

GRADEp-SC: Sensibilidade ao Contexto no Middleware GRADEp

Fernando A. Afonso¹, João L. B. Lopes¹, Adenauer C. Yamin¹, Luiz A. M. Palazzo¹

¹Escola de Informática – Universidade Católica de Pelotas (UCPEL)

afonso, joaolopes, adenauer, lpalazzo@ucpel.tche.br

Resumo. *Este trabalho tem como tema central uma proposta baseada em ontologias para o mecanismo de sensibilidade ao contexto do projeto GRADEp. O projeto GRADEp tem por finalidade definir a arquitetura para um ambiente de execução destinado às aplicações da Computação Perversiva, implementado na forma de um middleware.*

1. Introdução

A Computação Perversiva é a proposta de um novo paradigma que permite ao usuário o acesso a seu ambiente computacional independente de localização, tempo e equipamento. A proposta da computação perversiva pode ser construída pela integração da computação móvel, computação em grade e computação sensível ao contexto. Em um ambiente de computação perversiva, os dispositivos, serviços e agentes devem ser conscientes de seus contextos e automaticamente adaptar-se às suas mudanças, isso caracteriza a sensibilidade ao contexto [Yamin 2004].

A computação sensível ao contexto vem despertando muita atenção dos pesquisadores desde sua proposta [Schilit and Theimer 1994]. Diversos sistemas de sensibilidade ao contexto foram desenvolvidos para demonstrar a utilidade desta tecnologia, entretanto serviços de sensibilidade ao contexto nunca foram amplamente disponibilizados para os usuários e a construção de sistemas sensíveis ao contexto é ainda uma tarefa complexa e altamente consumidora de tempo, devido à falta de uma apropriada infraestrutura para suporte.

O suporte à adaptação no GRADEp está associado à operação do subsistema de reconhecimento de contexto e adaptação. Este subsistema inclui serviços que tratam desde a extração da informação bruta sobre as características dinâmicas e estáticas dos recursos que compõem o ambiente perversivo, passando pela identificação em alto nível dos elementos de contexto, até o disparo das ações de adaptação em reação a modificações no estado de tais elementos de contexto [Yamin 2004]. Neste subsistema, estão sendo inseridas novas funcionalidades através da proposição de um servidor de contexto baseado em ontologias [Lopes 2007]. Neste sentido, o presente trabalho pretende conceber uma versão baseada em ontologias para o mecanismo de sensibilidade ao contexto do GRADEp.

Deste perfil emergente para os ambientes computacionais modernos, decorre a motivação central deste trabalho, que é a qualificação do mecanismo para suporte à sensibilidade ao contexto do middleware GRADEp.

Mark Weiser [Weiser 1991], em seu artigo *The Computer for the 21st Century*, resume o que é esperado da computação perversiva (também chamada de computação

ubíqua ou de ubicomp): acesso do usuário ao ambiente computacional, de todo lugar e a todo momento, por meio de qualquer dispositivo.

A dificuldade se encontra em como desenvolver aplicativos que irão continuamente se adaptar ao ambiente e continuar funcionando, a medida que as pessoas se movem ou trocam de dispositivos.

As aplicações pervasivas precisam de um *middleware* para interfacear entre muitos dispositivos diferentes e a aplicação do usuário final [Saha and MUKHERJEE 2003]. O objetivo é esconder a complexidade do ambiente, isolando aplicações do gerenciamento explícito de protocolos, acesso distribuído a memória, replicação de dados, falhas de comunicação, etc. Um *middleware* também resolve problemas de heterogeneidade relacionados às arquiteturas, sistemas operacionais, tecnologias de redes e até mesmo de linguagens de programação, promovendo a interoperação entre esses componentes.

A computação sensível ao contexto é um paradigma computacional que se propõe a permitir que as aplicações tenham acesso e tirem proveito de informações que digam respeito às computações que realizam, buscando otimizar seu processamento. [Chen and KOTZ 2000].

Além de lidar com entradas explícitas, a computação sensível ao contexto também considera informação de contexto capturadas por meio de sensores, ou seja, entradas implícitas, tais como: localização, recursos e infra-estrutura disponíveis, preferências do usuário, atividade do usuário, número de dispositivos, tipo de dispositivo, carga computacional [Chen and KOTZ 2000].

As ontologias vêm sendo utilizadas por várias áreas da Ciência da Computação, principalmente com o intuito de dotar os sistemas de meta-conhecimento. A utilização de ontologias para descrição semântica de um determinado vocabulário proporciona um entendimento amplo das características e propriedades das classes pertencentes a um domínio, assim como seus relacionamentos e restrições e, por meio de regras de inferência, é possível derivar novos fatos baseando-se em fatos existentes [Gruber 1993].

2. GRADEp

O projeto GRADEp tem por finalidade definir a arquitetura para um ambiente de execução destinado às aplicações da Computação Pervasiva, implementado na forma de um *middleware* no qual o suporte à execução deve permitir que tanto as aplicações como ele próprio utilizem informações de contexto na gerência da adaptação de seus aspectos funcionais e não-funcionais. A premissa *sigame* das aplicações *pervasivas* deverá ser suportada, garantindo a execução da aplicação do usuário em qualquer tempo e lugar.

As aplicações-alvo são distribuídas, adaptativas ao contexto em que executam e compreendem a mobilidade lógica e a física. Sua proposta é modelar um *middleware* que faculte uma estratégia colaborativa com a aplicação nos procedimentos de adaptação. [Yamin 2004][Augustin et al. 2006].

3. GRADEp-SC

O sistema desenvolvido foi escrito na linguagem de programação Java, e foram utilizadas diversas bibliotecas auxiliares para dar suporte as diferentes tecnologias empregadas no

projeto. Bibliotecas as quais estão listadas a seguir:

- GRADEp - biblioteca responsável por fornecer suporte aos serviços pervasivos.
- Jena - API responsável por permitir a manipulação de ontologias dentro de aplicações Java.
- ARQ - biblioteca responsável por fornecer suporte a consultas SPARQL em aplicações baseadas na API Jena.
- DOM - biblioteca responsável por permitir a manipulação de arquivos XML em aplicações Java.

Foram utilizados serviços de comunicação do GRADEp na implementação do servidor com o intuito de abstrair aspectos de baixo nível relativos ao tratamento das comunicações em rede. A classe principal do servidor de contexto dispara uma *thread* responsável por utilizar esses serviços de comunicação para aguardar as atualizações de contexto dos nodos. O servidor de contexto implementado mantém uma base de dados ontológica com informações sobre o estado do ambiente pervasivo.

Anexado ao servidor de contexto foi desenvolvido um serviço de consultas o qual possibilita a realização de consultas sobre a ontologia utilizando a linguagem SPARQL. Algumas consultas envolvendo propriedades das classes da ontologia encontram-se parametrizadas, tais como: atividade do nodo, memória livre, CPU livre, espaço em disco. Além disso, também é possível realizar qualquer consulta válida na linguagem SPARQL. A linguagem de consulta SPARQL é utilizada através da biblioteca ARQ da API Jena.

Um aplicativo desenvolvido para testar os serviços foi o Inspetor de Contexto, o qual permite inspecionar o estado atual do contexto celular executando as consultas SPARQL pré-definidas no serviço de consulta. As consultas pré-definidas no serviço de consultas se basearam nas consultas utilizadas por esse aplicativo. Nessa aplicação é possível visualizar o estado geral da célula e o estado específico de cada nodo.

Para efetuar a coleta das informações de contexto dos nodos foram desenvolvidos os seguintes módulos: gradepSC-Coletor, gradepSC-Monitor e gradepSC-Sensor.

- gradepSC-Sensor: realiza a ativação dos sensores responsáveis pela obtenção dos dados de contexto. Os sensores são divididos em dois tipos, os que monitoram dados estáticos e os que monitoram dados dinâmicos.
- gradepSC-Monitor: os monitores são responsáveis por agrupar as informações de seus sensores monitorados e as enviarem para o coletor. Foram criados dois tipos de monitor, um responsável por agrupar as informações dos sensores estáticos, fazendo a atualização dos dados no módulo gradepSC-Coletor somente quando o aplicativo é executado, e outro responsável por agrupar as informações dos sensores dinâmicos, fazendo a atualização do Coletor sempre que ocorrerem mudanças significativas nos dados.
- gradepSC-Coletor: o coletor é responsável por formar a ontologia do estado atual do nodo e a enviar para o servidor de contexto. Ele a compõe agrupando as informações recebidas dos monitores e a envia utilizando os serviços de comunicação do GRADEp.

Esses módulos formam um serviço chamado Collector. Cada nodo tem seu próprio Collector instalado. Para obter as configurações da célula, necessárias para o Collector

funcionar a aplicação utiliza um *parser* XML chamado Dom para acessar o arquivo XML que armazena as configurações do nodo, sendo assim, o usuário não precisa fazer nenhum tipo de configuração para o Collector funcionar.

Para testar o sistema desenvolvido foi utilizada uma aplicação que realiza o cálculo do Pi através do método de Monte Carlo de forma distribuída. Essa aplicação dispara consultas SPARQL ao serviço de consultas para obter os computadores com maior poder computacional disponíveis para a realização dos cálculos. O teste obteve êxito.

4. Considerações Finais

O sistema desenvolvido apresentou resultados promissores nos testes realizados sobre uma célula do GRADEp, sendo possível obter informações de contexto diversas a partir do serviço de consultas desenvolvido.

As características de reuso e compartilhamento de conhecimento sobre o ambiente pervasivo se potencializam, uma vez que quando se utiliza ontologias para representação do mesmo o conhecimento se encontra padronizado e expresso em uma linguagem formal. Desta forma, se torna factível a leitura e interpretação da ontologia por outros domínios.

A utilização de ontologias também permite o desenvolvimento de um mecanismo de dedução sobre a base ontológica do ambiente pervasivo e a criação de heurísticas de escalonamento que utilizem o servidor de contexto desenvolvido.

References

- Augustin, I., Yamin, A. C., da Silva, L. C., Real, R. A., Frainer, G., and Geyer, C. F. R. (2006). Isadapt: abstractions and tools for designing general-purpose pervasive applications: Experiences with auto-adaptive and reconfigurable systems. *Softw. Pract. Exper.*, 36(11‐12):1231–1256.
- Chen, G. and KOTZ, D. (2000). A survey of context-aware mobile computing research.
- Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, pages 199–220.
- Lopes, J. L. B. (2007). Exehda-on: Uma abordagem baseada em ontologias para sensibilidade ao contexto na computação pervasiva. Technical report 2007-01, Universidade Católica de Pelotas - Programa de Pós-Graduação em Informática.
- Saha, D. and MUKHERJEE, A. (2003). Pervasive computing: a paradigm for the 21st century. 36(3):25–31.
- Schilit, B. and Theimer, M. (1994). Disseminating active map information to mobile hosts. *IEEE Network*.
- Weiser, M. (1991). The computer for the 21st century. *Scientific American*, 3(265):94–104.
- Yamin, A. (2004). *Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva*. Tese (doutorado em ciência da computação), Instituto de Informática, UFRGS, Porto Alegre, RS.