

Computação Orientada a Serviços

Evandro César Freiburger

6 de fevereiro de 2023

Agenda

- 1 Introdução
- 2 Computação Orientada a Serviços - Conceitos
- 3 Princípios de COS
- 4 Questões Tecnológicas para COS
- 5 Composição de Serviços
- 6 Considerações
- 7 Referências

Introdução

- As organizações sociais e econômicas, estão **baseadas em trocas de tarefas e produtos**;
- Cada pessoa ou grupo de pessoas se **especializa na execução de tarefas** e trocam os resultados umas com as outras;
- Quando as tarefas são bem definidas e organizadas **podem ser chamadas de serviços**;



Introdução

No modelo de computação predominante,

- Cada empresa, organização ou pessoa **possui seu próprio parque computacional**.
- A aquisição, administração e manutenção do sistema computacional são de **responsabilidade do proprietário**.



Introdução

Esse modelo,

- Exige que a organização gaste esforços em questões não vinculadas ao seu negócio, tais como:
 - Aquisição
 - Administração
 - Manutenção de equipamentos
 - Equipe especializada

Duas mudanças de paradigma de computação podem ser destacadas:

- Modelo de desenvolvimento - Computação Orientada a Serviços
- Infraestrutura de ambiente de execução - Computação em Nuvem

Introdução

Em 1969, Leonard Kleinrock disse:

A partir de agora, redes de computadores ainda estão na sua infância, mas à medida que crescem e se tornam sofisticadas, vamos provavelmente ver a propagação da computação utilitária que, como energia elétrica e telefone, estarão presentes nas residências e escritórios de todos os países.

Introdução

A computação deve desenvolver métodos e tecnologias que dêem suporte à automação de processos de negócio.

Um exemplo: Modelo de Desenvolvimento Orientado a Objetos.

Recentemente outro modelo de desenvolvimento ganhou destaque - Computação Orientada a Serviços (COS ou SOC em Inglês).

Objetivo é aproximar os processos de negócio dos recursos de tecnologia da informação.

Introdução

A infraestrutura de computação também passou por grandes transformações:

- 1 Computadores que centralizavam o processamento (*mainframe*)
- 2 Disseminação dos mini e microcomputadores promoveu a descentralização da computação
- 3 Computação distribuída torna-se realidade;
- 4 Tecnologias como CORBA, DCOM, RMI e Serviços Web, possibilitaram a distribuição de elementos de software pela rede.

Introdução

Pode-se observar **mudanças de paradigma da computação distribuída** [Joseph et al. 2004], tais como:

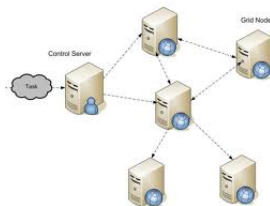
- Foco em orientação a serviços
- Integração por meio de padrões abertos
- Colaboração e virtualização



Introdução

A necessidade de **alto desempenho e disponibilidade**, fez surgir novas arquiteturas computacionais [Zhang et al. 2010], tais como:

- Computação em *Clusters*
- Computação em Grids
- Computação em Nuvem



Introdução

A Computação em Nuvem (*Cloud Computing*) é a mais recente revolução do paradigma da computação distribuída.

Contratação de **serviços sob demanda** e compartilhamento de recursos com maior nível de **flexibilidade e escalabilidade**.



Computação Orientada a Serviços

COS é um **paradigma**

- Utiliza **serviços como elemento básico**
- Apoia o desenvolvimento **rápido**, de **baixo custo** e de fácil **composição de aplicações distribuídas**
- Proporciona a cooperação entre serviços, onde os **componentes de aplicação são montados**
- Automatiza processos de negócio em diferentes organizações e plataformas de computação com flexibilidade e agilidade

[Papazoglou et al. 2008].

Computação Orientada a Serviços

Pode ser entendida como a junção de vários elementos, são eles:

Orientação a Serviços

- É um **paradigma de projeto** composto por um conjunto específico de princípios
- Descreve a forma de **decomposição e representação** de um domínio de problema
- A aplicação desses princípios define a **Lógica Orientada a Serviços**

[ERL 2009]

Computação Orientada a Serviços

Serviço

- É a unidade fundamental da lógica orientada a serviços.
- Como elemento físico, são programas de software fisicamente independentes.
- Cada serviço possui um conjunto de capacidades associadas a esse contexto.
- As capacidades são expostas como contratos de serviços, que estabelecem políticas de uso, formatos de chamada e resultados.

[ERL 2009]

Computação Orientada a Serviços

Arquitetura Orientada a Serviços (SOA)

- 1 Modelo arquitetural conceitual de computação orientada a serviço (por exemplo, modelos de referência de SOA).
- 2 Arquitetura tecnológica que estabelece uma combinação de tecnologias, produtos, API e infraestruturas de suporte.

Uma arquitetura implantada em uma organização é **particular a ela**, dando suporte a criação, execução e evolução de soluções orientadas a serviços daquela organização.

[ERL 2009]

Computação Orientada a Serviços

Composição de Serviços

- É a capacidade de produzir novos serviços a partir de outros serviços já existentes de maneira coordenada.

Inventário de Serviços

- É uma coleção padronizada e governada de serviços
- Uma organização pode conter diversos inventários de serviços
- Cada inventário pode ter sua própria padronização e governança apoiada por uma arquitetura orientada a serviços

[ERL 2009]

Computação Orientada a Serviços

Assim, COS é constituída de:

- uma arquitetura orientada a serviços que **representa uma instância tecnológica**;
- projetada para **dar suporte à lógica orientada a serviços**;
- que contém **serviços e composições de serviços catalogados e governados** por um inventário de serviços;
- em conformidade com o **paradigma da orientação a serviços**.

[ERL 2009]

Classificação de Serviço

Segundo Erl em [ERL 2009], é possível estabelecer uma classificação de serviços em três níveis: serviço de entidade, serviço-tarefa e serviço utilitário.

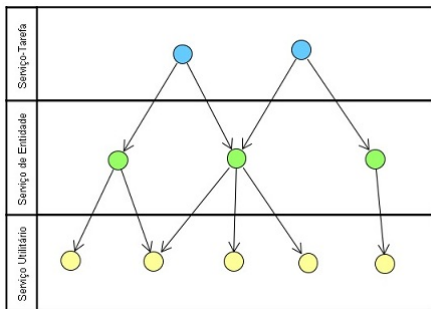


Figura: Classificação de Serviços, adaptado de [ERL 2009]

Classificação de Serviço

Serviço Utilitário

- não representam contextos funcionais de negócio e sim serviços voltados a questões técnicas de implementação.
- são projetados para proporcionarem reuso de serviços tais como: registro de eventos, tratamento de exceções, autenticação e autorização de usuários, etc.

A classificação é importante, pois estabelece níveis de abstrações dos serviços, promove maior clareza na decomposição do problema e aumenta as chances de reuso dos serviços.

Classificação de Serviço

Serviço de Entidade

- representa os serviços relacionados a entidades do domínio do problema da organização.
- tem existência própria independente de quais processos de negócio sejam mapeados.
- são equivalentes a entidades conceituais de um modelo orientado a objetos.
- normalmente possuem uma alta capacidade de reuso, já que podem ser usados em vários processos de negócio.
- Sua capacidade mais comum (manutenção) equivale as operações CRUD (create, read, update e delete).

Classificação de Serviço

Serviço-Tarefa

- representa um serviço relacionado a um processo de negócio.
- um serviço-tarefa normalmente é formado pela composição das capacidades de serviços de entidade.
- a principal diferença entre serviços tarefa e serviços de entidades é que estes tem seu escopo delimitado pelo conceito que ele representa (a entidade), já os serviços-tarefa extrapolam vários escopos de entidades.

Princípios de SOC

Contrato de Serviço Padronizado

- Mecanismo pelo qual se expõem o propósito e as capacidades dos serviços;
- O contrato de serviço expressa as funcionalidades, tipos e estruturas de dados, e políticas de uso.

[ERL 2009]

Princípios de SOC

Baixo Acoplamento de Serviço

- Minimizar a dependência entre a lógica e a implementação do serviço;
- Minimizar a dependência entre o contrato de serviço, sua implementação e consumidores;
- Consiste em trabalhar para que questões de projeto de serviços tenham o mínimo de dependência de questões técnicas de implementação.

[ERL 2009]

Princípios de SOC

Abstração de Serviços

- É necessário ocultar ao máximo características particulares do serviço;
- O consumidor do serviço deve conhecer o mínimo possível do serviço para usá-lo.

[ERL 2009]

Princípios de SOC

Capacidade de Reuso

- Característica intrínseca da orientação a serviço;
- Apoiada principalmente pelas especificações que definem padrões que permitem um alto grau de reuso.

[ERL 2009]

Princípios de SOC

Independência de Estado do Serviço

- Deve-se projetar serviços que não retenham informações de estado;
- Se possível somente quando for estritamente necessário;
- Informações de estado podem comprometer a disponibilidade do serviço e diminuir as possibilidades de escalabilidade.

[ERL 2009]

Princípios de SOC

Visibilidade do Serviço

- Serviços devem ser facilmente identificados e entendidos, principalmente em situações de reuso;

Composição de Serviço

- Capacidade de produzir novos serviços por meio do reuso de serviços existentes;
- Capacidade de produzir novos serviços por meio do reuso de serviços existentes;
- Composições complexas podem exigir grandes esforços de projeto para assegurar que não haja a necessidade de adequações dos serviços.

Princípios de SOC

Interoperabilidade

- Não tem uma definição única e independente;
- É obtida a partir da existência dos princípios anteriores;
 - Contrato de Serviço,
 - Baixo Acoplamento de Serviço,
 - Abstração de Serviços,
 - Capacidade de Reuso,
 - Autonomia de Serviços,
 - Independência de Estado do Serviço,
 - Visibilidade do Serviço,
 - Composição de Serviço.

[ERL 2009]

Questões Tecnológicas para SOA

Orientação a Serviços

- A abordagem orientada a serviços é independente de linguagens de programação ou sistemas operacionais.
- Permite que as organizações possam expor suas principais competências de programação.
- É baseada em padrões XML que descrevem linguagens, protocolos e implementação de interfaces autodescritivas.

[Papazoglou et al. 2008]

Questões Tecnológicas para SOA

Serviços

- São entidades autônomas que podem ser usadas de maneira independente de plataforma.
- Qualquer trecho de código ou componente de aplicativo implantado em um sistema pode ser reutilizado e transformado em um serviço disponíveis na rede.
- É baseado na ideia de compor aplicações por meio da descoberta e invocação de serviços disponíveis na rede, ao invés de construir novas aplicações.
- São construídos de forma independente do contexto em que eles são usados.

Questões Tecnológicas para SOA

Web Service

- Atualmente existem duas abordagens principais para implementar Web Service:
 - Web Service com SOAP (Simple Object Access Protocol, em português Protocolo Simples de Acesso a Objetos)
 - Web Service com REST (Representational State Transfer (REST), em português Transferência de Estado Representacional)

Questões Tecnológicas para SOA

Web Service

- SOAP é um protocolo de acesso a serviços baseados em padrões estabelecidos a um bom tempo. Originalmente desenvolvido pela Microsoft, o SOAP realmente não é tão simples quanto o acrônimo sugere.
- REST é o recém-chegado ao bloco. Ele procura corrigir os problemas com o SOAP e fornecer um método verdadeiramente simples de acesso aos serviços da Web.
- No entanto, em algumas situações o SOAP se mostra mais favorável ao uso, em outras, o REST. Contudo, ambas as técnicas têm problemas a serem considerados ao decidir qual abordagem de Web Service a ser usada.

Questões Tecnológicas para SOA

Web Service/REST

- REST fornece uma alternativa mais leve, uma vez que não usa o XML para fazer um pedido, o REST depende de um URL simples em muitos casos.
- Em algumas situações, você deve fornecer informações adicionais de maneiras especiais, mas a maioria dos serviços da Web usando o REST depende exclusivamente da obtenção das informações necessárias usando a abordagem de URL.
- Funciona exclusivamente sobre o protocolo HTTP.

Questões Tecnológicas para SOA

SOAP

- Independência de Linguagem, plataforma e transporte (o REST requer o uso de HTTP)
- Funciona bem em ambientes empresariais distribuídos (o REST assume uma comunicação ponto a ponto)
- Padronizado (WC3)
- Fornece extensibilidade na forma dos padrões WS *
- Manuseio de erros incorporado
- Automação da geração de código com a maioria das linguagens

Questões Tecnológicas para SOA

- SOA deve atender o requisito de baixo acoplamento, baseado em padrões e protocolos independentes para a implementação da computação distribuída.
- O cenário típico de SOA é de **operações de negócio representadas por uma série de invocações aos componentes de softwares distribuídos.**
- Para construir uma SOA é necessário uma comunicação altamente distribuível e um *backbone* de integração.

Questões Tecnológicas para SOA

- O ESB é um conjunto de recursos de infraestrutura implementado com tecnologia de *middleware*.
- Reduz problemas de incompatibilidades entre aplicações em plataformas heterogêneas.
- É uma combinação de servidores de integração corporativos (servidores de aplicação e *brokers* de integração, serviços Web, linguagens de transformação de dados e tecnologias de orquestração).
- Estabelece o controle adequado de mensagens (segurança, políticas, confiabilidade e contabilidade).

[Papazoglou et al. 2008]

Composição de Serviços

Uma das características importantes de COS é a capacidade de produção de soluções por meio da composição de serviços.

Existem duas abordagens gerais para compor serviços: **orquestração e coreografia**.



Composição de Serviços

Orquestração

No modelo de composição por orquestração, existe **um único ponto de controle**, denominado serviço controlador [Papazoglou et al. 2008].

As interações de processos de negócios são sempre **controladas a partir da perspectiva de uma das partes** envolvidas no processo.

A linguagem dominante para orquestração é a *Business Process Execution Language*, ou BPEL [Hwang et al. 2010].

Composição de Serviços

Coreografia

Coreografia possui uma filosofia mais colaborativa que a orquestração.

É descrita a partir das perspectivas de todas as partes, e define o comportamento observável de complementaridade entre os participantes.

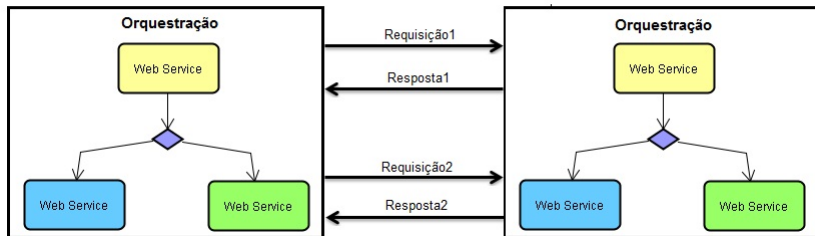
Uma linguagem bastante difundida para coreografia é a *Web Services Choreography Description Language*, abreviada como WS-CDL.

[Hwang et al. 2010]

Composição de Serviços

Orquestração e Coreografia

Coreografia



Considerações

A computação orientada a serviços permite o desenvolvimento, disponibilização e reuso de serviços a partir de novos componentes de softwares ou da transformação de elementos legados em serviços.

Permite a produção de aplicações a partir da composição de serviços em alto nível de abstração tecnológica e de plataforma de execução.

Referências



Bichier, M. and Lin, K. (2006).

Service-oriented computing.

Computer, 39(3):99–101.



Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. (2009).

Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility.

Future Generation Computer Systems, 25(6):599–616.



ERL, T. (2009).

Soa Princípios De Design De Serviços.

PRENTICE HALL BRASIL, primeira edition.



Foster, I., Zhao, Y., Raicu, I., and Lu, S. (2008).

Cloud computing and grid computing 360-Degree compared.

In *Grid Computing Environments Workshop, 2008. GCE '08*, pages 1–10. IEEE.



Hwang, S., Liao, W., and Lee, C. (2010).

Web services selection in support of reliable web service choreography.

In 2010 IEEE International Conference on Web Services (ICWS), pages 115–122. IEEE.



Joseph, J., Ernest, M., and Fellenstein, C. (2004).

Evolution of grid computing architecture and grid adoption models.

IBM Systems Journal, 43(4):624–645.



Martins, A. (2010).

Fundamentos de computação nuvem para governos.

In Amãpytuna - Computação em Nuvem: serviços livres para a sociedade do conhecimento, page 47 a 65.



Mell, P. and Grance, T. (2009).

NIST definition of cloud computing v15.



Papazoglou, M., Traverso, P., Dustdar, S., and Leymann, F. (2008).

Service-oriented computing: A research roadmap.

International Journal of Cooperative Information Systems, 17(2):223–255.



Papazoglou, M. and Van Den Heuvel, W. (2007).

Service oriented architectures: Approaches, technologies and research issues.

VLDB Journal, 16(3):389–415.



Wei, Y. and Blake, M. B. (2010).

Service-Oriented computing and cloud computing: Challenges and opportunities.

Internet Computing, IEEE, 14(6):72–75.



Zhang, Q., Cheng, L., and Boutaba, R. (2010).

Cloud computing: state-of-the-art and research challenges.

Journal of Internet Services and Applications, 1(1):7–18.