

Gerência de Infra-estrutura Distribuída na Arquitetura de Desenvolvimento de Aplicações Ubíquas Continuum (position paper)

Rodolfo Stoffel Antunes¹, Cristiano André da Costa¹,
Cláudio Fernando Resin Geyer², Adenauer Corrêa Yamin³

¹UNISINOS – Universidade do Vale do Rio dos Sinos
São Leopoldo, RS, Brasil

²UFRGS – Unifersidade Federal do Rio Grande do Sul
Porto Alegre, RS, Brasil

³UCPel – Universidade Católica de Pelotas
Pelotas, RS, Brasil

rodolfo.s.antunes@gmail.com, cac@unisinós.br, geyer@inf.ufrgs.br, adenauer@ucpel.tche.br

Resumo. *O avanço da computação ubíqua traz a necessidade do projeto de novas e complexas aplicações. Uma infra-estrutura de software permite a abstração de diversas questões de implementação, simplificando seu desenvolvimento. O Continuum é uma infra-estrutura de software que integra um framework e um middleware específicos para a computação ubíqua. Este artigo apresenta a pesquisa desenvolvida durante o projeto do serviço cell information base, utilizado pelo Continuum, bem como o modelo desenvolvido a fim de guiar a implementação deste serviço.*

Abstract. *The advances in ubiquitous computing brings the need to design new and complex applications. A software infrastructure can be used to abstract many implementation questions, simplifying their development. Continuum is a software infrastructure that integrates both a framework and a middleware specific for the ubiquitous computing. This paper presents the research carried on during the design of the service cell information base, used by Continuum, as well as the model created to guide the implementation of the service.*

1. Introdução

O termo computação ubíqua foi utilizado pela primeira vez em [Weiser 1991]. O autor afirma que é necessária uma maior integração da tecnologia da informação com o cotidiano, de forma que ela se torne transparente, auxiliando os usuários a lidar com a sobrecarga diária de informações. Pesquisas na computação ubíqua englobam questões de diversas áreas, dentre elas sistemas distribuídos e computação móvel, além de questões próprias da computação pervasiva [Satyanarayanan 2001].

[Costa et al. 2008] apresenta um estudo sobre os desafios que caracterizam a computação ubíqua, utilizando-o para definir um modelo abrangente de infra-estrutura de software para a computação ubíqua. O modelo tem como elementos principais um *framework*, para auxiliar no projeto de aplicações para a arquitetura, e um *middleware*, responsável por tratar diversas questões da execução destas aplicações.

Este modelo é a base da definição do Continuum [Costa 2008], uma infra-estrutura de software para aplicações ubíquas, baseada em uma arquitetura orientada a serviços (SOA). Uma SOA é composta por um conjunto de serviços base, que podem ser utilizados para a criação de novos serviços e aplicações através de composição e gerenciamento [Papazoglou and Georgakopoulos 2003]. Assim como no modelo abrangente, o Continuum é composto por um *framework* para auxílio ao desenvolvimento de aplicações, e um *middleware*, que tem como núcleo uma série de serviços plugáveis, responsáveis pelas principais funcionalidades do ambiente de execução.

Alguns dos serviços que compõe o *middleware* do Continuum já possuem implementação funcional. Há, porém, serviços básicos do *middleware*, apresentados em [Costa 2008], que possuem descrito apenas o modelo da interface que deveriam possuir. Um destes serviços é o *cell information base*, utilizado pelo Continuum para a manutenção do ambiente distribuído formado pelas entidades que o executam. Este artigo apresenta a pesquisa realizada para determinar os requisitos, além da metodologia, a serem empregados no projeto deste serviço. Em seguida, é apresentado o modelo criado para guiar implementação do serviço.

A seção 2 apresenta o modelo de abstração de entidades físicas utilizado pelo Continuum. Na seção 3, apresentam-se características de propostas similares a do *cell information base*. A seção 4 apresenta, em detalhes, o modelo proposto para a implementação do serviço no *middleware* do Continuum. Finalmente, na seção 5, são apresentadas considerações finais relativas ao trabalho desenvolvido e sua continuidade.

2. Modelo de Abstração de Infra-estrutura

[Costa 2008] propõe um modelo de abstração que permite a representação, no contexto de aplicações ubíquas, de entidades relevantes do mundo real. Também é proposta uma notação baseada na linguagem UML, que pode ser utilizada para diagramar ambientes representados através do modelo. A Figura 1 é um exemplo de um ambiente representado através desta notação.

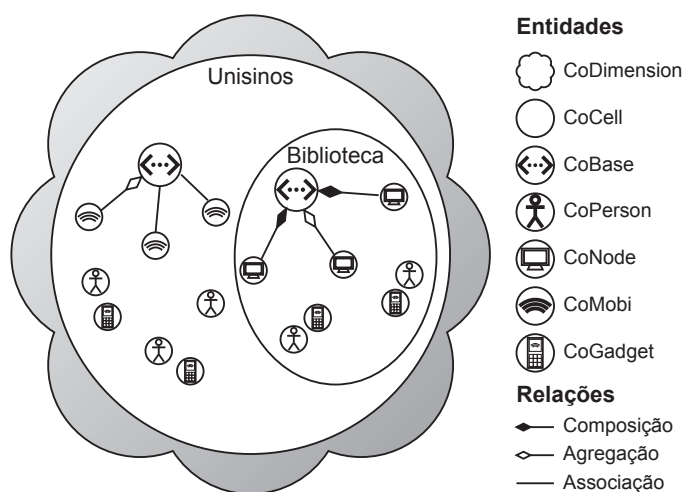


Figura 1. Ambiente modelado através das abstrações do Continuum

No modelo, uma *CoDimension* engloba todas as entidades que podem estar contidas no ambiente de uma aplicação. Locais presentes no ambiente são representados

através de células, denominadas *CoCells*. Além de estar contida na CoDimension, uma célula pode estar contida em outra célula, permitindo a representação de composições de locais. O nível de abstração representado por uma célula varia com a aplicação desenvolvida.

CoPersons representam as pessoas presentes no ambiente, que estão localizadas em uma célula, e podem se deslocar para outras. Os objetos representados pelo modelo, denominados *CoNodes*, englobam dispositivos eletrônicos presentes nas células, como computadores, sensores, ou dispositivos móveis. *CoNodes* padrão representam dispositivos estáticos básicos, tais como computadores de mesa. Além desses, existem tipos particulares de *CoNodes* empregados na infra-estrutura. Cada célula possui um nodo especial, denominado *CoBase*, responsável pela execução de serviços básicos para seu gerenciamento. Dispositivos com características móveis, como notebooks com conexão por rede sem fio, são denominados *CoMobis*. Dispositivos móveis com propósitos específicos, como celulares ou PDAs, são tratados de maneira diferenciada pelo ambiente e são denominados *CoGadgets*.

A relação entre um nodo e uma célula pode ser de três tipos: **composição**, quando um nodo é parte constituinte da célula, sem o qual ela não pode existir; **agregação**, quando o nodo possui uma relação forte com a célula; e **associação**, quando um nodo possui uma relação fraca com a célula. Uma relação entre duas células sempre será uma composição. Uma relação entre uma célula e um usuário, por sua vez, sempre será uma associação.

O objetivo do serviço *cell information base* é manter a estrutura de entidades formada através do ambiente modelado, assim como permitir a pesquisa aos seus atributos. Este serviço será apresentado em detalhes na seção 4.

3. Trabalhos Relacionados

A funcionalidade do CIB pode ser comparada a de um serviço de informações de contexto. Tais serviços, porém, têm uma abrangência maior no escopo das informações coletadas, o que não é objetivo do CIB. Para definir-se os requisitos para a modelagem do CIB, foram consultadas referências a serviços similares, presentes em outras infra-estruturas para a computação ubíqua.

O *contextual information services* (CIS) [Judd and Steenkiste 2003] é um serviço que provê informações de contexto sobre entidades e recursos em ambientes ubíquos. O principal componente do CIS é o *query synthesizer*, que é a interface para a execução de consultas no serviço. Ele divide uma consulta complexa em sub-consultas, que são enviadas para instâncias do serviço que mantêm as fontes de informação de contexto. A interface de consulta das fontes de informação é a mesma utilizada pelo *query synthesizer*, composta por um método de consulta com uma semântica similar à linguagem SQL. O CIS pode gerenciar vários tipos de informações de contexto, mas não considera a organização dos espaços físicos do ambiente. O CIB, por sua vez, armazena as relações entre os espaços físicos do ambiente, utilizando-as como meio para propagação de suas pesquisas.

A infra-estrutura para computação ubíqua Gaia [Roman et al. 2002] possui, em seu *middleware*, dois componentes para o gerenciamento de entidades. O *presence service* detecta e mantém informações das entidades físicas e digitais no ambiente, através de um mecanismo de *leasing*, que permite a eliminação de entidades que não renovam seu cadastro. Entidades digitais periodicamente entram em contato com o serviço para

renovar seu *lease*, enquanto entidades físicas são observadas pró-ativamente, por meio de sensores no ambiente. O *space repository*, por sua vez, armazena as informações coletadas pelo *presence service*. O meio de armazenamento utilizado pode ser composto de diferentes técnicas, como um banco de dados ou um espaço de tuplas. As consultas ao serviço utilizam a linguagem SQL. Estes serviços levam em consideração dispositivos e aplicações em execução, mas não possuem uma gerência específica dos espaços físicos e suas relações.

O EXEHDA [Yamin 2004] é um *middleware* para a execução de aplicações na infra-estrutura ISAM. O EXEHDA utiliza o conceito de célula para abstrair um espaço físico composto por diversos dispositivos que executam aplicações do ISAM, interligados através da mesma infra-estrutura de rede. Cada célula do EXEHDA possui uma instância do serviço também denominado *cell information base* (EXEHDAcib), responsável por armazenar as informações dos dispositivos presentes no ambiente, além de endereços para comunicação com outras células. Os dados coletados são armazenados em um serviço de diretório LDAP. O EXEHDAcib é limitado ao gerenciamento de apenas uma célula do ambiente, sendo necessária uma instância para cada espaço físico. O uso de um diretório LDAP como armazenamento dos dados também limita sua flexibilidade.

4. Modelo Proposto

Cada CoBase no ambiente do Continuum executa uma instância do CIB, que mantém armazenados os atributos de células, usuários e dispositivos que se conectam e se ao CoBase, e se mantêm ativos no ambiente. Quando há mais de uma instância do CIB no ambiente, as informações coletadas estarão distribuídas entre elas. Uma consulta ao serviço, então, deve ser propagada entre essas instâncias, para que todos os dados sejam recuperados.

4.1. Gerenciamento de Entidades

Cada CoBase é associado a uma ou mais células do ambiente, que são gerenciadas pela instância do CIB executada no CoBase. Segundo [Costa 2008], as relações entre células formam uma estrutura hierárquica, sendo a CoDimension a raiz dessa estrutura. Uma célula c possui uma célula pai c_p , que está um nível acima de c na hierarquia, e um conjunto de células filhas $c_{f_1}, c_{f_2}, \dots, c_{f_i}$, que estão um nível abaixo de c .

A hierarquia de células é mantida em uma estrutura de árvore, que armazena as relações entre as células gerenciadas. Quando há uma relação entre duas células gerenciadas por instâncias diferentes do CIB, a estrutura armazena o endereço para acesso a outra instância, de forma que as células relacionadas possam se comunicar. A Figura 2 ilustra uma configuração de diversas instâncias do CIB, e as células por elas controladas. Uma relação interna é mantida por uma instância do CIB. Uma relação externa, por sua vez, é mantida por duas instâncias.

Cada célula possui um conjunto básico de atributos, mantidos pelo serviço: um identificador único, utilizado nos métodos da interface para referência a uma célula específica; uma referência para a célula pai; e uma lista de referências para as células filhas. Estes atributos são atualizados de acordo com mudanças na hierarquia, e podem ser acessados pelo método de pesquisa do serviço. Outros atributos também podem ser associados à célula, de modo que as aplicações que utilizam o CIB possam incluir atributos a uma célula conforme sua necessidade.

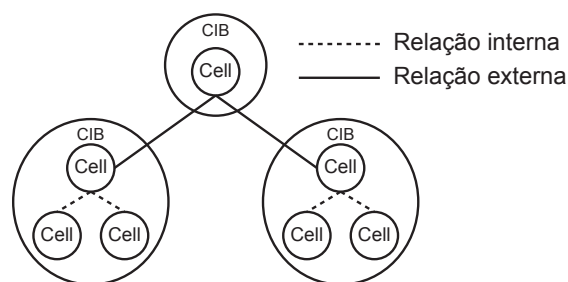


Figura 2. Ilustração das relações entre instâncias do CIB

O CIB também armazena informações de usuários e dispositivos presentes no ambiente. Usuários e dispositivos têm como atributos básicos um identificador único, utilizado pelos métodos do serviço, e o identificador da célula com a qual a entidade está associada, obtido através do CoBase. Atributos adicionais também podem ser incluídos para estas entidades, dependendo das necessidades das aplicações. Estes atributos são obtidos diretamente das entidades gerenciadas, que se conectam periodicamente ao serviço para atualizá-los.

As entidades mantidas possuem uma marcação de tempo, atualizada cada vez que a entidade acessa o serviço. Caso um dispositivo, por motivo de falha, abandone o ambiente, é possível que seus atributos permaneçam cadastrados no CIB. O serviço, neste caso, remove as informações de entidades que não entraram em contato com o serviço por um determinado período de tempo.

4.2. Consultas

As consultas do CIB têm como base a hierarquia mantida pelo serviço. Cada consulta tem como parâmetro uma célula-alvo, a partir da qual ocorre a propagação para todas as células filhas no ambiente. A propagação se encerra quando o nível mais baixo da hierarquia é atingido, ou quando a profundidade máxima da pesquisa, que também é um parâmetro, é atingida. A pesquisa é propagada apenas para as células filhas, ou seja, para os níveis inferiores da hierarquia.

A estrutura mantida pelo CIB, que descreve a hierarquia de células, é utilizada para propagar a consulta. Caso uma consulta deva ser propagada para outra instância do serviço, é utilizado o endereço de acesso cadastrado na estrutura. A segunda instância do CIB, neste caso, recebe uma consulta com parâmetros atualizados, que após executada é retornada para a instância original. O CIB agrega os resultados obtidos de outras instâncias do serviço, então, para formar a resposta final da consulta.

Além dos atributos que definem a célula-alvo e a profundidade, uma pesquisa possui outros dois conjuntos de atributos, denominados **atributos de seleção** e **atributos de retorno**. Os atributos de seleção definem um conjunto de filtros que serão aplicados durante a pesquisa, de forma que apenas entidades cujos atributos estejam de acordo com os filtros especificados sejam retornadas pela pesquisa. Os atributos de retorno, por sua vez, especificam quais informações, dentre aquelas cadastradas no serviço, deverão ser retornadas pela pesquisa. Tomando como exemplo a linguagem SQL, os atributos de seleção são equivalentes à cláusula `where`, e os atributos de retorno à cláusula `select`.

4.3. Armazenamento

As informações coletadas pelo CIB são armazenadas em um banco de dados relacional. Deste modo, caso uma falha ocorra no sistema que executa o serviço, os dados armazenados são restaurados após sua reinicialização. O banco de dados também é utilizado para carregar o estado inicial da hierarquia de células. Uma ferramenta auxiliar é utilizada para criar a hierarquia inicial do ambiente, que então é armazenada no banco de dados, para ser utilizada pelo CIB. Cada instância do serviço possui um banco de dados próprio, presente no mesmo dispositivo que executa a instância. Os dados armazenados em uma instância do serviço não serão armazenados no banco de dados de outras, podendo ser obtidos apenas através do método de consulta.

5. Considerações Finais

A computação ubíqua tem avançado rapidamente, graças à popularização dos dispositivos móveis, aliado a capacidade desses de acesso a redes sem fio. O desenvolvimento de uma aplicação ubíqua, por sua vez, tem como desafio o tratamento das diversas questões relacionadas com a área, tornando seu projeto uma tarefa complexa. Tendo como base um modelo atualizado para a computação ubíqua, a infra-estrutura de software Continuum apresenta diversas vantagens para o desenvolvimento destas aplicações.

Dando continuidade ao trabalho, será desenvolvido um protótipo do serviço, baseado no modelo apresentado, implementado na linguagem Java. A interface do serviço utilizará os padrões de *Web services* SOAP e WSDL. A descrição de entidades do serviço será realizada através de documentos XML, de forma que se possa avaliar o desempenho de um sistema de pesquisa para o CIB baseado no padrão XPath.

Referências

- Costa, C. (2008). *Continuum: A Context-aware Service-based Software Infrastructure for Ubiquitous Computing*. PhD thesis, UFRGS.
- Costa, C., Yamin, A., and Geyer, C. (2008). Towards a general software infrastructure for ubiquitous computing. *IEEE Pervasive Computing*, 7(1):64–73.
- Judd, G. and Steenkiste, P. (2003). Providing contextual information to pervasive computing applications. In *1st International Conference on Pervasive Computing and Communications, 2003, Fort Worth*, pages 133–142, Washington. IEEE.
- Papazoglou, M. and Georgakopoulos, D. (2003). Introduction: Service-oriented computing. *Communications of the ACM*, 46(10):24–28.
- Roman, M., Hess, C., Cerqueira, R., Rananathan, A., Campbell, R., and Nahrstedt, K. (2002). A middleware infrastructure for active spaces. *IEEE Pervasive Computing*, 1(4):74–83.
- Satyanarayanan, M. (2001). Pervasive computing: Vision and challenges. *IEEE Personal Communications*, 8(4):10–17.
- Weiser, M. (1991). The computer for the 21st century. *Scientific American*, 265(3):94–104.
- Yamin, A. (2004). *Arquitetura para um Ambiente de Grade Computacional Direcionada às Aplicações Distribuídas, Móveis e Conscientes de Contexto da Computação Pervsiva*. PhD thesis, UFRGS.