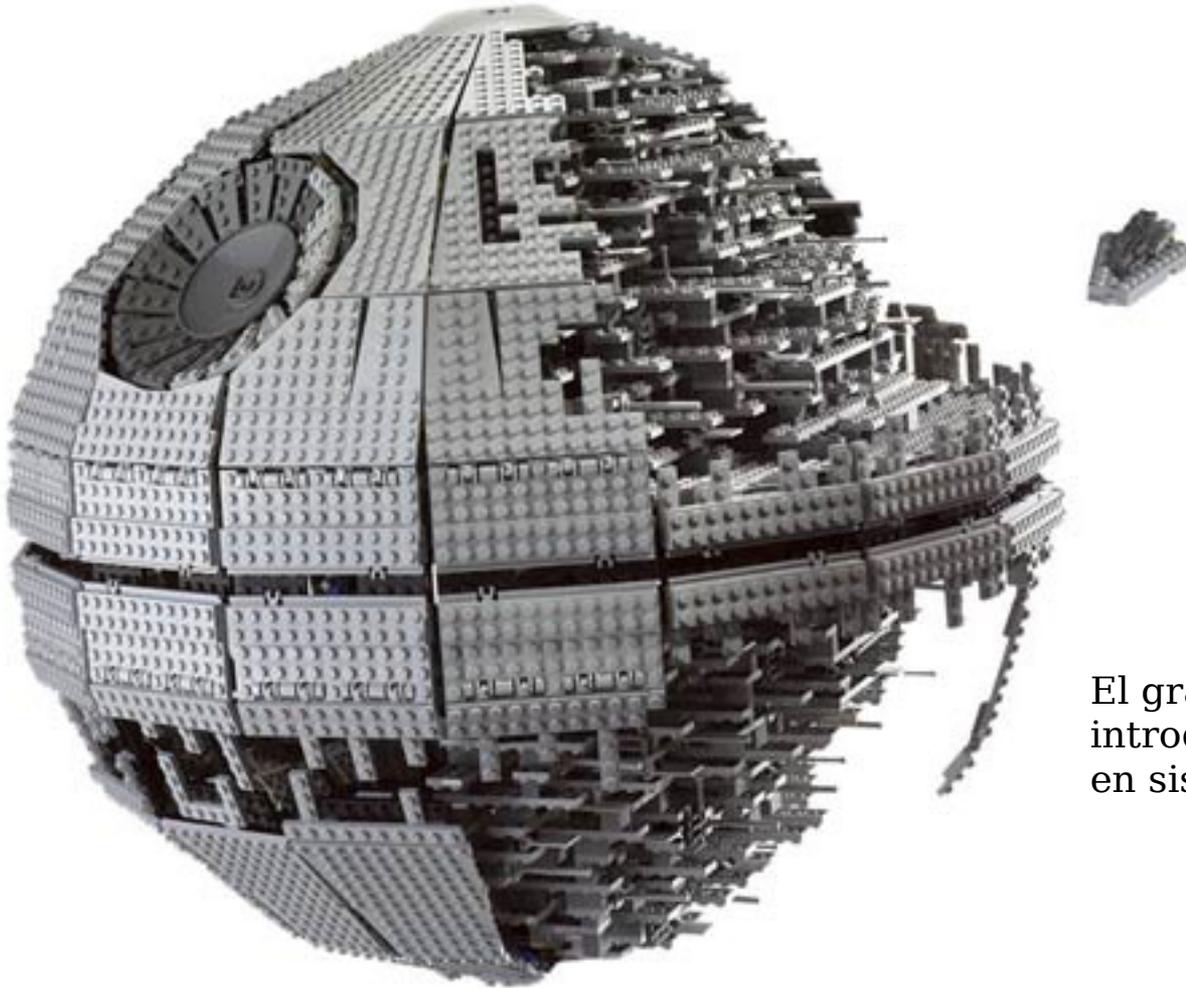
procesos, arquitectura e infraestructura

de implantación mediante distintas tecnologías

ESB

técnica para garantizar la interoperabilidad





El grado de complejidad que introduce SOA sólo se recomienda en sistemas:

- distribuidos
- de gran escala
- con distintos propietarios

En estos casos, SOA garantizará la escala, así como la flexibilidad y mantenimiento del sistema

SOA no es

32

- No es una solución para todo
 - Su importancia se ha sobreestimado y está de moda
 - Implantarlo requiere esfuerzo
 - Sólo está indicado en circunstancias muy especiales: sistemas distribuidos heterogéneos a gran escala con distintos propietarios
- No es un modelo de objetos distribuidos (e.g. CORBA)
 - No ofrece acceso remoto a objetos en sistemas externos
 - No requieren un modelo de objetos general en todo el sistema
 - Lo cual no escala demasiado bien
- No es una tecnología específica
 - No es un servicio web, sino un paradigma
 - Los servicios web se están convirtiendo en el estándar *de facto* para implementar SOA, pero no es el único modo

The SOA Holy Grail

Anyone working on enterprise systems in the last 10 years will remember the initial tenets of Service Orientated Architecture were to **decouple applications** and to **provide a well defined service interface**, which can be reused by applications and composed into business processes. The idea of reuse and composition made SOA an attractive proposition that sent thousands of organizations on a very challenging treasure hunt. We have since read SOA's obituary and its resurrection with many stories of woe peppered with some success, but **with very few achieving the holy grail** of SOA.

Meanwhile, the web has essentially become a service oriented platform, where information and functionality is available through an API; **the Web succeeded where the enterprise largely failed.**

This success can be attributed to the fact that the web has been **decentralized in its approach** and has adopted **less stringent technologies** to become service oriented.

[Mason11]

Resumen

34

- SOA es un **paradigma** para tratar con sistemas distribuidos a **gran escala** y bajo el control de **distintos propietarios**
- Los conceptos o principios clave son: **servicios, alta interoperabilidad, bajo acoplamiento**
- Los ingredientes clave son: **infraestructura, arquitectura y procesos** (incluyendo *gobierno*)
- Los factores clave para el éxito de una implantación SOA son: comprensión, gobierno, apoyo a la gestión y trabajo
- SOA **no es** una tecnología específica ni un comodín, no siempre es apropiada
- Los **servicios web** son un modo de implantar SOA, no el único pero sí uno que se está convirtiendo en estándar *de facto*.

Referencias

- [Josuttis07] Nicolai M. Josuttis. *SOA in practice. The Art of Distributed System Design*. O'Reilly, **2007**. Ch 1/2.
- [Purer10] Klaus Purer. *Web service composition in Drupal*. Universidad de Viena, **2010**
 - <http://klau.si/sites/default/files/thesis-klaus.pdf>
- [Booch06] Grady Booch. *Avoid the 'stupid' SOA approach*. GCN, **2006**.
 - <http://gcn.com/Articles/2006/07/12/Grady-Booch--Avoid-the-stupid-SOA-approach.aspx?Page=1>
- [OASIS07] OASIS. *Reference model for service oriented architectures*. Committee Draft 1.0, **2007**
 - <http://xml.coverpages.org/SOA-RM-ReferenceModel200602-CD.pdf>
- [Mason11] Ross Mason. *How REST replaced SOAP on the web: what it means to you*. InfoQ, **2011**
 - <http://www.infoq.com/articles/rest-soap>

