Geração Automatizada de Portais Web para Grades Computacionais

Leonardo G. Früh[1], Andrea S. Charão[1]

[1]Laboratório de Sistemas de Computação
Universidade Federal de Santa Maria (UFSM) - Santa Maria, RS - Brasil
{lgfruh,andrea}@inf.ufsm.br

Resumo. *Este artigo apresenta um estudo da possibilidade de criação de um aplicativo para geração automatizada de portlets para o framework GridSphere. Este estudo é importante para facilitar a criação de aplicativos executados em portais de grade, algo muito popular atualmente.*

1. Introdução

Grades computacionais constituem uma importante plataforma de execução distribuída, capaz de agregar recursos geograficamente dispersos para viabilizar aplicações computacionalmente intensivas [Fox, G. Ed. Wiley 2003]. Para facilitar a interação dos usuários finais com a grade, geralmente utiliza-se um portal Web através do qual se acessa serviços essenciais, como a configuração e a submissão de tarefas à grade, o histórico de tarefas processadas, entre outros.

A construção de portais Web dedicados a grades computacionais pode ser feita utilizando-se *frameworks* amplamente disponíveis [Wang, X. D.; Yang, X.; Allan, R. 2006]. Mesmo com a existência destas ferramentas, administradores geralmente dedicam um tempo considerável à integração de novas aplicações à grade e ao portal. Além disso, a maioria dos portais para grades não acompanha os avanços da Web 2.0, limitando a interatividade e a colaboração [Pierce, Marlon; Fox, G.; Choi, Jong; Guo, Zhenhua; Gao, Xiaoming; Ma, Yu 2009]. Neste contexto, este trabalho apresenta uma solução para se abreviar e simplificar o processo de criação e integração de novas aplicações aos portais Web. No restante deste artigo, apresenta-se o *framework* utilizado para este projeto e suas principais características, os avanços realizados até o momento, e, ao final, o que ainda precisa ser feito para se alcançar os objetivos estipulados inicialmente.

2. O Framework GridSphere

O *framework* escolhido para este projeto foi o GridSphere [GridSphere 2002], por sua extensibilidade e por ser amplamente utilizado mundialmente, principalmente na Europa e em grades brasileiras. Pode-se citar o UK E-Science Program, o D-Grid, o P-GRADE e o K*Grid como exemplos de grandes projetos que o utilizam fora do Brasil [Guerra 2006].

O GridSphere é um *framework* utilizado para a criação de portais Web baseados em *portlets*. A API (*Application Programming Interface*) de desenvolvimento de *portlets* do GridSphere é totalmente compatível com a especificação JSR-168, a qual padroniza o desenvolvimento deste tipo de aplicativo Web.

O pacote padrão do *framework* inclui uma biblioteca de *tags* (*GridSphere User Interface tag library*) para facilitar a criação de interfaces para os *portlets*, suporte a utilização do *framework* de persistência de dados Hibernate, suporte ao desenvolvimento de *portlets* multi-idiomas e um conjunto de *portlets* (GridSphere *core portlets*) que formam a estrutura básica do portal GridSphere. Estes *portlets* oferecem as funcionalidades administrativas básicas do portal, como: gerenciamento de usuários e grupos de usuários, gerenciamento dos *portlets* em execução no portal, personalização do layout do portal, implementação das ações de *login* e *logout*, e seleção do idioma do *portlet*.

2.1. Arquitetura do GridSphere

O *framework* é executado em um servidor Tomcat, o qual suporta aplicações Java para Web, dentre elas, os *portlets* desenvolvidos a partir da especificação JSR-168.

Os principais componentes do GridSphere são o portal, os *portlets* e o *portlet container*.

O portal é a infra-estrutura que agrega o conjunto de *portlets* e permite que os usuários personalizem sua estrutura, selecionando os *portlets* de interesse e editando suas configurações. O portal também permite a atribuição de papéis e permissões a grupos de usuários.

Os *portlets* são os aplicativos Web que constituem o portal. Eles podem fornecer informações ou prover recursos, e são gerenciados pelo *portlet container*.

O *portlet container* é a entidade que processa as requisições realizadas para cada *portlet*, invoca os métodos adequados para o tipo de requisição e gera dinamicamente o conteúdo que será mostrado ao cliente.

2.2. Criação de Portlets no GridSphere

Para a criação de um novo *portlet*, primeiramente é necessário executar o comando *ant new-project* do GridSphere para a criação da estrutura de arquivos e diretórios padrão da JSR-168.

Um *portlet* pode ser implementado como uma subclasse de *GenericPortlet* ou *ActionPortlet* e ter seus métodos customizados conforme as necessidades do usuário. Esta parte constitui os arquivos-fontes Java do *portlet*.

A interface do *portlet* pode ser totalmente desenvolvida no código Java, mas para uma melhor modularização são utilizadas páginas JSP (*Java Server Pages*).

Conforme a especificação JSR-168, também é necessária a criação de um arquivo descritor do *portlet*, chamado *portlet.xml*.

Os passos descritos acima são o mínimo que deve ser feito para se criar um novo *portlet*. Porém, quando se adiciona um novo *portlet* ao portal são necessários alguns passos adicionais para que ele seja configurado corretamente, inicializado e passe a fazer parte do portal.

3. Requisitos para Automatização

Como a única maneira de se definir com exatidão o comportamento de um *portlet* é através de um algoritmo, o que impossibilita seu desenvolvimento por usuários sem conhecimento técnico apropriado, esse projeto focaliza-se na geração de um tipo específico de aplicativos: *portlets* que adicionam *jobs* a serem executados na grade.

Para a automatização do processo de criação de *portlets* buscou-se uma abordagem que simule os passos realizados para a criação manual dos mesmos. Ou seja, primeiramente cria-se um *portlet* que captura as informações gerais sobre o *portlet* a ser criado, de forma que o comportamento e interface do mesmo possam ser definidos.

Tendo essas informações, scripts podem ser utilizados para a criação dos arquivos-fontes Java. Como os comportamentos dos *portlets* são semelhantes (adicionar *jobs* para serem executados) a principal diferença nessa parte será a implementação dos métodos de renderização, pois cada aplicação pode ter diferentes maneiras de apresentar os resultados da execução dos *jobs*, bem como parâmetros variados no modo de edição do *portlet*.

Scripts também podem ser utilizados para criar as páginas JSP. Nesse caso, as principais modificações a serem realizadas na criação de cada *portlet* estão relacionadas à estrutura da interface: quantos e quais componentes irão compor a interface do *portlet*.

Após a criação dos arquivos Java e JSP, ainda é necessário editar os arquivos de configuração e descrição do *portlet* para que ele seja devidamente integrado ao portal. Primeiramente deve-se editar o arquivo descritor *portlet.xml*, adicionando informações como o nome do *portlet* e o pacote do código-fonte Java. Para que o *portlet* seja adicionado à lista de *portlets* em execução do portal, deve-se editar o arquivo *loggedin.xml*. Isso pode ser feito de duas formas: pelo usuário através dos *core portlets* do GridSphere ou de forma automática pela própria aplicação em questão. Enquanto a segunda forma proporciona maior velocidade ao processo, a primeira oferece ao usuário um maior poder de customização do portal, já que o *core portlet* que adiciona aplicativos ao portal, também oferece a opção de personalização do layout do mesmo. Além disso, o usuário pode desejar apenas a criação do *portlet* em um pacote *.war* para adicioná-lo a qualquer portal GridSphere e não especificamente o portal que está sendo utilizado para a execução da aplicação geradora de *portlets.* Neste momento, o processo de geração pode tomar dois caminhos distintos: criar e integrar o *portlet* ao portal sendo utilizado, ou criar o *portlet* e deixá-lo disponível ao usuário como um pacote *.war*.

Para que o primeiro caminho seja realizado deve-se executar o comando de criação de novo projeto do GridSphere, adicionar os arquivos gerados até o momento aos seus respectivos diretórios, editar o arquivo *loggedin.xml* através de um script, fazer o *deploy* do novo *portlet* ao portal e reiniciar o servidor Tomcat.

Para que o segundo caminho seja realizado deve-se apenas criar um pacote *.war* com os arquivos gerados nas fases anteriores.

4. Avaliação

Como o presente projeto ainda está em fase de desenvolvimento, existem questões a serem resolvidas. Uma delas é como e quais *middlewares* serão suportados pela aplicação. Sabe-se que cada *middleware* requer uma série de comandos e/ou arquivos de

descrição pré-definidos para que *jobs* sejam adicionados a grade. Por isso ainda deve-se estudar as peculiaridades de diversos *middlewares* para determinar se é possível suportar qualquer *middleware* ou se é necessário escolher um conjunto de *middlewares* que serão suportados pela aplicação.

Outra questão ainda não resolvida é como aprimorar a interatividade e a colaboração em portais de grade. Para solucionar esta questão deve-se estudar componentes interativos que poderiam ser adicionados a interface dos *portlets* para auxiliar na colaboração entre os usuários do portal.

Um último ponto a ser ressaltado é que, com o uso de simples scripts para a criação dos arquivos, deverá ser feita uma versão do software para cada sistema operacional. Para resolver esse problema cogita-se o uso de alguma ferramenta ou API para a geração de arquivos. Como exemplo tem-se a JET (*Java Emitter Template*) [Viera 2008].

5. Conclusão

A partir dos requisitos apresentados, pode-se ver que o processo de automatização da criação de *portlets* pode ser feito apenas automatizando os processos que são feitos manualmente. Obviamente, perde-se algum nível de customização para facilitar e agilizar a geração dos aplicativos. A principal vantagem da criação deste aplicativo gerador de *portlets* é que qualquer usuário poderá criar seus aplicativos e adicioná-los aos seus portais sem que seja necessário o conhecimento de técnicas avançadas de programação.

Como também foi visto, este trabalho ainda está em desenvolvimento, necessitando a continuação dos estudos sendo realizados até o momento para sua finalização.

Referências

Fox, G.; Berman, F.; Hey, T. Grid Computing: Making the Global Infrastructure a Reality, Fox, G. Ed. Wiley, 2003.

GridSphere. www.gridsphere.org. Europa, 2002. Disponível em: <http://www.gridsphere.org/>. Acesso em: 2 jan. 2010.

Guerra, E. L. **InGriDE**: Um Ambiente Integrado de Desenvolvimento para Computação em Grade. 2006. 55f. Dissertação (Mestrado em Ciência da Computação) – Universidade de São Paulo, São Paulo, 2006.

Pierce, Marlon; Fox, G.; Choi, Jong; Guo, Zhenhua; Gao, Xiaoming; Ma, Yu. Using Web 2.0 for Scientific Applications and Scientific Communities. Concurrency and Computation: Practice and Experience, v. 21, n.5, Apr. 2009.

Viera, M. A. **Um portlet genérico para construção de portais web para grades computacionais utilizando o framework GridSphere.** 2008. 37f. Monografia (Trabalho de Graduação) – Universidade Federal de Santa Maria, Santa Maria, 2008.

Wang, X. D.; Yang, X.; Allan, R. Top Ten Questions to Design a Successful Grid Portal. Proceedings of the Second International Conference on Semantics, Knowledge, and Grid, 2006.