

389640

# ISAMadapt: ambiente de desenvolvimento de aplicações para o paradigma da *Pervasive Computing*

Iara Augustin<sup>1</sup>  
Adenauer Correa Yamin<sup>2</sup>  
Rodrigo Araujo Real<sup>3</sup>  
Luciano Cavalheiro da Silva<sup>4</sup>  
Jorge Luis Victória Barbosa<sup>5</sup>  
Cláudio Fernando Resin Geyer<sup>6</sup>

## Resumo

ISAMadapt é um ambiente de desenvolvimento de aplicações pervasivas com comportamento adaptativo ao contexto em que executam. As principais abstrações para o projeto de aplicações incluem: contexto, comportamentos alternativos, mecanismos e políticas de adaptação. A aplicação torna-se consciente de seu contexto, e adapta-se a ele, através de um processo que envolve duas fases: concepção e execução. Comandos relativos às abstrações são adicionados à linguagem-base Holo, e o ambiente de execução EXEHDA implementa a funcionalidade dinâmica dessas abstrações.

## 1 Introdução

À medida que computadores tornam-se mais portáteis, as pessoas desejam acessar informações em qualquer lugar, a qualquer hora, com os dispositivos pessoais que carregam junto a si ou com dispositivos que estejam disponíveis no local em que se encontram. Este novo cenário computacional liberta os usuários da localidade de seu ambiente computacional, e requer que os elementos computacionais possam estar espalhados e acessíveis em toda a área de deslocamento do usuário. Este cenário está sendo conhecido como *Pervasive Computing*

<sup>1</sup>august@inf.ufrgs.br bolsista CAPES/PCIDT

<sup>2</sup>adenauer@inf.ufrgs.br bolsista CAPES/PCIDT

<sup>3</sup>rreal@inf.ufrgs.br bolsista CAPES

<sup>4</sup>lucc@inf.ufrgs.br bolsista CNPq

<sup>5</sup>barbosa@inf.ufrgs.br

<sup>6</sup>geyer@inf.ufrgs.br

(SATYANARAYANAN, 2001). Os paradigmas de programação tradicionais que assumem um ambiente de execução estacionário, como o paradigma distribuído, não são mais apropriados para projetar aplicações neste cenário (AUGUSTIN, 2002).

O paradigma da *Pervasive Computing* impõe requisitos de operação aos sistemas, muitos deles novos (AUGUSTIN, 2001). Entre eles estão a capacidade de:

- suportar flutuações na disponibilidade e acessibilidade aos recursos;
- operar em vários níveis de conectividade à rede;
- executar em diversas plataformas;
- possibilitar que o ambiente do usuário e de aplicações sejam instalados automaticamente pelo sistema;
- permitir o acesso ao ambiente do usuário e a suas aplicações, independente do lugar e do dispositivo de acesso;
- permitir que a aplicação em execução acompanhe o usuário em seu deslocamento (*follow-me applications*).

Assim, uma aplicação consciente do contexto deve ser projetada com base em funcionalidades que se ajustem ao ambiente corrente, e estas são selecionadas automaticamente pelo sistema de gerenciamento das aplicações.

Para tratar as características da *Pervasive Computing*, heterogeneidade e dinamicidade, a adaptabilidade é um conceito chave na arquitetura de software. Porém, os atuais modelos de software não acomodam a generalidade necessária a essa classe de aplicações (AUGUSTIN, 2002). O foco da pesquisa do **projeto ISAM** - Infra-estrutura de Suporte às Aplicações Móveis (ISAM, 2002) - é o estudo das questões envolvidas na modelagem e execução de aplicações móveis conscientes do contexto em um ambiente de *pervasive computing* (AUGUSTIN, 2002). Por questões de escopo, este texto aborda dentro da arquitetura ISAM somente a descrição das abstrações para expressar um comportamento adaptativo ao contexto sob a ótica da programação.

## 2 Abstrações ISAMadapt

A arquitetura ISAM é disponibilizada através de um ambiente de desenvolvimento de aplicações, ISAMadapt, e de um middleware, EXEHDA, que gerencia os aspectos da adaptação (YAMIN, 2003). Os aspectos da adaptação referem-se a três abstrações: contexto, adaptadores e políticas de adaptação. Cada uma dessas abstrações é implementada em duas visões: tempo de programação, e tempo de execução. Essas abstrações foram adicionadas a uma linguagem já existente: HoloLinguagem (BARBOSA, 2002).

## 2.1 Fase de Programação

Na fase de programação são codificados os vários aspectos da adaptação, com o uso do ambiente de desenvolvimento ISAMadapt. Durante esta fase são definidos os elementos de contexto de interesse da aplicação, codificados os comportamentos alternativos ajustados às condições ambientais, e selecionadas as políticas de adaptação.

### Declarando elementos de contexto

No ISAM, contexto é definido como *toda informação relevante para a aplicação e que pode ser obtida por esta* (AUGUSTIN, 2001). O programador explicitamente identifica as entidades e define seus atributos os quais integram o contexto da aplicação. Por exemplo, a entidade *host* poderá ter como elemento de contexto: ociosidade, carga computacional, poder computacional, número de *hosts*.

Em ISAMadapt, existem três formas de expressar consciência de contexto no código da aplicação: usando variáveis de ambiente, declarando os elementos de contexto de interesse através do comando *context* e definindo atributos dos elementos de contexto através do menu *context* do ambiente de desenvolvimento. Este último é usado para definir elementos de contexto mais elaborados, chamados de contexto especializado, cuja definição e interpretação de estado depende de cada aplicação. Neste caso, os estados do elemento de contexto podem ser em número variado, e expressos em nomes significativos (mnemônicos) definidos pelo programador. Por exemplo, para tipos de dispositivos: "celular", "palm", "normal"; para banda passante: "alta", "média", "baixa"; para localização: "casa", "escritório", "desconhecido", "outro". O comando *context* declara o uso de elementos de contexto nativos do ISAMadapt no código da aplicação.

### Expressando o comportamento adaptativo

No ISAMadapt são disponibilizadas várias formas de tornar o código consciente do contexto. Está a cargo do projetista da aplicação decidir qual a mais conveniente em cada caso.

Para tratar com elementos de contexto direto, onde serviços e recursos tornam-se disponíveis e indisponíveis, usa-se o comando *onContext*, associado à declaração *context*. Este comando é inserido em outros comandos da linguagem, como comandos de seleção (if), repetição (while), ou escrita (out). Por exemplo,

```
clone (worker, #worker_id) onContext host_idle;
```

A semântica deste comando é ficar bloqueado até receber a notificação do Servidor de Contexto (*ISAMcontextServer*) informando da disponibilidade de um *host*, quando criará um novo ente *worker*.

Na adaptação de código a definição funcional da aplicação não é alterada. Desta, alguns métodos comportam-se como funções genéricas, cujos comportamentos alternativos são baseados no estado do elemento de contexto de interesse da aplicação. Esta abstração é expressa pelo predicado *adaptive* junto ao nome do método no comando de chamada de método. Por exemplo, *adaptive display (argumentos)*. O método *display* será implementado por diferentes códigos de adaptadores, escritos em Holo, relativos às possibilidades de dispositivos de saída (por exemplo, "celular", "palm", "normal"), que integram o contexto de adaptação da aplicação.

O código da aplicação pode também fazer uso de comandos para expressar estratégias adaptativas independentes do contexto. Os comandos disponibilizados são:

- *move* - migração de um ente para dentro de outro;
- *clone* - criação e replicação de entes;
- *prefetch* - busca antecipada de arquivos;
- *disconnect* - desconexão da rede;
- *reconnect* - reconexão à rede;
- *install* - instala uma aplicação no destino;
- *reschedule* - permite ao programador controle sobre o escalonamento físico;
- *discovery* - descoberta de recurso ou serviço.

### Definindo políticas de adaptação

Políticas implementam o mecanismo de colaboração entre o sistema e a aplicação, e estão associadas à implementação dos mecanismos de adaptação disponibilizados. Referem-se às orientações da aplicação para a tomada de decisão do sistema, que controla o comportamento geral da aplicação. A principal vantagem desta abordagem é o equilíbrio entre transparência (uso de políticas default) e consciência da aplicação. As políticas podem ser de natureza geral, válidas para toda aplicação, ou específicas, válidas para componentes em particular. Os exemplos a seguir referem-se à desconexão: *DISCONNECT migrate nextServer* indica que antes da operação de desconexão ser realizada, as atividades no *host* devem ser migradas para o servidor mais próximo; *DISCONNECT cache* indica o chaveamento das operações de E/S para a cache local.

## 2.2 Fase de Execução

Cada abstração expressa no código da aplicação (contexto, adaptadores, mecanismos e políticas de adaptação) têm um componente correspondente no Sistema de Execução, o qual é responsável por implementar sua funcionalidade. O contexto relaciona-se com o servidor de reconhecimento de contexto, *ISAMcontextServer*; os adaptadores relacionam-se com o reconfigurador da *ISAMadaptEngine*; e as políticas relacionam-se com os mecanismos de adaptação implementados pelo EXEHDA.

## 3 Trabalhos Relacionados

Recentemente, começou a surgir alguns projetos que estão desenvolvendo a infra-estrutura para a produção de software para o paradigma da *Pervasive Computing*. Entre estes, citam-se o projeto Aura (GARLAN; STEENKISTE; SCHMERL, 2002), e o projeto Gaia (ROMAN, 2002). Ambos tem objetivos semelhantes ao projeto ISAM, porém com soluções diferentes. Sob a ótica da programação, alguns dos poucos suportes atuais para a programação de aplicações móveis de propósito-geral são fornecidos em termos de *toolkits* para detecção e notificação das alterações ambientais, deixando a cargo do programador a tarefa de definir o que fazer com essas informações, como em *Context Toolkit* (DEY, 2001). Considerando linguagens de programação, a computação pervasiva ainda não foi abordada adequadamente. Algumas propostas abordam adaptação específica a um elemento, como a linguagem Klaim (BETTINI; LORETI; PUGLIESE, 2001) que permite construções para projetar o comportamento dinâmico da conectividade, ou C2 style (BELLAVISTA, 1999) para mobilidade de código.

## 4 Conclusões

A *Pervasive Computing* é um novo paradigma que tem despertado muita atenção recentemente. A importância da adaptação neste ambiente é largamente reconhecida. Entretanto, adaptação dinâmica não é facilmente implementada. Hoje, as aplicações móveis que expressam comportamento adaptativo o fazem usando uma abordagem *ad-hoc*, específica da aplicação ou de uma classe de aplicações. Neste texto, apresentou-se a proposta ISAMadapt para tratamento de propósito-geral da adaptação, e concentrou-se nas abstrações de linguagem para expressar consciência do contexto. As abstrações para definir contexto, comportamento alternativo e políticas de adaptação, foram adicionadas à linguagem Holo, uma linguagem com mobilidade implícita e interação por multi-espacos de objetos.

## Referências

- AUGUSTIN, I. et al. Towards a taxonomy for mobile applications with adaptive behavior. In: (*PDCN02*). Innsbruck, Austria: International Association of Science and Technology for Development (IASTED), 2002.
- AUGUSTIN, I. et al. Requisitos para o projeto de aplicações móveis distribuídas. In: *VIII CACIC Congreso Argentino de Ciencias de la Computación*. Santa Cruz, Argentina: [s.n.], 2001.
- AUGUSTIN, I. et al. Isam - a software architecture for adaptive and distributed mobile applications. In: *VII ISCC 2002 IEEE SYMPOSIUM ON COMPUTERS AND COMMUNICATIONS*. New York: IEEE Press: [s.n.], 2002.
- BARBOSA, J. L. V. *Holoparadigma: Um Modelo Multiparadigma Orientado ao Desenvolvimento de Software Distribuído*. Tese (Tese (Doutorado em Ciência da Computação)) — Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002. 213p.
- BELLAVISTA, P. A secure and open mobile agent programming environment. In: PRESS, I. (Ed.). *ISADS '99*. [S.l.: s.n.], 1999.
- BETTINI, L.; LORETI, M.; PUGLIESE, R. Modeling node connectivity in dynamically evolving networks. In: ENTCS (Ed.). *CONCOORD International Workshop on Concurrency and Coordination*. [S.l.: s.n.], 2001. v. 54.
- DEY, A. Understanding and using context. *Personal and Ubiquitous Computing*, n. 5, 2001.
- GARLAN, D.; STEENKISTE, P.; SCHMERL, B. Project aura: Toward distraction-free pervasive computing. *IEEE Pervasive Computing*, v. 1, n. 3, september 2002. Disponível em: <[www.cs.cmu.edu/aura](http://www.cs.cmu.edu/aura)>.
- ISAM. *Projeto ISAM*. 2002. WWW. Disponível em <http://www.inf.ufrgs.br/~isam>. Acesso em dezembro 2002.
- ROMAN, M. Gaia: a middleware infrastructure to enable active spaces. *IEEE Pervasive Computing*, v. 1, n. 4, december 2002. Disponível em: <[www.cs.uiuc.edu/gaia](http://www.cs.uiuc.edu/gaia)>.
- SATYANARAYANAN, M. Pervasive computing: Vision and challenges. *IEEE Personal Communications*, p. 10–17, ago. 2001. Disponível em: <[citeseer.nj.nec.com/satyanarayanan01pervasive.html](http://citeseer.nj.nec.com/satyanarayanan01pervasive.html)>.
- YAMIN, A. C. et al. Towards merging context-aware, mobile and grid computing. In: LONDON:SAGE. *Journal of High Performance Computing Applications (JHPCA)*. [S.l.], 2003.