

Avaliação de um Serviço para Gerenciamento de Entidades Físicas em Aplicações Ubíquas

Rodolfo Antunes¹, Cristiano Costa², Jorge Barbosa²,
Cláudio Geyer¹, Adenauer Yamin³

¹UFRGS – Universidade Federal do Rio Grande do Sul
Porto Alegre, RS, Brasil

²UNISINOS – Universidade do Vale do Rio dos Sinos
São Leopoldo, RS, Brasil

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

rsantunes@inf.ufrgs.br, cac@unisinis.br, jbarbosa@unisinis.br,

geyer@inf.ufrgs.br, adenauer@ucpel.tche.br

Abstract. *Ubiquitous computing enables applications that require information about the physical environment in which they execute. This paper presents the model for the cell information base service, which allows applications to have a view of the physical entities in their execution context, based on an abstraction model. The proposed model uses a cell hierarchy to manage the environment with queries regulated by selection and return attributes. The paper presents the proposal's conceptual foundation, a review of related work, and a evaluation of the service, based on a prototype developed from the model.*

Resumo. *A computação ubíqua promove aplicações que necessitam de informações sobre o ambiente físico no qual executam. Este artigo apresenta a modelagem do serviço cell information base, que permite às aplicações terem uma visão das entidades físicas em seu contexto de execução, com base em um modelo de abstração. O modelo proposto utiliza uma hierarquia de células para gerenciar o ambiente com pesquisas reguladas por atributos de seleção e de retorno. São apresentados os fundamentos conceituais da proposta, uma revisão dos trabalhos relacionados, e uma avaliação do serviço, com base em um protótipo desenvolvido a partir do modelo.*

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, além de aspectos próprios, questões das áreas de sistemas distribuídos e computação móvel [Satyanarayanan 2001].

Um estudo sobre os desafios que caracterizam a computação ubíqua é apresentado em [Costa et al. 2008]. Estes desafios são utilizados na formulação de um modelo

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, que possuem restrições de hardware, são tratados de maneira diferenciada pelo ambiente e são denominados *CoGadgets*.

O objetivo do serviço *cell information base* é manter as informações sobre 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 3.

3. Serviço de Gerenciamento do Ambiente

O *cell information base* (CIB) é um serviço básico do *middleware* do Continuum, responsável por armazenar atributos de células, dispositivos, e usuários presentes no ambiente. Cada célula, incluindo os usuários e dispositivos nela presentes, são associados a um determinado *CoBase*, e poderão fazer uso dos serviços de gerenciamento por ele oferecidos. Por este motivo, haverá uma instância do CIB em execução em cada *CoBase* do ambiente, que será responsável por gerenciar as entidades nele conectadas.

Quando existir apenas um *CoBase*, a instância do CIB será responsável pela gerência de todas as entidades presentes no ambiente, devendo responder por todas as consultas realizadas no serviço. Havendo mais de um *CoBase*, para que a sobrecarga de gerenciamento do ambiente seja distribuída entre mais de um dispositivo, também haverá um número equivalente de instâncias do CIB, sendo cada uma responsável pela gerência de um subconjunto das entidades do ambiente. Uma consulta ao serviço, então, deve ser propagada entre essas instâncias, para que todos os dados sejam recuperados.

3.1. Gerenciamento de Células

O modelo de abstração proposto permite o estabelecimento uma hierarquia formada pelas relações existentes entre as células do ambiente, 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, onde cada nodo representa uma das células gerenciadas, que serão ligados por arestas que representam as relações entre eles. A Figura 2(a) ilustra um exemplo onde um único CIB é responsável pelo gerenciamento de três células. Neste caso, haverá um objeto representando cada uma

das células do ambiente, os quais armazenarão uma lista com referências para suas células filhas, além da referência para a célula pai. No exemplo, c_0 armazenará referências para c_1 e c_2 , enquanto c_1 e c_2 possuirão uma referência para c_0 .

Quando há mais de uma instância do CIB em execução no ambiente, a manutenção das células será dividida entre todas as instâncias. Neste caso, a estrutura de células do ambiente será particionada, sendo cada instância responsável pelo gerenciamento de uma destas partições. Cada instância do CIB deverá gerenciar pelo menos uma das células do ambiente, que será denominada sua célula raiz. Uma instância poderá gerenciar, além disso, um subconjunto das células filhas de sua célula raiz. Este caso é ilustrado pela Figura 2(b), onde há três instâncias do CIB, responsáveis pelo gerenciamento de seis células. A instância CIB_2 possui c_1 como célula raiz, e também é responsável pela gerência de c_3 e c_4 , que são filhas de c_1 . O mesmo ocorre com a instância CIB_3 que possui como raiz c_2 , e gerencia também c_5 e c_6 . A instância CIB_1 , por sua vez, é responsável apenas pela gerência de c_0 , sua célula raiz.

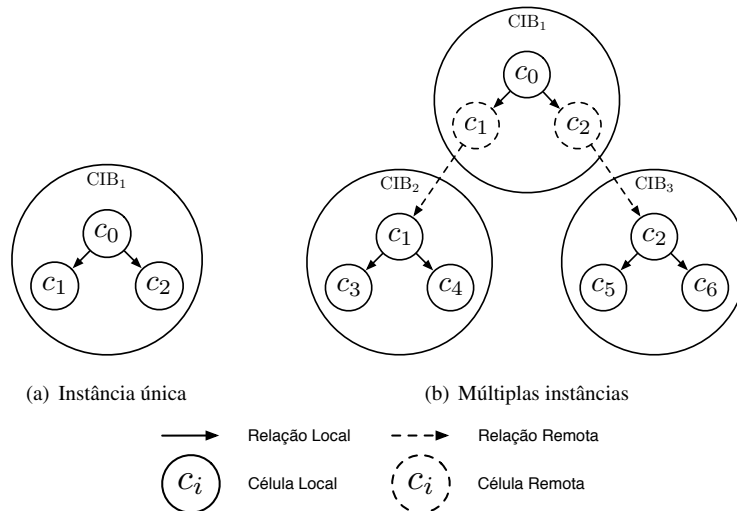


Figura 2. Ilustração das relações entre instâncias do CIB

Ainda no exemplo da Figura 2(b), apesar da hierarquia estar dividida entre as três instâncias, podem existir relações entre as células gerenciadas por instâncias diferentes, como no caso de c_2 e c_1 , que são filhas de c_0 . Neste caso, são utilizados objetos especiais na estrutura de células para o armazenamento de referências a células que são gerenciadas por outras instâncias, mas com as quais há uma relação. No exemplo, este é o caso dos objetos mantidos por CIB_1 que referenciam as células c_1 e c_2 , mantidas pelas outras instâncias. A função principal destes objetos é armazenar o endereço de comunicação com a instância na qual a célula referenciada é gerenciada, para que as instâncias do serviço possam propagar mensagens através das relações existentes na estrutura de células.

Cada célula possui um conjunto básico de atributos mantido pelo serviço: um identificador único, utilizado pelos 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 pelas aplicações que utilizam o Continuum através do 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 conforme sua necessidade. A entidade responsável pela atualização dos atributos das células no CIB é o CoBase.

3.2. Gerenciamento de Usuários e Dispositivos

Dispositivos e usuários presentes no ambiente, também denominados de entidades, são associados a uma célula, de acordo com informações obtidas do CoBase durante sua inicialização no ambiente, momento no qual estas entidades também entram em contato com o CIB para o cadastro de seus atributos no serviço. Entidades possuem, como atributos básicos: um identificador, que representa unicamente a entidade no ambiente, utilizado para referência nos métodos do serviço; e o identificador da célula ao qual a entidade está associada, obtido através do CoBase. Assim como no caso das células, outros atributos poderão ser associados as entidades, dependendo das necessidades das aplicações que utilizam o serviço. Os atributos cadastrados no serviço são informados diretamente pelas entidades, durante o cadastro, e também são periodicamente atualizados, em caso de mudança, pela própria entidade cadastrada.

Entidades mantidas pelo CIB também possuem como atributo uma marcação de tempo, atualizada cada vez que a entidade entra em contato com o serviço. Este atributo é utilizado pelo serviço para o controle do intervalo de tempo desde o último contato de uma entidade. Em condições normais, ao se desconectar, uma entidade solicita ao serviço sua retirada do ambiente. Caso uma entidade, por motivo de falha, deixe de existir no ambiente, seus atributos permanecerão cadastrados no serviço por um intervalo definido de tempo, após o qual a entidade será considerada desativada. Em ambos os casos, os atributos de entidades desativadas são movidos para um serviço de histórico, que armazena as informações sobre as entidades que deixaram de estar ativas no ambiente.

3.3. Consultas

Os atributos armazenados pelo *cell information base* podem ser necessários para outros serviços em execução no ambiente, e portanto podem ser obtidos através do método de consulta disponível na interface do CIB, denominado *query*. Uma consulta no CIB é formada por dois conjuntos de atributos. **Atributos de seleção** são formados por um conjunto de tuplas atributo/valor, que definem um conjunto de filtros que serão aplicados durante a pesquisa. Apenas as entidades que possuem os atributos e valores definidos na pesquisa serão adicionados ao resultado. Os atributos de seleção tem por objetivo atuar de modo similar a cláusula *where* da linguagem SQL. **Atributos de retorno** são formados por um conjunto de nomes de atributos, que é utilizado para selecionar, dentre os atributos cadastrados nas entidades que serão retornadas pela pesquisa, quais serão de fato incluídos no resultado, sendo os outros descartados. Novamente realizando uma comparação com a linguagem SQL, o objetivo dos atributos de retorno é atuar de modo similar a cláusula *select*.

O CIB executa uma consulta através de sua propagação na estrutura de células mantida pelo serviço. O controle da pesquisa é realizado por uma série de parâmetros, informados ao método *query*, que definem como a pesquisa deve ser propagada entre as células. Pesquisas são iniciadas a partir de uma célula-alvo, que define o ponto inicial de propagação da consulta na hierarquia de células. Qualquer célula do ambiente pode ser definida como célula-alvo de uma pesquisa.

Iniciando pela célula-alvo, o método `query` acessa as entidades presentes na célula, selecionando aquelas que têm seus atributos dentro do especificado no conjunto de atributos de seleção. Os atributos das entidades selecionadas são copiados, então, para o resultado da pesquisa, levando em consideração aqueles que foram informados no conjunto de atributos de retorno. Após este processamento, o método acessa a lista de células filhas da célula atual, e prossegue com a busca por entidades em cada célula contida na lista. Este procedimento pode ser executado até que uma célula que não possua filhas seja encontrada, ou até que um limite de propagação, definido como parâmetro do método `query`, seja atingido. Além desses dois casos, a pesquisa pode ser encerrada caso um limite máximo de resultados, que também é um parâmetro de `query`, seja atingido.

Durante a propagação da consulta na hierarquia, o método poderá encontrar uma referência a uma célula gerenciada por uma instância diferente do CIB (no exemplo da Figura 2(b), este seria o caso de uma consulta iniciada em c_0 , que seria propagada para c_1 e c_2). Neste caso, o método acessa o objeto que mantém a referência para a célula remota, que conterá o endereço da instância do CIB que gerencia a célula. Através deste endereço, o método envia uma mensagem solicitando que a instância continue a execução da consulta. Esta mensagem conterá os atributos de seleção e retorno originais da consulta, e terá seus parâmetros modificados para refletir o estado atual da propagação, tais como os níveis já percorridos e o número de resultados já encontrados.

Ao finalizar o processamento de uma consulta propagada, o serviço retornará os resultados para a instância do serviço da qual a consulta foi recebida. A instância na qual se encontra a célula-alvo, portanto, receberá todos os resultados obtidos pelas instâncias as quais a consulta foi propagada, e será responsável por agregar estes resultados em um único, que será retornado para a entidade que realizou a invocação do método `query`.

4. Avaliação

A modelagem descrita na seção anterior foi utilizada como base para o desenvolvimento de um protótipo do serviço, que foi utilizado para uma análise baseada nos experimentos apresentados nesta seção. O principal objetivo destes experimentos é a avaliação do tempo necessário para a execução de consultas em diferentes configurações da hierarquia de células mantida pelo serviço, bem como o volume de dados gerado por uma instância do serviço durante a propagação de consultas e resultados.

Para avaliação do serviço, foram utilizadas hierarquias contendo entre 1 e 4096 células. Dispositivos fictícios foram cadastrados nestas hierarquias, sendo realizadas pesquisas percorrendo toda a hierarquia, com o objetivo de retornar os identificadores destes dispositivos. As pesquisas realizadas tiveram como célula-alvo a raiz da hierarquia, e não tiveram limitados o número de resultados obtidos nem a profundidade a ser percorrida. Foram avaliadas configurações com uma, três e cinco instâncias do serviço. Os experimentos foram realizados em computadores Intel Core2 de 2.4 GHz e 4 GB de memória, e executavam o sistema operacional Linux.

4.1. Tempo de Pesquisa

A primeira métrica analisada durante a execução dos experimentos foi o tempo necessário pelo serviço para a propagação de uma consulta na hierarquia, desde a célula raiz até todas as células folha. A Figura 3 apresenta os resultados obtidos nos experimentos. O eixo

horizontal apresenta o número de células presente na hierarquia durante o experimento e o eixo vertical o tempo necessário para o processamento da consulta em segundos. Cada curva do gráfico apresenta a métrica analisada para uma das configurações das instâncias do serviço utilizadas durante os experimentos.

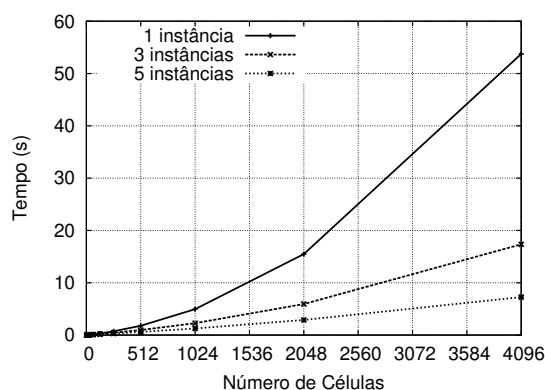


Figura 3. Tempo de execução de pesquisas

O resultado mostra que o tempo necessário para a execução de uma consulta possui um crescimento próximo do linear para hierarquias até 2048 células. A partir deste ponto, porém, o tempo necessário para a execução da consulta possui um crescimento acentuado para o caso onde o serviço foi utilizado de forma centralizada, sendo necessários 53 segundos para que a consulta fosse completada na hierarquia de 4096 células. Ao utilizar-se três instâncias do serviço, o tempo necessário para a execução da consulta apresenta um decrescimento considerável, passando a ser 17 segundos no caso da hierarquia de 4096 células. Utilizando-se cinco instâncias do serviço, observa-se novamente uma redução no tempo necessário para a conclusão da consulta, que passa a ser de 7 segundos para a hierarquia de 4096 células.

Os resultados obtidos mostram que a execução do serviço de modo centralizado faz com que haja uma latência de resposta considerável para as consultas. Considerando-se que a latência na resposta do serviço é um fator crítico para a eficiência das aplicações que o utilizam, e que uma hierarquia com um elevado número de células também possui um número elevado de clientes do CIB, poderá haver uma degradação tanto do serviço quanto das aplicações que o utilizam. O uso de múltiplas instâncias do serviço, por sua vez, se mostra como uma solução para este problema, havendo um ganho de 67% na latência de resposta com o uso de três instâncias do serviço, e 86% com cinco instâncias.

4.2. Tamanho de Mensagens

A segunda métrica avaliada durante os experimentos foi o tamanho das mensagens que cada instância do serviço necessitou gerar para a transmissão dos resultados da pesquisa. Considerando-se uma hierarquia de tamanho fixo, quanto maior for o número de instâncias utilizadas para sua gerência, menor será o número de células pelas quais cada instância será responsável. Consequentemente, cada instância receberá um menor número de consultas, devido a maior distribuição destas mensagens. Esta avaliação, então, tem por objetivo comparar o tamanho das mensagens de resposta geradas pelo serviço, variando-se o número de células gerenciadas e o número de instâncias do serviço. Esta análise inclui apenas as mensagens enviadas entre as instâncias do serviço. Também considera-

se apenas os dados contidos como resposta a consulta, sem a sobrecarga gerada pelos protocolos utilizados.

A Figura 4 apresenta os resultados obtidos na avaliação. O eixo horizontal mostra, novamente, o número de células presente na hierarquia durante o experimento. O eixo vertical, por sua vez, apresenta a média do tamanho das mensagens, em KiloBytes, que foram geradas por cada instância do serviço. Cada curva do gráfico apresenta a métrica analisada para um determinado número de instâncias responsáveis pela gerência das hierarquias de células utilizadas durante os experimentos.

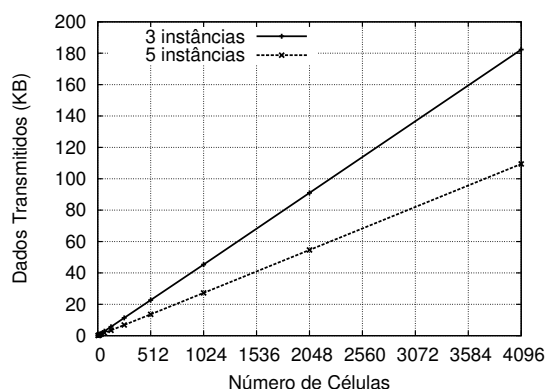


Figura 4. Tráfego gerado por uma instância do serviço

O resultado mostra que o tamanho das mensagens transferidas entre as instâncias do serviço durante uma consulta cresce de forma linear conforme o número de células do ambiente. Tal crescimento se deve ao fato de que o número de resultados retornados pela consulta também sofre incrementos, devido à configuração utilizada para os experimentos. Com três instâncias do serviço, a média de tamanho gerado por cada instância se mantém em 182 KB para a hierarquia de 4096 células. Com cinco instâncias, por sua vez, o tamanho médio gerado se mantém em 109 KB para a hierarquia de 4096 células.

Este resultado mostra que ao utilizar-se um maior número de instâncias do serviço, o tamanho das mensagens geradas, e conseqüentemente o tráfego de rede gerado por cada instância, tenderá a diminuir. Isto se deve ao fato de que cada instância deverá gerenciar uma quantidade menor de informações, reduzindo o volume de dados gerado em resultados de consultas. Nos resultados avaliados, a média do tamanho de mensagens gerado por cada instância foi reduzido em 40%. Embora um número maior de instâncias aumente o desempenho das instâncias como um todo, a rede utilizada para comunicação das instâncias do serviço deve suportar o tráfego gerado pela propagação de pesquisas, evitando que a rede se torne um ponto de falha do ambiente de execução das aplicações que utilizam o serviço.

5. Trabalhos Relacionados

A funcionalidade do serviço de gerenciamento pode ser comparada a de um serviço de informações de contexto. Existem diversas propostas de serviços nesta categoria, com propostas diferentes quanto à abrangência das entidades e informações por eles gerenciadas. A seguir serão apresentados alguns destes serviços, que apresentam funcionalidades similares ao *cell information base*.

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 ICE [Strohbach et al. 2007] é um *middleware* baseado em sessões de contexto, definidas como um fluxo de notificações, entre uma fonte e um receptor, sobre a alteração no estado das informações de contexto. Segundo os autores, o uso da comunicação orientada a eventos para disseminação de informações de contexto amplia a escalabilidade do sistema, pois não há a necessidade da atualização destas informações em um servidor central responsável por seu gerenciamento. A arquitetura utiliza dois protocolos separados, um para o gerenciamento de fontes e assinantes do mecanismo publicar-assinar, e outro unicamente para a disseminação das notificações das fontes aos assinantes cadastrados.

O *context management framework* (CMF) [van Kranenburg et al. 2006] é voltado para a aquisição e distribuição de informações de contexto a partir de diversas fontes. Aplicações utilizam uma interface de comunicação por eventos para ter acesso às informações disponibilizadas por um provedor de contexto. Este provedor, por sua vez, é responsável por gerenciar assinaturas de clientes e permissões de acesso às informações. Um provedor de contexto pode fazer uso de múltiplas fontes de informação, que podem ser um simples *wrapper* para um sensor responsável por coletar dados sobre o ambiente, ou ainda um *context reasoner*, que emprega métodos inteligentes para extração e derivação de outras informações a partir de fontes de dados diversas.

Estes serviços atuam como entidades responsáveis por obter informações distribuídas pelo ambiente de execução, tornando-as disponíveis para que as aplicações possam utilizá-las. Estes serviços possuem um escopo mais genérico com relação ao tipo de informações por eles gerenciadas, procurando capturar o maior número de dados possível sobre o contexto, tornando-os disponíveis. A principal vantagem do CIB está em seu foco no gerenciamento das entidades que compõem a infraestrutura na qual as aplicações estarão em execução. A funcionalidade do CIB é manter atualizada a visão que as aplicações possuem sobre o ambiente, de acordo com a modelagem apresentada na seção 2, principalmente no que diz respeito à hierarquia de células formada pelos espaços físicos representados.

6. Considerações Finais

Os recentes avanços nas áreas da computação móvel e ubíqua requerem a criação de novas aplicações, as quais possuem projetos cada vez mais complexos devido ao número de questões que devem ser por elas tratadas. Neste contexto, a infraestrutura de software Continuum se apresenta como uma ferramenta que visa simplificar o projeto de tais aplicações, tratando diversos detalhes presentes na computação ubíqua através dos serviços oferecidos por seu *middleware*.

Neste artigo, foi apresentado o modelo de abstração de entidades físicas proposto pelo Continuum, que permite a representação, no contexto das aplicações, das entidades relevante no mundo real. Também foi apresentado o modelo do serviço *cell information*

base, presente no *middleware* da infraestrutura de software do Continuum, responsável pela gerência do ambiente representado através do modelo proposto. O modelo utiliza uma hierarquia de células, que representa os espaços físicos do ambiente de execução. Esta hierarquia é utilizada para gerenciar a propagação de consultas entre as diversas instâncias do serviço no ambiente. Um protótipo do serviço foi implementado, para permitir a avaliação de suas principais características. Os resultados obtidos mostram que o uso de múltiplas instâncias para o gerenciamento de um ambiente traz vantagens, principalmente em relação ao desempenho das consultas, bem como no tráfego de rede gerado pelos dispositivos que executam o serviço.

Para permitir que as aplicações desenvolvidas através do *framework* do Continuum possam utilizar as funcionalidades do serviço desenvolvido, este será integrado ao *middleware* utilizado pela infraestrutura de software. Esta integração permitirá que novos projetos sejam desenvolvidos utilizando a infraestrutura do Continuum como base, tais como um sistema para a gerência de uma rede social espontânea, com ênfase em interações ubíquas, atualmente em desenvolvimento.

Referências

- Costa, C. (2008). *Continuum: A Context-aware Service-based Software Infrastructure for Ubiquitous Computing*. PhD thesis, UFRGS.
- Costa, C., Kellermann, F., Antunes, R., Cavalheiro, L., Yamin, A., and Geyer, C. (2009). Continuum: A service-based software infrastructure for ubiquitous computing. In *7th International Conference on Pervasive Computing and Communications, 2009, Galveston*, pages 1–4, Washington. IEEE.
- 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.
- Satyanarayanan, M. (2001). Pervasive computing: Vision and challenges. *IEEE Personal Communications*, 8(4):10–17.
- Strohbach, M., Bauer, M., Kovacs, E., Villalonga, C., and Richter, N. (2007). Context sessions: a novel approach for scalable context management in NGN networks. In *Workshop on Middleware for next-generation converged networks and applications, 2007, Port Beach*, pages 1–6, New York. ACM.
- van Kranenburg, H., Bargh, M. S., Iacob, S., and Peddemors, A. (2006). A context management framework for supporting context-aware distributed applications. *IEEE Communications Magazine*, 44(8):67–74.
- Weiser, M. (1991). The computer for the 21st century. *Scientific American*, 265(3):94–104.