

Geração de Comportamentos de Agentes para Acessar Serviços *Web*

Tarcísio Holanda, Leonardo Ayres e Vasco Furtado

Universidade de Fortaleza – UNIFOR

thq@edu.unifor.br

leoayresm@yahoo.com.br

vasco@unifor.br

1. Introdução

A valorização do conhecimento pelas organizações tem ganhado maior relevância atualmente devido à alta competitividade do mercado e à alta velocidade em que novas informações surgem. Para isso, a organização e compartilhamento do conhecimento dentro de todos os níveis de uma organização tornam-se uma saída possível através da Gestão do Conhecimento. Segundo [1], Gestão de Conhecimento pode ser entendida como um conjunto de processos para capturar, armazenar e disseminar conhecimento dentro de uma organização com o objetivo de contribuir com o compartilhamento e reuso de conhecimento dentro da organização. Muito das informações que poderiam auxiliar determinada ação dentro de uma organização não são aproveitadas, seja pelo desconhecimento desta ou até mesmo por um deficiente mecanismo de busca e recuperação da “informação certa no momento certo à pessoa certa”. É, portanto, na organização dessas informações que a Gestão do Conhecimento atua.

Segundo [2], o uso de agentes como base tecnológica para ambientes de Gestão de Conhecimento é bem apropriado levando-se em consideração as características básicas de agentes - autonomia, habilidade social, reatividade e pró-atividade. Isso porque existem características típicas de ambientes de Gestão de Conhecimento – natureza distribuída do conhecimento da organização, baixa prioridade dos processos de Gestão do conhecimento, etc – que se relacionam com as características básicas de agentes inteligentes.

Um exemplo do uso da tecnologia de agentes inteligentes dentro do contexto de Gestão do Conhecimento, que será apresentado neste trabalho, é o *framework* K-Org [3]. O K-Org utiliza-se de agentes no auxílio da criação e manutenção da memória organizacional, mais especificamente no domínio do processo de desenvolvimento de software – fase modelagem de requisitos.

Nesse *framework* existe uma biblioteca de agentes especializados (K-Agts) que executam tarefas específicas, sendo reutilizados e contextualizados de acordo com cada domínio definido. Esta especialização de agentes está relacionada com os tipos de comportamento ou ações que cada agente executa.

O objetivo desse trabalho é ampliar essa biblioteca de K-Agts do K-Org através da geração de comportamentos que invocam Serviços *Web* (*Web Services*), o que resulta em um maior poder de expansão desta biblioteca visto que qualquer Serviço *Web* poderá resultar em um comportamento.

Assim, no decorrer do artigo será possível entender como ocorre essa geração de comportamentos e ter uma noção do poder de ampliação que a biblioteca de comportamentos de K-Agts do K-Org adquiriu com esses comportamentos que invocam Serviços *Web*.

2. Serviço Web

Os Serviços *Web* surgiram primeiramente através de um consórcio de empresas, entre elas Microsoft e IBM, até tornar-se hoje um padrão do W3C. De maneira bem simples, um Serviço *Web* nada mais é que um meio prático para que aplicações se comuniquem através da Internet.

Para que essa interoperabilidade seja possível foi criado o SOAP (Simple Object Access Protocol), o protocolo básico para a construção dos Serviços *Web*. Baseado em XML e nos protocolos de transporte da Internet (notadamente HTTP), essa tecnologia apresenta grande simplicidade e padronização. Assim, qualquer plataforma que interprete o protocolo HTTP e manipule XML poderá utilizar-se dos Serviços *Web*.

O WSDL (*Web Service Description Language*) é responsável pela definição ou descrição do serviço, de forma que nesse documento são informados quais métodos e parâmetros existem para tal serviço, além de como fazer uma requisição. A partir desse documento é possível iniciar a implementação de qualquer serviço, utilizando-se de preferência linguagens de programação voltadas para aplicações na Internet.

Para a facilitação da invocação dos Serviços *Web*, utilizamos nesse trabalho o WSIF [4]. O WSIF (*Web Service Invocation Framework*) é um *framework* que implementa a interface (em JAVA) que realiza a invocação de um serviço. Como vantagens do seu uso, está o fato dela abstrair a necessidade de se trabalhar diretamente com protocolos de acesso. O desenvolvedor que usa o WSIF não precisa se preocupar em como está implementado o serviço *web* e nem como é acessado. Assim o WSIF é flexível ao máximo, facilitando o desenvolvimento de aplicações sem uma maior preocupação com detalhes de invocação de métodos.

3. Framework K-Org

O K-Org é uma metodologia voltada para a análise de organizações e tem o propósito de modelar o conhecimento destas, mantendo as informações numa espécie de memória e tendo como objetivo a identificação de agentes que estejam envolvidos num domínio. Nessa metodologia existem fases de criação de modelos e para esses modelos são identificadas classes de agentes, denominadas K-Agts ou agentes de conhecimento. Os K-Agts irão oferecer suporte para as informações, pois incorporam tecnologia para atender ao que é requisitado no modelo.

O *framework* K-Org é um ambiente automatizado que ajuda na construção destes modelos e na geração dos K-Agts mapeados. Os K-Agts gerados são agrupados em um sistema multiagentes e assim proverão conhecimento para um domínio. Na sua estrutura tem-se a interface gráfica, onde há o registro dos modelos e a associação dos K-Agts com cada modelo. A base de dados é o conteúdo dos modelos e a definição dos K-Agts correspondentes aos modelos. O construtor de K-Agts é um “agente principal” que gera agentes a partir do que está definido na base de dados. Esses K-Agts são gerados em três classes, podendo ser um agente de domínio, que é o K-Agt do domínio estudado, um agente de controle que gerencia atividades e o agente de tecnologia, que são especializados na aplicação de tecnologias úteis para que as atividades dos agentes de domínio e de controle sejam efetuadas.

No caso deste trabalho, nosso foco é aumentar o número de K-Agts de tecnologia através da geração de comportamentos que invocam Serviços *Web* já existentes e disponíveis na rede. Assim, caso exista algum Serviço *Web* cuja funcionalidade seja de interesse para um domínio específico em que o K-Org esteja sendo aplicado, basta que seja gerado um comportamento que invoque este Serviço *Web*, e, assim, este novo comportamento com a funcionalidade deste Serviço *Web* ficará disponível na biblioteca de comportamentos do K-Org.

4. Implementação

A explicação de como acessar um serviço *web*, com a finalidade e as ferramentas já definidas neste artigo, será feita com base em um serviço disponível do GOOGLE, *website* conhecido por oferecer serviços de busca.

4.1 Criação de comportamentos

A figura 1 demonstra como o usuário deve proceder para gerar ou configurar um comportamento no K-Org. O primeiro passo é informar o endereço que contém o documento WSDL, ou seja, onde está descrito tal serviço *web*. Após isso é possível a análise do documento, através do botão Analisa, onde, após isso, é mostrado ao usuário uma árvore hierarquizada de todas as operações e parâmetros existentes para o serviço. No item três da figura, vemos por exemplo, as três operações disponíveis do Serviço *Web* do Google, o *doGetCachedPage*, o *doSpellingSuggestion*, o *doGoogleSearch*. A primeira retorna páginas armazenadas em *cache*, nos servidores do Google, de algum *site* específico definido pelo usuário. A segunda oferece um serviço de correção de palavras e a terceira oferece um serviço de busca em *sites* de toda a Internet através de palavras-chave. Em cada operação pode ser visualizado o seu tipo, “Request_Response” por exemplo, bem como os argumentos ou parâmetros requeridos para cada operação. No caso da figura 1, por exemplo, os argumentos do método *doGetCachedPage* são uma *URL* e uma *key*, que corresponde a uma senha para acessar o serviço.

Para a geração do comportamento, o usuário deve, primeiramente, escolher qual operação do serviço o comportamento irá utilizar. As operações selecionadas são listadas no item 4 da figura 1. Após isso, o comportamento pode ser gerado através do botão Gerar Comportamento, no quinto item, onde é gerada uma classe de comportamento com os métodos do Serviço *Web* selecionados pelo usuário. A partir desse momento, essa classe é cadastrada na biblioteca de comportamentos do K-Org, ficando disponível para que seja utilizada em qualquer K-Agt que necessitar das operações deste Serviço *Web*.



Figura 1 – Configuração de comportamentos

4.2 Criação de Agentes

Depois de gerar um comportamento, o usuário pode criar um K-Agt e associá-lo com um ou mais comportamentos disponíveis na biblioteca de comportamentos do K-Org.. A figura 2 mostra a tela de configuração de agentes do K-Org. Primeiro é preciso definir um nome para o agente e, opcionalmente, também pode ser definida alguma descrição das funcionalidades que o agente irá dispor. Além disso, a condição ou evento que irá disparar as ações do agente deve ser selecionado dentre os existentes. Nota-se também que existe uma lista ou biblioteca de comportamentos disponíveis, do lado direito da figura. Essa lista é incrementada a cada comportamento gerado e a partir dela pode-se associar agente(s) e comportamento(s). No exemplo da figura 2, o comportamento do agente KagtBuscaCacheGoogle é o BuscaCacheGoogle (*doGetCachedPage*). Por fim, com os argumentos definidos, o usuário gera o arquivo de configuração deste agente. Esse arquivo é um XML que tem as informações recém configuradas pelo usuário. No lado esquerdo da figura 2, o arquivo gerado é listado, podendo ser testado previamente. Esse agente com comportamento passa a ser visto pelo K-Org como um agente de tecnologia pelo fato de ter uma função ou especialização, que nesse exemplo é o acesso a um serviço *web* do Google.

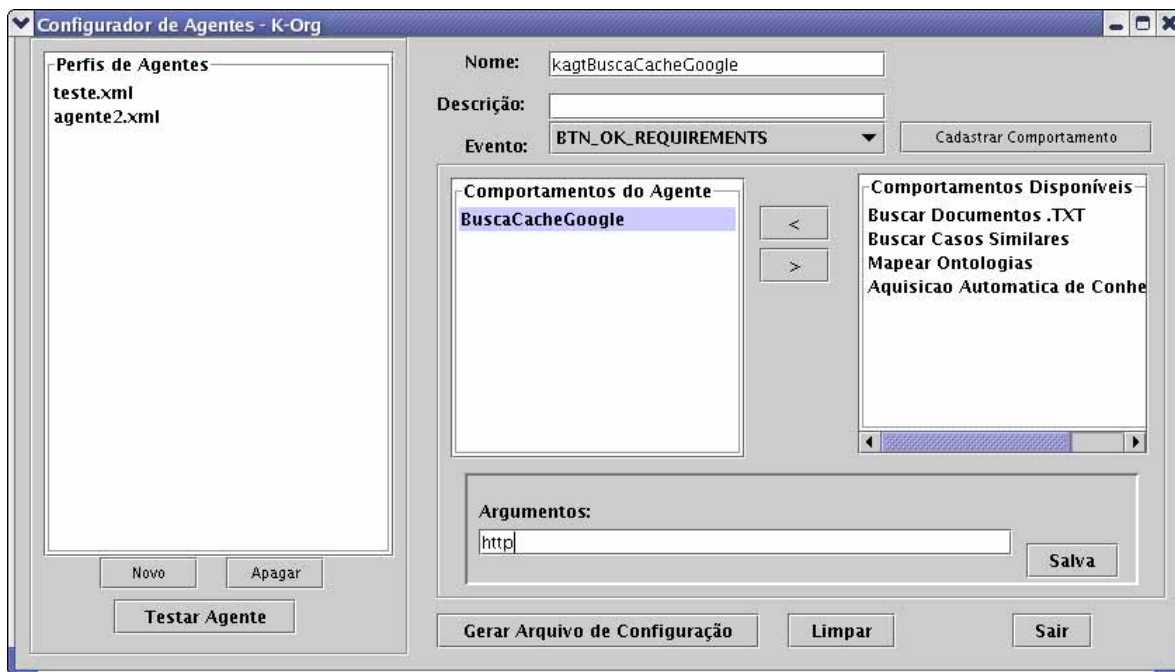


Figura 2 – Criação de agentes e associação de comportamentos

4.3 Invocação do Serviço Web

A figura 3 mostra a invocação do serviço *web* previamente definido. Relembrando que, após a configuração do serviço *web* e a seleção das operações que serão utilizadas, uma classe que representa o comportamento do agente é gerada. Uma vez que essa classe existe, ela ficará disponibilizada na biblioteca de comportamentos do K-Org. Após a criação de um agente que utilize esse comportamento, este ficará ativo no ambiente aguardando o evento em que será requisitado.

A ação do K-Agt ocorrerá no momento em que seu evento configurado é disparado (passo 1 ocorre quando evento é disparado). Quando o evento acontece, o comportamento do K-Agt passará os parâmetros, que foram definidos pelo usuário, para uma outra classe, chamada de classe de invocação do *framework* WSIF (passo 1). Essa classe de invocação, se comunica com o serviço *web* definido no comportamento (passo 2), que devolve o resultado correspondente com o

tipo de serviço requerido (passo 3), para a classe de invocação. Essa resposta é retornada ao K-Agt (passo 4), que exibirá o resultado da requisição (passo 5).



Figura 3 – A invocação e a resposta do serviço

5. Conclusão

Pelo que foi exposto é possível concluir que a tecnologia de Serviços *Web* oferece um grande número de possíveis aplicações para diversos ramos, basta saber usá-las da maneira que melhor se aplica às nossas necessidades, tanto pessoais como profissionais.

Como a meta estabelecida neste trabalho era enriquecer o *framework* K-Org, mais especificamente a biblioteca de comportamentos para agentes, conclui-se que o objetivo foi atingido através de uma implementação que, utilizando-se de tecnologias e conceitos atuais, possibilitou a associação de comportamentos de agentes a Serviços *Web*. Assim, como qualquer Serviço *Web* possa vir a tornar-se um comportamento de um K-Agt do K-Org, podemos perceber que o poder de ampliação dessa biblioteca de comportamentos cresce consideravelmente.

Além disso, pode-se dizer que a utilização de Serviços *Web* em sistemas multiagentes é vantajosa pois permite o reuso de aplicações existentes, sem maiores preocupações com as plataformas utilizadas. Isto mostra que cada vez mais as aplicações buscam padrões de desenvolvimento a fim de possibilitar o seu reuso por outros tipos de aplicações e tecnologias.

6. Referências

Strike Iron: <http://www.strikeiron.com>. 28/06/2004.

Xmethods: <http://www.xmethods.com>. 28/06/2004.

WSDL: <http://www.w3.org/TR/>. 27/06/2004.

JADE: <http://jade.cselt.it>. 25/06/2004.

WSDL to Agent: <http://sas.ilab.sztaki.hu:8080/wsdl2agent/index.html>. 28/06/2004.

[1] Prahalad, C.K.; Hamel, Gary. **The Core Competence of the corporation**. Junho/1990.

[2] Van Elst, Ludger. **Towards Agent-Mediated Knowledge Management**.

<http://www.dfki.uni-kl.de/~elst/papers/lnai29260001.pdf>. 30/06/2004.

[3] Salles, Lina. **Modeling Multi-Agent Systems for Knowledge Management in Software Development Process**. Abril/2004.

[4] Apache WSIF: <http://ws.apache.org/wsif/index.html>. 28/06/2004.