

IMPACTS OF AUTONOMOUS COMPOSITION OF SERVICES AT SERVICE-ORIENTED ARCHITECTURE

Flávio de Oliveira Silva (Escola Politécnica da USP, São Paulo, Brasil) –
flavio@pad.lsi.usp.br

Jorge L. R. Becerra (Escola Politécnica da USP, São Paulo, Brasil) -
jorge.becerra@poli.usp.br

Selma S. S. Melnikoff (Escola Politécnica da USP, São Paulo, Brasil) -
selma.melnikoff@poli.usp.br

Sérgio Takeo Kofuji (Escola Politécnica da USP, São Paulo, Brasil) -
kofuji@pad.lsi.usp.br

Colaborador

Pedro Frosi Rosa (Universidade Federal de Uberlândia, Minas Gerais, Brasil) –
frosi@facom.ufu.br

SOA has in its conception the possibility of autonomous composition of services, thereby enabling services to adapt to while running through the observation of some of its parameters or attributes to ensure the Quality of Service (QoS) and Quality of Experience (QoE). Starting from the high-level view of SOA, this paper presents several approaches that are then mapped into its basic components, making it possible to identify its impacts on the basic vision of the SOA, thus allowing that a software architect can use it as a starting point in the choice of architectural patterns for the composition of autonomous services.

Keywords. Autonomous, Composition, Service, Architecture, Patterns.

IMPACTOS DA COMPOSIÇÃO AUTÔNOMA DE SERVIÇOS NA ARQUITETURA SOA

SOA possui em sua concepção a possibilidade de composição autônoma de serviços, permitindo desta forma que os serviços possam adaptar-se durante sua execução através da observação de alguns de seus parâmetros ou atributos a fim de garantir a qualidade do serviço (Quality of Service - QoS) bem como a qualidade da experiência (Quality of Experience - QoE). Partindo da visão de alto nível da SOA este trabalho apresentará várias abordagens que serão então mapeadas em seus componentes básicos, possibilitando identificar seus impactos na visão básica da SOA, permitindo desta forma que o arquiteto de software possa utilizá-lo como um ponto de partida na escolha de padrões arquiteturais para a composição autônoma de serviços.

Palavras-Chave: Autônomos, Composição, Serviços, Arquitetura, Padrões.

1. Introdução

A arquitetura orientada a serviços (Service-Oriented Architecture – SOA) é uma abordagem que possui como proposta a possibilidade de construção de aplicações baseadas em serviços. Do ponto de vista da engenharia de software, SOA pode ser considerada um estilo arquitetural, conforme Viering et al. (2009). Em sua concepção SOA apresenta a possibilidade de composição autônoma de serviços, permitindo desta forma que os serviços possam adaptar-se durante sua execução através da observação de alguns de seus parâmetros ou atributos. Desta forma, uma variedade de aplicações pode ser construída, conforme exemplos mostrados por Huang et al. (2010). Nestas aplicações é possível considerar, durante a execução, atributos de qualidade de forma a garantir a qualidade dos serviços (Quality of Service – QoS) bem como a qualidade da experiência (Quality of Experience – QoE) no uso deste serviço, definidos por Hershey et al. (2009)

Através dos Web Services é possível uma realização da SOA, porém, em relação a composição autônoma de serviços pode-se afirmar que esta arquitetura não foi realizada completamente visto que a maioria das aplicações são construídas no estilo arquitetural conhecido como Chamada e Retorno (Call and Return), conforme classificação mostrada por Majidi et al. (2010).

Este trabalho tem por objetivo investigar o estado atual da realização da SOA em relação à composição autônoma de serviços. Além disso, o mesmo apresentará possíveis abordagens arquiteturais que possibilitam este tipo de composição, permitindo desta forma que o arquiteto de software possa utilizá-lo como um ponto de partida na escolha de padrões arquiteturais para a composição autônoma de serviços.

Partindo da visão de alto nível da SOA as várias abordagens serão mapeadas em seus componentes básicos, sendo possível desta forma esclarecer o impacto destas abordagens na visão básica da SOA.

Várias publicações tratam deste assunto. Por exemplo: Rambold et al. (2009) aborda a descoberta de serviços, um aspecto importante relacionado na composição, porém a SOA engloba outros igualmente relevantes; Ibrahim; Le Mouél (2009) realizam uma classificação de middlewares para composição de serviços e para isto propõem uma arquitetura de referência realizando a classificação através do mapeamento destes middlewares nesta arquitetura proposta versus atributos de qualidade; Rao; Su, X. (2005) realizam uma classificação de abordagens de composição automática de Web Services baseado em conceitos de fluxo de trabalho e inteligência artificial; Mascolo et al. (2005) apresentam a arquitetura de vários middlewares com foco em sistemas embarcados interconectados em rede e neste caso a classificação destes middlewares é feita baseada em um conjunto de requisitos funcionais e não-funcionais adequados para este domínio, além disso, nem todos são baseados na SOA; Kuropka; Meyer (2005) abordam o assunto da composição de serviços baseada em uma abordagem manual versus uma abordagem automatizada.

O restante deste artigo está organizado em seções assim divididas: Seção 2 apresenta a SOA e alguns conceitos relacionados. A seção 3 apresenta os conceitos envolvidos na composição autônoma de serviços. A seção 4 apresentará arquiteturas SOA para composição dinâmica de serviços e um mapeamento destas arquiteturas na visão padrão da SOA. Finalmente a seção 5 apresenta as conclusões e trabalhos futuros relacionados com o tema.

2. Service-Oriented Architecture (SOA)

O termo arquitetura orientada a serviços (Service-Oriented Architecture – SOA) foi apresentado pela primeira vez por Schulte; Natis (1996). O conceito de SOA por sua vez pode ser realizado por diferentes tecnologias como, por exemplo, utilizando CORBA (Common Object Request Broker Architecture) como exposto por Schulze; Madeira (1997). Porém com o advento dos Web Services, definidos em Booth et al. (2004), foi possível a realização desta arquitetura considerando a infraestrutura da Web.

Do ponto de vista da Engenharia de Software, SOA pode ser considerado um estilo arquitetural. Porém no geral SOA está relacionado com conceitos mais amplos como SOC (Service-Oriented Computing) proposta por Papazoglou (2003) que englobam não somente a implementação de serviços, mas, uma teoria de engenharia de software completa que envolve o ambiente, a metodologia para desenvolvimento, os estilos arquiteturais, o modelo de programação e de testes, como descrito por Chen, Y. et al. (2010)

No geral SOA está relacionado com sistemas distribuídos sendo alguns de seus princípios básicos, conforme mostrado por Valipour et al. (2009): o baixo acoplamento entre os componentes; a interoperabilidade entre os mesmos; a transparência de localização; a capacidade de composição.

A Figura 1 mostra a visão básica da SOA, apresentando seus principais componentes em uma visão de alto nível. Um serviço é oferecido pelo “Service Provider”, que fornece também uma descrição básica deste serviço, representada pelo elemento “Contract” na figura. Este contrato pode ser então publicado em um registro. O “Service Registry” é o elemento responsável por manter um registro dos serviços disponíveis. Um consumidor destes serviços, o “Service Consumer”, poderá realizar uma procura dos serviços neste registro e à medida que um serviço é escolhido, haverá uma ligação (bind) direta entre o provedor e o consumidor.

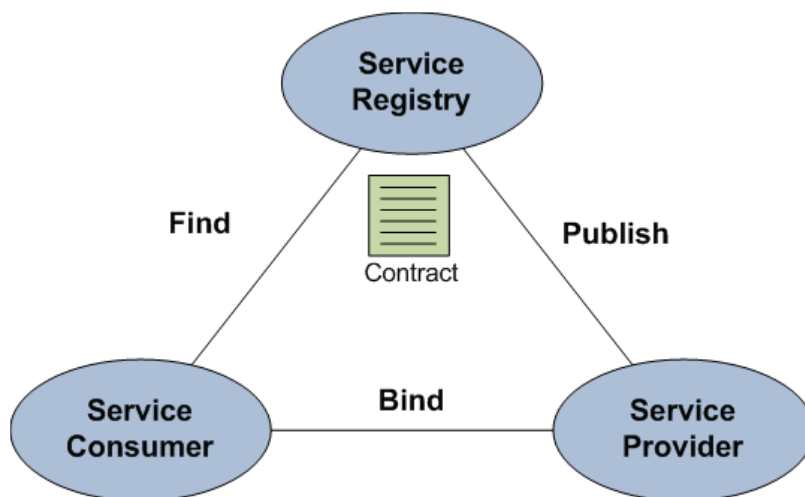


Figura 1 – Visão básica da SOA

O contrato representa a interface com o serviço e deve especificá-lo de forma clara. No caso dos Web Services, o contrato representado na figura é implementado pela WSDL (Web Services Description Language) definida por Christensen et al. (2001) e atualizada por Chinnici et al. (2007).

Um aspecto que também pode ser considerando é a maturidade em SOA. Em Welke et al. (2005), o autor apresenta níveis de maturidade no uso de SOA e neste modelo o nível mais alto, indicado como “Otimizado” vislumbra uma arquitetura adaptável com a presença de sistemas autônomos. Através desta visão é possível perceber que a realização da arquitetura pode acontecer em diferentes níveis, sendo que a maioria deles não consideram a arquitetura em toda sua plenitude e capacidade.

3. Composição Autônoma de Serviços

A computação autônoma definida por Bhakti; Abdullah (2010) e por Kephart; Chess (2003), é inspirada no funcionamento do sistema nervoso autônomo, que é responsável por prover “serviços” como, por exemplo, os batimentos cardíacos.

Através desta abordagem os sistemas de software podem adaptar-se durante a sua execução alterando seu funcionamento baseado em fatores relacionados, por exemplo, com a qualidade do serviço (QoS) ou mesmo a qualidade de experiência (QoE) no uso deste serviço.

Para realizar este comportamento os sistemas de computação autônoma devem apresentar algumas propriedades que devem ser satisfeitas. Estas propriedades, conhecidas como Self-*, são mostradas na Tabela 1.

Tabela 1- Propriedades dos Sistemas Autônomos

PROPRIEDADE	VISÃO
Auto-configurável (Self-Configurable)	Configuração do sistema e de seus componentes segue políticas previamente definidas, e é realizada de forma automática
Auto-otimizável (Self-optimising)	Sistema e seus componentes buscam continuamente oportunidades para melhorar seu desempenho e eficiência
Auto-corrigível (Self-healing)	Sistema detecta, realiza o diagnóstico e repara componentes afetados
Auto-proteção (Self-protection)	Sistema possui a capacidade de defender-se contra ataques maliciosos ou falhas

Em resumo os sistemas autônomos são auto-gerenciáveis e para tanto é necessário em sua arquitetura um componente capaz de realizar a gestão do sistema e sua adaptação conforme o evento ocorrido. Este componente deve ser capaz de: monitorar, analisar, planejar e executar ações sob um componente ou sistema que está sob a sua gestão.

A composição de serviços é um princípio fundamental para a computação autônoma, visto que um serviço ou sistema monolítico não poderia adaptar-se. Para tanto é necessário que a concepção da arquitetura deste sistema considere que o mesmo possuirá diferentes componentes ou serviços que serão responsáveis pela execução do comportamento desejado.

A Figura 1, que apresenta os componentes básicos da SOA, contém implícita a possibilidade de realização da computação autônoma através da composição dinâmica de serviços.

Neste caso, um serviço que é composto de outros serviços, poderia em tempo de execução, realizar uma busca no registro, encontrar outro serviço que atenderia as

especificações e de posse de seu contrato, realizaria a ligação com o mesmo, sendo possível, a partir deste momento, utilizar este novo serviço encontrado e selecionado.

4. Arquiteturas para Composição Autônoma de Serviços

Existem várias arquiteturas propostas para a composição autônoma de serviços. A seguir serão apresentadas algumas delas e será feito um mapeamento das mesmas na visão básica da SOA, apresentada na Figura 1.

4.1. Enterprise Service Bus (ESB)

Xudong Li et al. (2009) apresentam uma arquitetura de referência para a composição dinâmica de serviços.

O ESB (Enterprise Service Bus) possui múltiplas responsabilidades como, por exemplo: segurança, autenticação, autorização, gestão de acesso, roteamento de serviços, conversão de dados, garantia de entrega de mensagens, controle de transação. Além disso, o ESB pode tratar de aspectos de serviços como: registro, busca, monitoramento, entre outros. Desta forma o ESB é responsável por muitas das propriedades Self-* de um sistema autônomo.

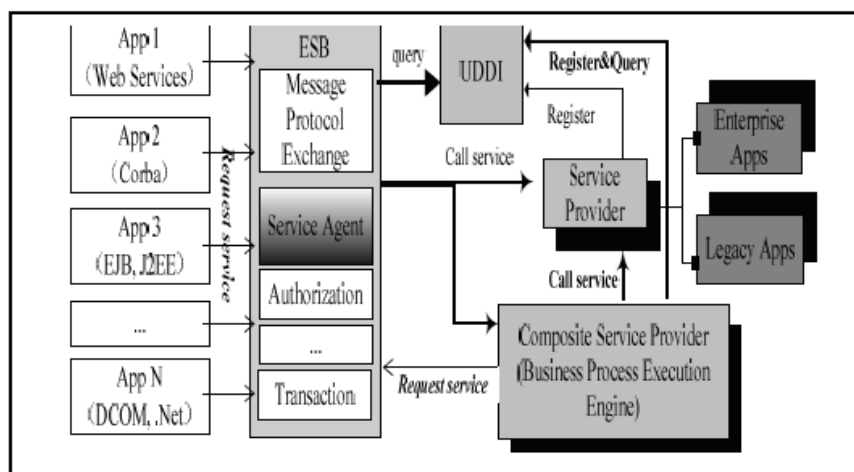


Figura 2 – Arquitetura de Referência para um Enterprise Service Bus (ESB) apresentada Xudong Li et al. (2009)

O ESB pode ser visto como uma camada que está acima da infraestrutura de rede e que funciona como um barramento onde é possível a obtenção dos serviços, como descrito por Wen et al. (2009). Neste caso um serviço poderia ser requisitado ao ESB. O ESB seria responsável por localizar este serviço no registro.

Outro aspecto importante do ESB é que o mesmo pode implementar diferentes interfaces para os diferentes provedores de serviço oferecendo, por outro lado, uma interface única para os consumidores acessarem tais serviços.

Uma aplicação poderia solicitar um serviço ao ESB, que neste caso realiza uma pesquisa em um registro, baseado em Universal Description, Discovery and Integration (UDDI). O serviço solicitado neste caso é um serviço composto de vários outros serviços. O ESB é responsável pela execução deste serviço composto bem como por manter a sua composição ao longo do tempo, ou seja, os componentes deste serviço podem ser alterados pelo ESB.

Se mapearmos a estratégia do ESB, pode-se constatar que a visão básica da SOA é a mostrada na Figura 3.

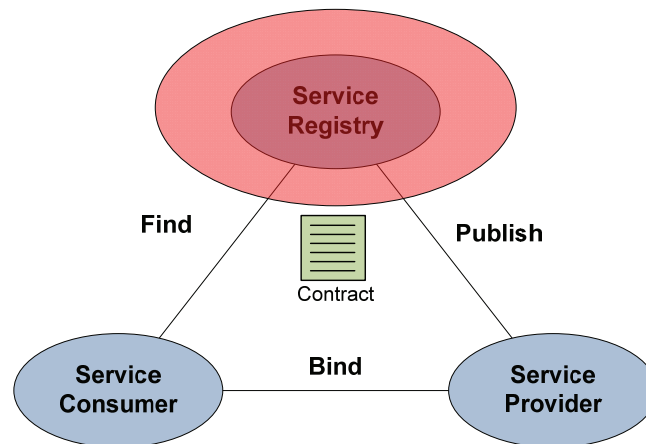


Figura 3 – Impacto do ESB na visão básica da SOA, adaptado de Tian; Su, Y. (2010)

Neste caso, apenas o uso do registro não é suficiente para permitir a composição autônoma de serviços. O ESB, com todas as funcionalidades, é o elemento da arquitetura que contém o registro dos serviços e que propicia a composição autônoma de serviços. Portanto o elemento básico da SOA, o registro, pode ser visto como um componente expandido.

4.2. Service Composition Middleware (SCM)

Um ambiente onde SOA pode ser utilizada é na computação pervasiva, como exposto por Jiehan Zhou et al. (2010). Neste cenário a composição autônoma realiza um importante papel, visto que os serviços podem ser disponibilizados por ambientes heterogêneos, móveis, dinâmicos por natureza. Ibrahim; Le Mouél (2009) apresentam várias arquiteturas disponíveis para a composição de serviços em um ambiente de computação pervasiva.

Como este artigo apresenta várias arquiteturas, o mesmo utiliza uma arquitetura de referência, chamada Service Composition Middleware (SCM). As arquiteturas são então mapeadas neste modelo e classificadas em relação ao mesmo e aos seus atributos de qualidade.

Esta arquitetura de referência, mostrada na Figura 4, será utilizada neste trabalho como uma visão de arquitetura para a composição autônoma de serviços.

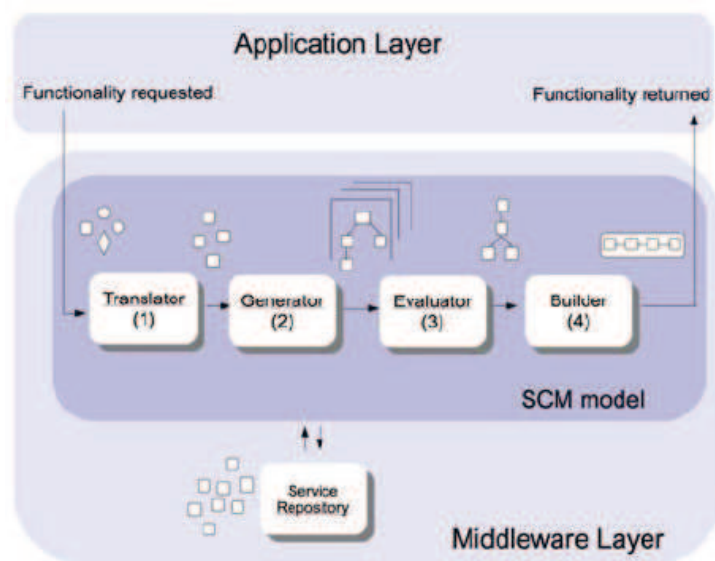


Figura 4 – Arquitetura de referência SCM proposta por Ibrahim; Le Mouël (2009)

O SCM possui quatro componentes principais responsáveis por implementar a composição autônoma de serviços: Translator, Generator, Evaluator e Builder.

Uma aplicação que necessite de um serviço enviará uma requisição ao SCM. Esta requisição deverá conter uma descrição das funcionalidades requeridas. Considerando que essas requisições poderão ser realizadas em diferentes formatos é necessário que o SCM faça uma tradução desta requisição em um formato padrão de uso interno do mesmo. Este recurso é importante em sistemas pervasivos, pois diferentes dispositivos podem prover ou requisitar serviços de distintas maneiras.

Assim que uma requisição é traduzida a mesma estará em um formato padrão utilizado pelo middleware e desta forma será possível que esta requisição seja tratada pelo componente seguinte, o Generator. Este componente é responsável por avaliar a solicitação e gerar os planos de composição possíveis naquele momento a fim de responder à requisição. Em um ambiente pervasivo uma mesma solicitação pode ser respondida de diferentes formas, cabendo a este módulo gerar as possíveis respostas. Este componente é responsável por interagir com o registro a fim de obter os possíveis serviços, seus atributos e montar os planos de execução.

O componente Evaluator recebe então os planos de composição possíveis em um dado contexto e considerando propriedades de QoS, avalia estes planos escolhendo o mais adequado em um dado momento. Deve ser ressaltado que este processo deve ser continuamente executado, visto que um dispositivo que em um dado momento pode fornecer um serviço, em momento posterior, pode ser desativado. Neste caso um novo plano deve ser escolhido para execução daquele mesmo serviço.

Finalmente o componente Builder é responsável por executar o plano escolhido realizando então a composição e a execução do serviço requisitado. Este componente é responsável por realizar a ligação com serviços e invocá-los.

O processo de geração e avaliação precisa ser executado continuamente a fim de o serviço composto oferecido possa ser alterado de acordo com as variações ao longo do tempo de seus componentes. Desta forma as propriedades Self-* poderão ser garantidas por esta arquitetura.

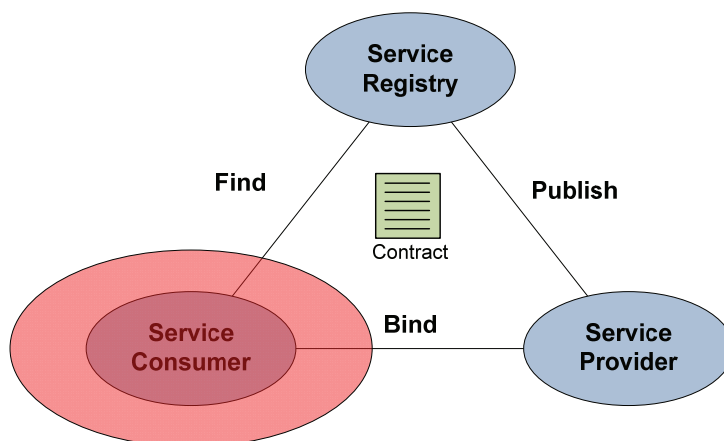


Figura 5 – Impacto do SCM na visão básica da SOA

Esta arquitetura de referência mostra uma abordagem possível para a composição autônoma de serviços. Ao mapearmos o SCM na visão básica da SOA, podemos notar o impacto na mesma, conforme apresentado na Figura 5.

Neste caso para realizar a composição autônoma de serviços fez-se necessário expandir o lado consumidor da SOA. Esta expansão é representada pelo SCM, que não faz parte do cliente propriamente dito, mas é um componente adicional entre o consumidor e o registro de serviços.

5. Conclusão e Trabalhos Futuros

Uma arquitetura para a composição autônoma de serviços necessita de vários mecanismos a fim de garantir as propriedades Self-* e os atributos de qualidade especificados, sendo que ainda existem vários desafios para o arquiteto de software neste campo.

A visão básica dos componentes da SOA, mostrada na Figura 1, apesar de conter a possibilidade de composição autônoma de serviços não é suficiente para a complexidade envolvida. Faz-se necessário envolver o registro com componentes e comportamentos para realizar a composição autônoma, conforme mostrado na Figura 3 ou então, é necessário que o consumidor utilize novos componentes que propiciarão a composição autônoma, conforme mostrado na Figura 5.

Por outro lado, o provedor, o outro componente da visão básica da SOA não sofre variação, sendo que neste caso a composição autônoma é realizada por um dos outros dois componentes. Podemos concluir que para a composição autônoma de serviços a visão básica da SOA demanda necessariamente um componente entre o consumidor e o registro.

Um aspecto importante é que a composição autônoma está associada ao contexto sendo que as soluções apresentadas são para domínios específicos não sendo encontrada uma arquitetura de referência que possa ser utilizada entre diferentes domínios.

Como trabalho futuro, as várias arquiteturas, encontradas para a composição autônoma de serviços, poderiam ser detalhadas e mapeadas em estilos arquiteturais genéricos, facilitando assim a disseminação de estratégias e a escolha de padrões arquiteturais, iluminando ainda mais, esta área da engenharia de software.

Referências

- BHAKTI, M. A.; ABDULLAH, A. B. Design of an autonomic services oriented architecture. In: Information Technology (ITSim), 2010 International Symposium in. **Anais...** . v. 2, p.805–810, 2010.
- BOOTH, D.; HAAS, H.; MCCABE, F.; ET AL. Web Services Architecture. Disponível em: <<http://www.w3.org/TR/ws-arch/>>. Acesso em: 29/11/2010.
- CHEN, Y.; LI, X.; YI, L.; ET AL. A ten-year survey of software architecture. In: Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on. **Anais...** . p.729–733, 2010.
- CHINNICI, R.; MOREAU, J.; RYMAN, A.; WEERAWARANA, S. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. Disponível em: <<http://www.w3.org/TR/wsdl20/>>. Acesso em: 29/11/2010.
- CHRISTENSEN, E.; CURBERA, F.; MEREDITH, G.; WEERAWARANA, S. Web Service Definition Language (WSDL). Disponível em: <<http://www.w3.org/TR/wsdl/>>. Acesso em: 29/11/2010.
- HERSHEY, P.; PITTS, J.; OGILVIE, R. Monitoring real-time applications events in net-centric enterprise systems to ensure high quality of experience. In: Military Communications Conference, 2009. MILCOM 2009. IEEE. **Anais...** . p.1-7, 2009. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5379864>. Acesso em: 5/11/2010.
- HUANG, J.; ZHANG, Y.; YEN, I. L.; ET AL. Real-Time Service-Oriented Distributed Governance. In: 2010 6th World Congress on Services. **Anais...** . p.479–484, 2010. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5575557&isnumber=5575460>>. Acesso em: 2/11/2010.
- IBRAHIM, N.; LE MOUËL, F. A Survey on Service Composition Middleware in Pervasive Environments. **International Journal of Computer Science Issues(IJCSI)**, v. 1, p. 1, 2009.
- JIEHAN ZHOU; GILMAN, E.; YLIANTTILA, M.; RIEKKI, J. Pervasive Service Computing: Visions and Challenges. In: Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on. **Anais...** . p.1335-1339, 2010.
- KEPHART, J.; CHESS, D. The vision of autonomic computing. **Computer**, v. 36, n. 1, p. 41-50, 2003.
- KUROPKA, D.; MEYER, H. **Survey on Service Composition**. Univ.-Verl., 2005.
- MAJIDI, E.; ALEMI, M.; RASHIDI, H. Software Architecture: A Survey and Classification. In: Communication Software and Networks, 2010. ICCSN'10. Second International Conference on. **Anais...** . p.454–460, 2010.
- MASCOLO, C.; HAILES, S.; LYMBEROPOULOS, L.; ET AL. Survey of middleware for networked embedded systems. **Project Report: http://www.ist-runes.org/docs/deliverables/D5_01.pdf**, 2005.
- PAPAZOGLU, M. P. Service-oriented computing: Concepts, characteristics and

directions. , 2003.

RAMBOLD, M.; KASINGER, H.; LAUTENBACHER, F.; BAUER, B. Towards Autonomic Service Discovery A Survey and Comparison. In: Services Computing, 2009. SCC'09. IEEE International Conference on. **Anais...** . p.192–201, 2009.

RAO, J.; SU, X. A survey of automated web service composition methods. **Semantic Web Services and Web Process Composition**, p. 43–54, 2005.

SCHULTE, W. R.; NATIS, Y. V. "Service Oriented" Architectures, Part 1. , 1996. Gartner Group.

SCHULZE, B.; MADEIRA, E. Contracting and moving agents in distributed applications based on a service-oriented architecture. In: Mobile Agents. **Anais...** . p.74–85, 1997.

TIAN, Y.; SU, Y. Research of service-oriented analysis and design method. In: Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on. **Anais...** . v. 9, p.733–736, 2010.

VALIPOUR, M. H.; AMIRZAFARI, B.; MALEKI, K. N.; DANESHPOUR, N. A brief survey of software architecture concepts and service oriented architecture. , 2009.

VIERING, G.; LEGNER, C.; AHLEMANN, F. The (Lacking) Business Perspective on SOA–Critical Themes in SOA Research. **Proceedings of Wirtschaftsinformatik**, v. 2009, p. 45–54, 2009. Disponível em: <[http://web.iwi.unisg.ch/org/iwi/iwi_pub.nsf/wwwPublYearGer/70CF80884613176AC125750F005D0727/\\$file/SOA%20Impact%20Final%2015%20GVi.pdf](http://web.iwi.unisg.ch/org/iwi/iwi_pub.nsf/wwwPublYearGer/70CF80884613176AC125750F005D0727/$file/SOA%20Impact%20Final%2015%20GVi.pdf)>. Acesso em: 1/11/2010.

WELKE, R.; HIRSCHHEIM, R.; SCHWARZ, A. Service Oriented Architecture Maturity. **Computer**, v. 99, n. 1, 2005.

WEN, R.; MA, Y.; CHEN, X. ESB Infrastructure's Autonomous Mechanism of SOA. In: Intelligent Ubiquitous Computing and Education, 2009 International Symposium on. **Anais...** . p.13–17, 2009.

XUDONG LI; TAO XU; CHUNXIA ZHANG. A Reference Framework For Service Oriented Architecture of Enterprise Application. In: Management and Service Science, 2009. MASS '09. International Conference on. **Anais...** . p.1-5, 2009.