

Utilizando a tecnologia de orientação a agentes para implementar a arquitetura orientada a serviços

Leandro Ramos da Silva¹, Márcia Ito²

Laboratório de Pesquisa em Ciências de Serviço (LaPCiS) - Centro Estadual de Educação Tecnológica “Paula Souza” – CEETEPS – São Paulo – Brasil

¹lramos@gmail.com,²marcia.ito@centropaulasouza.sp.gov.br

Resumo: O conceito para a implementação de arquiteturas de software orientada a serviços faz com que o projeto de software seja desenvolvido de forma a compartilhar funcionalidades com outros softwares. Organizações como a *Object Oriented Management Group* (OMG) disponibilizam padrões de implementações para arquiteturas orientadas a serviços, porém nem tudo foi explorado, assim este artigo propõe a apresentar o uso da tecnologia orientada a agentes na implementação de arquiteturas de software orientadas a serviços (SOA).

Palavras-chave: Tecnologia de Orientação a Agentes, Arquitetura Orientada a Serviços, Arquitetura de Software

Introdução

Segundo a *Object Oriented Management Group* (OMG, 2007) e Carril (2004), a Tecnologia da Informação (TI) é tratada com certa moderação quanto aos investimentos, o que justifica a preocupação de diversos segmentos sobre a forma de aplicar seus recursos em Sistemas de TI, e a crescente demanda pela utilização de softwares prontos.

A Arquitetura Orientada a Serviços ou *Service Oriented Architecture* (SOA) é uma solução, que ao utilizar conceitos de computação distribuída, permite projetar componentes de software que, pela rede, interagem entre si. Esses componentes são denominados “serviços”, pois são funcionalidades desenvolvidas para que se possam ser reutilizados entre diversas organizações (IBM, 2007). No entanto essa utilização pode se tornar demasiadamente complexa, segundo Davydov (Davydov, 2007).

Um sistema complexo implica em um grande quantidade de partes interagindo de uma forma difícil de entender e prever. O alto número de

componentes cria dificuldade na compreensão da estrutura relacionada, e o número não trivial de interações entre os componentes cria incerteza de prever o comportamento do sistema, sem aparente padrão, fazendo com que ele pareça aleatório (Tan, 2005).

A tecnologia de agentes possui características poderosas para o projeto de softwares complexos, dado seu seguinte funcionamento: um agente é abstraído para resolver um problema em particular de forma autônoma, mas é capaz ainda de interagir com outros de forma a possibilitar a resolução de um problema que individualmente não conseguiria (Wooldridge, 2002).

Assim sendo, o presente trabalho se propõe a apresentar como a tecnologia de orientação a agentes (IEEE, 2006) pode auxiliar na implementação do SOA através de uma *Plataforma de Agentes* (INOQQ, 2007), onde possa ser suportada a característica de inteligência distribuída orientada a agentes.

Arquitetura Orientada a Serviços Auxiliando a Governança Corporativa

A SOA, é largamente utilizado para a integração de sistemas. Seu objetivo principal, segundo a IBM e OMG (apud. Linthicum, 2006), é de prover serviços de software de uma maneira mais dinâmica que a usual, com redução de custos e simplificação nas integrações da grande quantidade de sistemas existentes atualmente, tais como sistemas de relacionamento com o cliente (CRM), de administração corporativa (ERP), de gerenciamento, entre outros (IBM, 2007).

Ainda segundo a IBM (IBM, 2007), SOA como um paradigma transfere a preocupação dos detalhes de implementação (em se tratando de projeto de sistemas de informação) para uma preocupação maior com os grandes componentes do sistema. Em um nível macro, transfere a preocupação do reuso para um formato de construção de software baseado na “construção em blocos” (Silva, 2003).

A implementação de um serviço em SOA, deve ser fracamente acoplados, independentes de localização física, trabalhar em conjunto com outros serviços, permitir composição de serviços e possuir um registro de serviços, onde todos os serviços são catalogados (IBM, 2007).

A Orientação a Agentes

Um Agente é um componente de software projetado e implementado para resolver um problema específico de forma autônoma, com controle sobre suas ações dada uma seqüência de regras e objetivos pré-definidos (Wooldridge,2002),(Ito, 1999).

De acordo com Ito (2006), “Um agente pode questionar outro na execução de alguma ação, comunicá-lo de que há falta de dados para executá-la, pedir outras informações para complementá-la, ou até mesmo recusar a sua execução.”

Estas regras possibilitam que ele readapte seu comportamento em tempo de execução à medida que são recebidos novos estímulos do ambiente que ele executa (Wooldride, 2002).

Isso é possível devido à estrutura sobre a qual um agente é construído, que é baseada no contexto de sensor de ambiente – capaz de monitorar as alterações no ambiente o qual está sendo executado; estrutura de decisão – onde através de uma seqüência de regras pré-definidas um agente toma decisões; e um atuador – por onde um agente interage com o ambiente. A figura 1 ilustra a atuação do agente conforme explicado anteriormente. (Wooldridge, 2002, p.23)

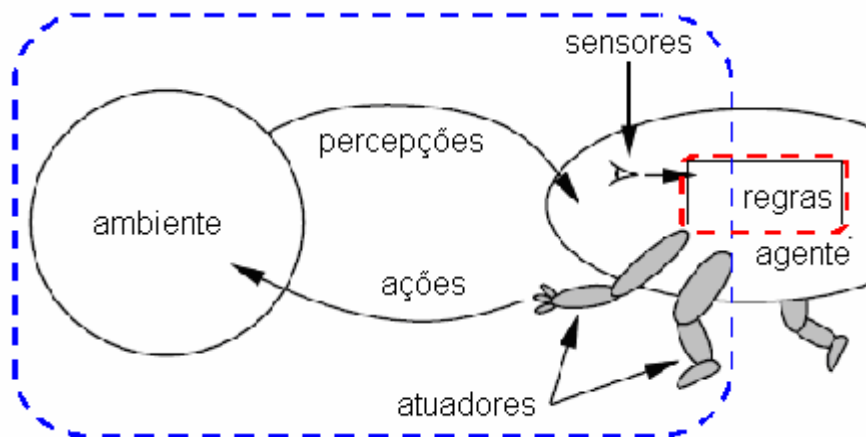


Figura 1– A atuação de um agente – adaptado de Jade (2007)

Porém, dado um determinado problema a ser resolvido um só agente pode não ser capaz de produzir uma resposta, e dessa forma a interação entre diversos agentes torna-se necessária. Ao conjunto de agentes que interagem entre si é chamado de sociedade (Gell-Mann, 1994). Assim, segundo Wooldridge (2002), “um esquema de agentes define como um agente interage com os demais, e

como as informações e recursos fluem externamente para outros agentes”. E esta característica de autonomia e ao mesmo tempo capacidade de atuação em sociedade é muito aderente ao conceito do SOA.

Utilizando a tecnologia de agentes para implementar a arquitetura orientada a serviços

Segundo o grupo de pesquisas de SOA do INNOQ (INNOQ, 2007), diferentes padrões para implementação de SOA vêm sendo desenvolvidos de forma a possibilitar a integração de sistemas tão distintos na base já instalada de softwares no cenário de TI mundial. Mas por outro lado, segundo a OMG (OMG, 2007), a diversidade de protocolos, especificações e ferramentas dificultam muito a implementação de mecanismos como SOA e BPM, razão pela qual quando nos referimos à integração de diversos serviços em plataformas tão distintas coloca-se em evidência a falta de uma diretriz prática de implementação sobre como fazer os sistemas em SOA operarem.

Um exemplo disto são os problemas de negociação em determinadas circunstâncias que ainda não estão resolvidos, como a busca de serviços (ZHANG, 2005). Apesar das iniciativas como o *WS-Discovery* (Beatty, 2005) para busca de serviços na Web, o mecanismo de contrato e negociação ainda é desprovido de inteligência, ou de uma estrutura que permita a tomada de decisões, e é baseado no mecanismo de *broadcasting*, no intuito de encontrar um serviço que possa responder às suas requisições.

O mesmo pode ser observado para as iniciativas de padronização dos formatos de transação, como *Web-Service Coordination*, *Web-Service Atomic Transaction* e *WS-Cordination Framework* (Bunting, 2003)

Como citado anteriormente, a título de exemplo com relação à busca de serviços por *Broadcasting*, a orientação a Agentes já implementa modelos de agentes em sociedades Federadas (Ito, 1999), onde há um agente que facilita a busca de serviços para posterior negociação entre os demais agentes envolvidos.

E com relação às demais características da implementação do SOA, segundo o Linthicum Group (Linthicum, 2006), as características mais relevantes são a agilidade, a adaptabilidade, o reuso funcional, o gerenciamento de mudança

independente, a interoperabilidade, ao invés de integração ponto-a-ponto e orquestração ao invés de integração.

Organizações de Agentes já estabelecidas, como o CENINT (Ito, 2006) ou a iniciativa Open-Source Jade (2007), disponibilizam tais implementações prontas. Nos dois casos citados, ambos implementam um modelo de Busca, Negociação e Contratação de serviços denominada ContractNET (Ito, 1999).

Além disso, uma das opções identificadas pela OMG (OMG, 2007) para adaptação de sistemas já construídos para utilização de SOA é a utilização de *middlewares* - camada entre o sistema e suas aplicações (Gialdi, 2004) que são capazes de gerenciar as informações dos sistemas existentes de forma a disponibilizá-las a outros sistemas. Assim, Gialdi (Gialdi, 2004) propôs um *Middleware Reflexivo* (Kon, 2002) baseado em agentes, permitindo que aplicações distintas utilizem os Agentes como mediadores de aplicação.

Conclusões e possibilidades de estudos futuros

Como devido às suas características cada agente é capaz de atuar independentemente, a orientação a Agentes é uma grande candidata para prover implementação prática ao SOA, uma vez que seu modelo de representação de serviços e modelo de componentes preza pela padronização e independência de arquitetura. Seja para implementação de novos serviços em SOA, ou até mesmo para a migração de sistemas já existentes para o formato de serviços.

Bibliografia

BEATTY, John; KAKIVAYA, Gopal; KEMP, Devon; Et. al. Web Services Dynamic Discovery (WS-Discovery). 2005. Disponível em: <<http://specs.xmlsoap.org/ws/2005/04/discovery/ws-discovery.pdf>>. Acessado em: Julho de 2007.

BOOCH, G. "Object-oriented analysis and design with applications" Addison Wesley. 1994

BUNTING, Doug; CHAPMAN, Martin; HURLEY, Oisin; Et.al. Web Services Composite Application Framework (WS-CAF). 2003. http://www.arjuna.com/library/specs/ws_caf_1-0/WS-CAF-Primer.pdf

Carr, Nicholas G., *Does IT Matter?*, Harvard Business School Press, 2004.

DAVYDOV, Mark. Soa Adventures. IBM Developer Works. <http://www.ibm.com/developerworks/webservices/library/ws-soa-adventure2/index.html>. Acessado em Agosto de 2007

Gell-Mann, M. The Quark and the Jaguar. Freeman and Co., NY, 1994

GIALDI, Marcos Vinicius. Dissertação (mestrado). Universidade Estadual de Campinas . Instituto de Computação. Campinas, SP. 2004

Kon, F., Costa, F., Campbell, R., Blair, G. *The Case for Re°ective Middleware*. Communications of the ACM, 45(6), Junho 2002.

IBM. Definindo soluções em SOA e aplicando governança de projeto, técnica e operacional. Disponível em: <https://www-304.ibm.com/jct09002c/university/scholars/courseware/repository/SOA/SW718/SW718Topic1.pdf>. Acessado em: Setembro de 2007

IEEE, Comitê Técnico de Agentes Industriais – Grupo de Arquitetura Orientada a Serviços. Sistema Multiagentes e Arquitetura Orientada a Serviços. 2006. Disponível em: <http://www.tcia.ipb.pt/?m_id=tc.soa&c=tc.soa>. Acessado em: Setembro de 2007.

INOQQ. Web Services Standards. Site da Instituição. Disponível em: <<http://www.innoq.com/soa/wsstandards/poster/>>. Acesso em: Agosto de 2007.

ITO, M. UM MODELO DE GESTÃO DE PACIENTE CRÔNICO BASEADO NOS CONCEITOS DE RELACIONAMENTO COM O CLIENTE. São Paulo, 2006. 153

p. Dissertação (Doutorado em Engenharia). Escola Politécnica, Universidade de São Paulo, São Paulo, 2006.

ITO, M. Uma análise do fluxo de comunicação em organizações dinâmicas de agentes. São Paulo, 1999. 141 p. Dissertação (Mestrado em Engenharia Elétrica). Escola Politécnica, Universidade de São Paulo, São Paulo, 1999.

JADE, Ambiente de Desenvolvimento de Aplicação Java. Site da Instituição. Disponível em: <http://jade.tilab.com>. Acessado em: Junho de 2007

LINTHICUM, David. Steps to SOA Success. OMG SOA Information Day. 2006. Disponível em: http://soa.omg.org/SOA-Info-Day_12-06.htm. Acessado em Junho de 2007;

OASIS. Arquitetura Orientada a Serviços. Conferência SOA ABSIG. St. Louis, EUA, 2006. Disponível em: <http://soa.omg.org/SOA%20ABSIG%20Meeting%20Presentations%20April%202006.htm>. Acesso em: Julho de 2007.

OMG. UML Profile and Metamodel for Services (UPMS). Site da Instituição. Disponível em <<http://soa.omg.com/>>. Acesso em: 04/09/2007.

SILVA, Leandro Ramos da. Reuso com Orientação a Objetos. Monografia de Graduação. Faculdade de Tecnologia de São Paulo. São Paulo, 2003

TAN, Joseph;, WEN, H.Joseph; AWAD, Neveen. Health care and services delivery systems as complex adaptative systems. Communications of the ACM. Volume 48, Número 5 (2005), Páginas 36-44.

WEAVER, Rick. The Business Value of the Service Component Architecture (SCA) and Service Data Objects (SDO) in SOA Programming Model. IBM, November 2005. (cited 2006, Nov 26). Available from: <http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-sca/SOAProgrammingModelBusinessValue.pdf>

WOOLDRIDGE, Michael J. An introduction to multiagent systems. England: John Wiley & Sons, 2002.

ZHANG, T.I.; JIANG, H. A Framework of Incorporating Software Agents into SOA. From Proceeding (481) Artificial Intelligence and Soft Computing. 2005. <http://www.actapress.com/PaperInfo.aspx?PaperID=21829>