

# Uma Proposta para Consciência de Contextos Dinâmicos na Ubicomp

João L. B. Lopes<sup>1,4</sup>, Rodrigo S. Souza<sup>1,4</sup>, Gizele I. Gadotti<sup>3</sup>, Márcia Z. Gusmão<sup>1,3</sup>,  
Adenauer C. Yamin<sup>2,3</sup>, Claudio F. R. Geyer<sup>4</sup>

<sup>1</sup>Instituto Federal Sul-rio-grandense (IFSul)

<sup>2</sup>Universidade Católica de Pelotas (UCPel)

<sup>3</sup>Universidade Federal de Pelotas (UFPel)

<sup>4</sup>Universidade Federal do Rio Grande do Sul (UFRGS)

{mzgusmao, gizele.gadotti, adenauer}@ufpel.edu.br

{jlblopes, rssouza, geyer}@inf.ufrgs.br

**Abstract.** *Ubiquitous computing (Ubicomp) environments are characterized by high distribution, heterogeneity and dynamism. In these environments, applications must be aware of their contexts and adapt to changes in them. Thus, a major research challenge in the area of Ubicomp is related to context awareness. Considering the characteristics of ubiquitous environments, this paper presents a proposal for context awareness, called DynamiCC (Dynamic Context Composition), that includes context modeling, contextual data collection, actuation on the environment, and composition and interpretation of contextual information. We understand that the main contributions of this work are the employment of a hybrid approach for context modeling and the proposal of an architecture that supports the interpretation and the composition of dynamic contexts, which enables the construction of complex contexts in runtime. To assess the functionality of the DynamiCC, we did a discussion of usage scenarios.*

**Resumo.** *Os ambientes para computação ubíqua (Ubicomp) caracterizam-se pela elevada distribuição, heterogeneidade e dinamismo. Nestes ambientes as aplicações devem estar cientes de seus contextos e adaptarem-se às mudanças nos mesmos. Assim, um dos grandes desafios de pesquisa da área de Ubicomp está relacionado à consciência do contexto. Considerando as características dos ambientes ubíquos, este artigo apresenta uma proposta para consciência do contexto, denominada DynamiCC (Dynamic Context Composition), que trata aspectos relacionados a modelagem do contexto, a coleta de dados contextuais e a atuação sobre o meio, e a interpretação e composição das informações contextuais. Entende-se como contribuições centrais do trabalho, o emprego de uma abordagem híbrida para modelo de contexto, bem como a proposição de uma arquitetura para suporte à interpretação e à composição de contextos dinâmicos, obtidos por diferentes sensores em células distribuídas do ambiente ubíquo. A premissa é que a composição dinâmica viabilize a construção de contextos complexos, em tempo de execução das aplicações. Para avaliar as funcionalidades da DynamiCC é feita a discussão de cenários de uso.*

## 1. Introdução

Mark Weiser, em seu artigo [Weiser 1991], que constitui referência histórica da computação ubíqua, idealizou ambientes físicos com dispositivos computacionais inte-

grados que auxiliariam indivíduos na realização de suas tarefas cotidianas. Weiser denominou de computação ubíqua a área de pesquisa que estuda a integração da tecnologia às atividades humanas de forma transparente, quando e onde for necessário.

Considerando esse cenário idealizado por Weiser, a Ubicomp vem sendo considerada como o novo paradigma computacional do século XXI, tendo sido apontada pela SBC (Sociedade Brasileira de Computação) como parte de um dos cinco grandes desafios para a pesquisa na área de computação no período de 2006 a 2016, conforme consta no documento “Grandes Desafios da Pesquisa em Computação no Brasil - 2006 - 2016”<sup>1</sup>.

A infraestrutura computacional para Ubicomp tende a ser altamente distribuída e heterogênea, apresentando elevado dinamismo pelo fato dos elementos de contexto, dispositivos e componentes de software mudarem com frequência seu status no ambiente ubíquo. Assim, um dos desafios dessa área de pesquisa diz respeito à capacidade dos recursos para desenvolvimento de software destes ambientes explorarem a consciência do contexto para o processo de adaptação das aplicações [Costa et al. 2008].

Nos ambientes computacionais para provimento de aplicações ubíquas a informação contextual é obtida a partir de vários sensores. Assim, nestes ambientes, os componentes de software das aplicações devem estar cientes de seus contextos de interesse e automaticamente adaptarem-se às suas mudanças, caracterizando com isso a consciência do contexto [Kjeldskov and Skov 2007].

As mudanças no estado do contexto do ambiente em que executam as aplicações ubíquas produzem eventos, que podem se originar de fontes heterogêneas. Também, o contexto pode ser provido em diferentes níveis de abstração, desde uma informação sobre uma propriedade do ambiente físico, até informações sobre atividades de um usuário obtidas a partir de aplicativos e bancos de dados [Baldauf et al. 2007]. Assim, os dados de contexto obtidos a partir de fontes diversas podem ser dinamicamente processados e compostos em informações contextuais de nível mais elevado, que possam ser utilizadas para qualificar o processo de adaptação das aplicações ubíquas, em tempo de execução.

A revisão de literatura aponta que a construção de uma arquitetura para consciência do contexto apresenta desafios, dentre eles: (i) a representação de um modelo de contexto que possua características como extensibilidade, padronização, expressividade e raciocínio; (ii) a aquisição do contexto a partir de fontes heterogêneas e distribuídas; (iii) o processamento das informações de contexto adquiridas e a respectiva atuação sobre o meio; e (iv) a disseminação do contexto a consumidores interessados de forma distribuída e no momento oportuno [Baldauf et al. 2007] [Bettini et al. 2010].

Considerando a premissa de integração com o *middleware* EXEHDA [Augustin et al. 2008] e o fato do mesmo não atender esses desafios, neste artigo é apresentado o estado atual da proposta para consciência do contexto DynamiCC. Esta proposta tem como premissa fundamental a concepção de um modelo de contexto e uma arquitetura de software que possibilitem a interpretação e a composição dinâmica das informações adquiridas por sensores posicionados em diferentes localidades do ambiente ubíquo. Com isso, a proposta busca viabilizar a construção de contextos complexos, cujas informações contextuais possam ser obtidas de forma distribuída, bem como ter suas regras de processamento alteradas sem a necessidade de interromper a

---

<sup>1</sup>documento disponível em <http://www.sbc.org.br>

execução das aplicações.

O artigo está organizado da seguinte forma: a seção 2 sintetiza os trabalhos relacionados. A seção 3 apresenta a concepção da proposta. A seção 4 discute o modelo de contexto proposto, e a seção 5 descreve o modelo arquitetural do software concebido para sua manipulação. A seção 6 destaca cenários de uso da proposta. Por fim, a seção 7 apresenta as considerações finais.

## 2. Trabalhos Relacionados

WComp [Ferry et al. 2011] é um *middleware* para computação ubíqua, baseado em uma infraestrutura de software, uma arquitetura para composição de serviços e um mecanismo para adaptação. Para gerenciar o dinamismo e a heterogeneidade das entidades na infraestrutura de software, é utilizado WSOAD (*Web Service Oriented Architecture for Device*). As aplicações ubíquas são baseadas em um conjunto de serviços Web para dispositivos. Os serviços não são editáveis, logo para adicionar novas funcionalidades, uma aplicação tem que ser uma composição de serviços para dispositivos. A proposta apresenta algumas limitações para lidar com a consciência do contexto [Ferry et al. 2009], sendo uma das principais restrições a não definição de um modelo expressivo para representação das informações contextuais.

CARE [Agostini et al. 2009] é um *middleware* para suporte a adaptação consciente do contexto de serviços Internet em um ambiente de computação móvel. CARE emprega uma abordagem para modelar o contexto baseada na integração de um modelo de marcação e um modelo ontológico. A integração entre estes modelos é obtida através da representação dos dados de contexto com perfis CC/PP, os quais contêm referências para classes e relacionamentos da ontologia. O processo de raciocínio sobre a ontologia também é executado para verificar a consistência do modelo de contexto. O *middleware* CARE tem um foco mais restrito, abrangendo a adaptação dos serviços usuais de Internet na perspectiva da computação móvel.

Solar [Chen et al. 2008] é um *middleware* para suporte à consciência do contexto, que coleta dados de fontes heterogêneas, agrega estes dados para obter contextos de alto nível e realiza a disseminação dessas informações de contexto através de nodos de rede. O enfoque central do Solar é na aquisição, transformação e síntese do contexto, disponibilizando uma combinação flexível e extensível de operadores para realizar esses processos. O Solar emprega um modelo informal para representação do contexto. Também, não registra dados históricos do contexto e a coleta de dados exige envolvimento direto das aplicações, já que estas precisam gerenciar sensores, operadores e requisições.

HiCon [Cho et al. 2008] é um *framework* para monitoramento e composição de contexto. O HiCon gerencia um grande volume de dados gerados por sensores em tempo real. Esse *framework* considera diferentes níveis de contexto, pessoal, regional e global, permitindo a composição dinâmica de diversos contextos. O HiCon combina contextos destes três níveis para criar um conjunto abrangente de aplicações escaláveis, baseadas em contexto. Para melhorar a escalabilidade, o HiCon funde e filtra dados contextuais fornecidos por níveis mais baixos em descritores de contexto mais expressivos e compactos. O HiCon tem um enfoque maior na filtragem dos dados contextuais, do que nas questões relacionadas com o processamento do contexto.

Em comparação com os trabalhos relacionados, a DynamiCC propõe um modelo

híbrido para representação do contexto, que proporciona elevada expressividade para o processamento do mesmo, situação que não é identificada nos trabalhos apresentados. A DynamiCC fundamenta sua proposta em uma visão semântica e composicional do contexto, uma abordagem que a diferencia dos trabalhos relacionados discutidos e que contempla as premissas de distribuição, heterogeneidade e dinamismo inerentes aos ambientes ubíquos. Além disso, diferente dos trabalhos analisados, a arquitetura de software da DynamiCC viabiliza a coleta de dados de contexto e a atuação sobre o meio, de forma distribuída. Ainda, nenhum dos trabalhos analisados pode ser considerado uma solução abrangente para os desafios da computação ubíqua. Por sua vez, a proposta DynamiCC, pela premissa de integração ao *middleware* EXEHDA, viabiliza às funcionalidades que dão suporte aos requisitos inerentes à área de Ubicomp.

### 3. Concepção da Proposta

As aplicações do ambiente gerenciado pelo EXEHDA são, por natureza, distribuídas, móveis e adaptativas ao contexto, estando disponíveis a partir de qualquer lugar, todo o tempo. O ambiente ubíquo, gerenciado pelo *middleware*, é formado por células compostas por um equipamento responsável por todos os serviços básicos da célula e por equipamentos dos usuários [Lopes et al. 2007].

A definição de contexto proposta por [Dey et al. 2001] “Qualquer informação que possa ser utilizada para caracterizar a situação de entidades (pessoa, lugar ou objeto) que sejam consideradas relevantes para a interação entre um usuário e uma aplicação, incluindo o usuário e a aplicação” é uma das mais aceitas e utilizadas atualmente por pesquisadores da área de Computação. Essa definição é abrangente quanto aos tipos de dados que podem ser considerados como contextos, sendo suficientemente ampla para incluir as diversas necessidades específicas de cada aplicação. Também, não há restrição quanto às fontes de contextos possíveis de serem utilizadas, permitindo que os dados reflitam a situação de qualquer entidade relevante para cada caso em particular [Loureiro et al. 2009].

Considerando que para processar as informações contextuais é necessário definir um modelo de contexto [Soldatos et al. 2007] e, com a perspectiva de contemplar a definição de [Dey et al. 2001], na concepção da proposta é empregada uma abordagem híbrida para modelagem e processamento do contexto. A integração de diferentes modelos pode propiciar maior flexibilidade e generalidade na construção de aplicações conscientes do contexto, minimizando limitações apresentadas individualmente pelos modelos existentes [Bettini et al. 2010]. Assim, o modelo proposto do DynamiCC integra uma abordagem semântica, baseada em ontologias (vide seção 4.1), para suporte à realização de inferências, e uma abordagem lógica, que emprega um banco de dados relacional (vide seção 4.2), para apoio a consultas e armazenamento do histórico do contexto.

Sendo oportuno que as informações de contexto sejam mantidas permanentemente [Costa 2008], a expectativa é que essa ontologia venha a ter um grande número de instâncias. Com isso, o armazenamento na forma de arquivos de texto OWL (*Web Ontology Language*)<sup>2</sup> pode implicar em acentuada demora no tempo de processamento da ontologia, não apenas na consulta a suas instâncias, mas também na realização de inferências, já que todo o conteúdo do arquivo precisaria ser manipulado. Nesse sentido, o

---

<sup>2</sup><http://www.w3.org/TR/owl2-overview/>

uso de bancos de dados para o armazenamento das instâncias da ontologias se mostra um meio eficiente para o processamento e otimização de consultas. Além disso, os bancos de dados contemplam características de robustez, controle de concorrência, escalabilidade e capacidade de recuperação de dados [Venturi Filho et al. 2010].

O modelo concebido prevê o processamento dinâmico das informações de contexto, à medida que as aplicações ubíquas são executadas. Além disso, a definição, instanciação e monitoramento do contexto foram concebidos para serem externos às aplicações. Esta abordagem potencializa aspectos de modularidade e reuso na construção de novas aplicações.

Na concepção da proposta é empregada uma abordagem orientada a eventos [Tigli et al. 2009], tanto entre os módulos da arquitetura de software como nas aplicações, para viabilizar a possibilidade de associação de regras de processamento aos contextos de interesse, as quais podem ser disparadas automaticamente quando ocorre um evento. Os eventos, na perspectiva da proposta, dizem respeito a mudanças no estado do contexto, tais como: uma propriedade do ambiente físico (temperatura fora da faixa definida, por exemplo) e ocorrências relacionadas aos componentes de software das aplicações, como intervalo de tempo e conclusão de um processamento.

A composição dinâmica de contextos é aspecto central na concepção da proposta, correspondendo à capacidade de construir contextos complexos, em tempo de execução das aplicações, que podem ser compostos a partir de dados de contexto obtidos por diferentes sensores em células distribuídas do ambiente ubíquo. Com isso, fica potencializada a expressividade das informações contextuais, que podem ser disponibilizadas aos processos de adaptação ao contexto das aplicações ubíquas.

## **4. Modelo de Contexto**

A modelagem do contexto provido pela DynamiCC segue uma abordagem híbrida, constituída por um modelo semântico, baseado em ontologias, e um modelo lógico, baseado em entidades e relacionamentos, implementado em banco de dados.

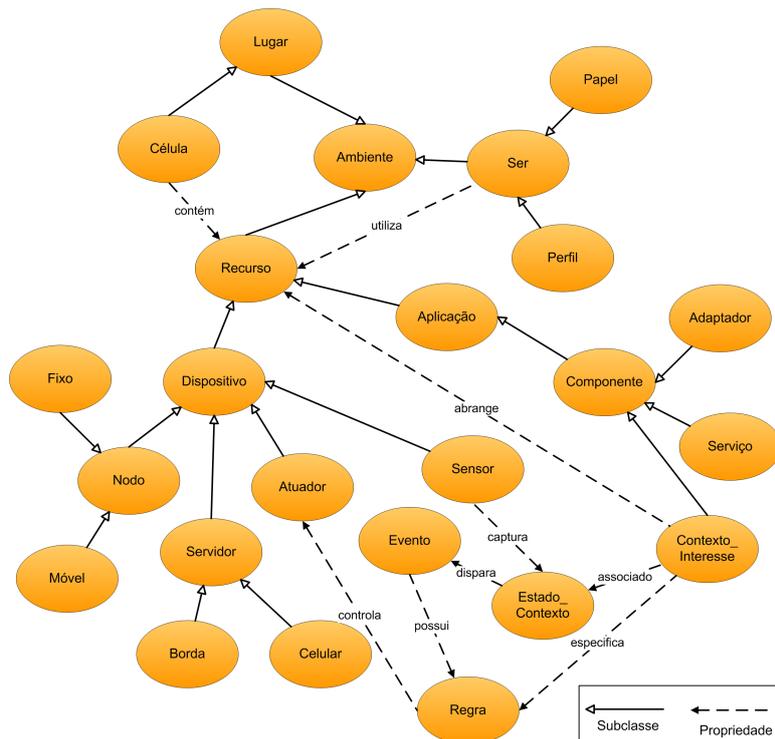
### **4.1. Modelo Semântico**

A construção da ontologia que representa o contexto do ambiente ubíquo é resultante de uma constante revisão e expansão do modelo ontológico desenvolvido pelo grupo de pesquisa G3PD<sup>3</sup>, que vem sendo feita em diversos trabalhos [Lopes 2008] [Venecian 2010] [Warken 2010] [Dilli 2010]. A ontologia apresentada na Figura 1 está integrada à arquitetura de consciência do contexto proposta (vide seção 5) e prevê suporte ao raciocínio, através da realização de inferência sobre as informações contextuais.

A ontologia é utilizada para modelar as entidades que estão envolvidas no desenvolvimento e execução das aplicações ubíquas. Nesse sentido, como meta de projeto entende-se que essa ontologia deve ser suficientemente geral para modelar diferentes tipos de aplicações. Deve ser possível modelar todas as entidades usadas pela infraestrutura computacional, incluindo recursos, pessoas e lugares. Além disso, devem ser consideradas as abstrações propostas para modelar o mundo real, bem como representadas as

---

<sup>3</sup>Grupo de Pesquisa em Processamento Paralelo e Distribuído que abrange Instituições da Região Sul



**Figura 1. Ontologia do Contexto do Ambiente Ubíquo**

informações de contexto do ambiente ubíquo, que venham a ser utilizadas para a consciência do contexto, através do modelo ontológico.

#### 4.2. Modelo Lógico

O modelo lógico do contexto atende as demandas de armazenamento das informações contextuais. Esse modelo, assim como o modelo semântico, também está integrado à arquitetura de consciência do contexto proposta.

O modelo lógico construído é resultante do mapeamento da ontologia em um modelo de entidades e relacionamentos, implementado em banco de dados relacional. Nessa estratégia de armazenamento o esquema da base relacional e a ontologia são mantidos separadamente. Com isso, deve ser estabelecida uma correspondência entre os elementos que formam a estrutura dos dois modelos, possibilitando que as consultas às instâncias sejam feitas no modelo lógico e a realização de inferências, considerando as classes, propriedades e relacionamentos, no modelo semântico [Venturi Filho et al. 2010]

A estrutura do banco de dados contempla as relações entre as aplicações, componentes, sensores, ambientes e os contextos de interesse. O banco de dados ainda armazena informações de configuração da arquitetura e as publicações provenientes dos sensores existentes no ambiente ubíquo.

Dentre as informações de configuração, destacam-se os aspectos relacionados aos sensores e aos contextos em que esses estão envolvidos, tais como: os limites máximos e mínimos dos valores publicados para cada sensor, pertencente a um determinado contexto de interesse. Esses dados são utilizados, por exemplo, por aplicações síncronas ao contexto, cujas funcionalidades são disparadas quando algum dos valores atinge os limites de

interesse da aplicação.

## 5. Modelo Arquitetural

A arquitetura de software para consciência do contexto proposta tem como premissas: (i) o emprego de um modelo híbrido de contexto (vide seção 4), constituído por uma ontologia e uma base de dados relacional; (ii) o processamento das informações adquiridas por sensores em diferentes células do ambiente ubíquo; (iii) a notificação às partes interessadas, conforme as regras de processamento do contexto utilizadas pelas diferentes aplicações; (iv) a possibilidade dos sensores e/ou atuadores serem ativados, desativados, configurados e consultados a qualquer momento, através de seus respectivos tratadores, de forma ubíqua, em qualquer célula do ambiente; e (v) a possibilidade de alterações das regras de processamento do contexto durante a execução das aplicações.

A arquitetura de software, apresentada na Figura 2, é composta por módulos distribuídos entre os dois tipos de equipamentos servidores: Celular e de Borda. Estes servidores estão presentes nas células do ambiente ubíquo, gerenciado pelo *middleware* EXEHDA, onde cada célula possui um Servidor Celular e pode conter vários Servidores de Borda.

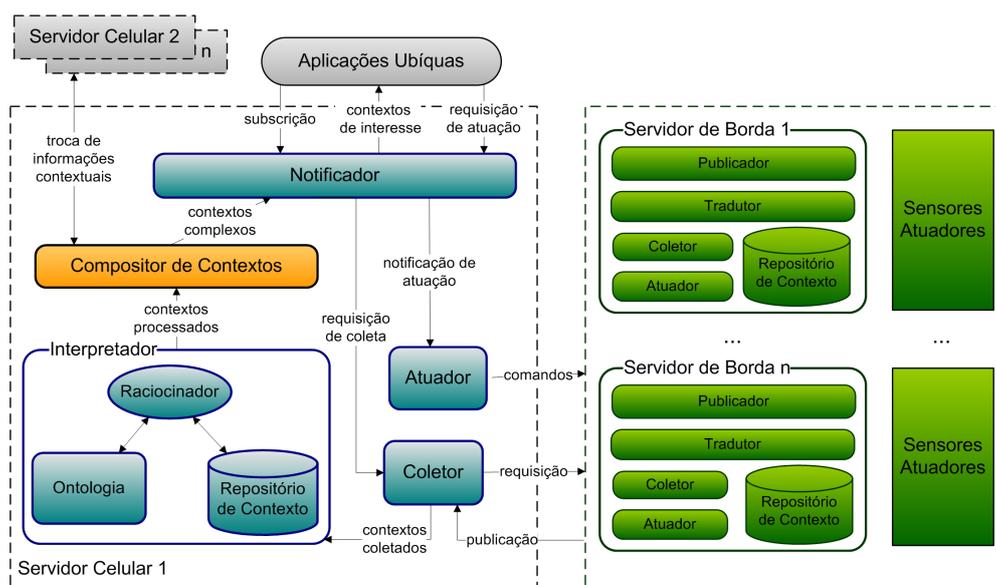


Figura 2. Arquitetura para Consciência do Contexto

Os serviços da arquitetura instalados no Servidor Celular são responsáveis por: (i) requisitar e coletar as informações de contexto publicadas pelos Servidores de Borda; (ii) manter o modelo do contexto (ontologia e banco de dados - Repositório de Contexto); (iii) registrar de forma permanente o histórico do contexto; (iv) interpretar o contexto, através do processamento das regras de inferência, com base no modelo de contexto utilizado pela arquitetura; (v) realizar a composição dinâmica das diferentes informações de contextos recebidas tanto dos Servidores de Borda como dos demais Servidores Celulares existentes no ambiente ubíquo, produzindo contextos de interesse complexos; (vi) notificar aos consumidores a ocorrência dos contextos de interesse subscritos pelos mesmos; (vii) construir comandos de atuação, os quais são enviados aos Servidores de Borda para serem processados.

Nos Servidores de Borda os serviços da arquitetura são responsáveis pela realização dos registros temporários dos dados de contexto e pelo processamento das regras de tradução relacionadas aos dados sensorizados. As regras de tradução envolvem operações com menor complexidade, não abrangendo o raciocínio e composição do contexto realizadas somente no Servidor Celular. Um exemplo de regra de tradução seria a conversão de escala de uma temperatura coletada. Ainda, nos Servidores de Borda estão os serviços responsáveis pela ativação, desativação e configuração dos atuadores empregados pelas aplicações. Além disso, a arquitetura disponibiliza um procedimento de suporte a falhas de comunicação nos Servidores de Borda, o qual persiste os dados adquiridos pelos sensores em um repositório de contexto local, garantindo sua disponibilização ao Servidor Celular quando forem restabelecidas as comunicações. A utilização de um servidor de contexto local pode permitir o tratamento de informações de contexto que sejam críticas ao ambiente, de forma independente do Servidor Celular.

## 6. Cenários de Uso

A proposta DynamiCC não é específica de um domínio de problema particular, mas sim é concebida para ser abrangente, buscando atender diferentes domínios. Nesse sentido, alguns dos cenários de uso, apresentados nesta seção, apesar de estarem relacionados a projetos de domínio específico (agropecuária), AMPLUS<sup>4</sup> e plenUS<sup>5</sup>, podem ser adaptados para outros domínios de aplicação, sem alteração tanto no modelo de contexto como na arquitetura de software.

O objetivo destes cenários de uso é avaliar, através de protótipos e testes, algumas características do modelo de contexto e da arquitetura de software, bem com as tecnologias empregadas, abrangendo os seguintes aspectos: (i) coleta de dados do meio e atuação sobre o mesmo; (ii) protocolo de comunicação entre os servidores; (iii) suporte a falhas de comunicação no Servidor de Borda; (iv) interpretação do contexto; e (v) composição de contextos dinâmicos.

As versões atuais dos servidores de Borda e Celular vem sendo desenvolvidas através de um processo de software baseado em prototipação [Fedeli et al. 2002], utilizando a linguagem de programação Java<sup>6</sup>, tendo em vista a perspectiva de integração com a API Jena<sup>7</sup> e a plataforma de serviços OSGi<sup>8</sup>. A API Jena é utilizada para processamento do modelo ontológico. Por sua vez, a plataforma OSGi é empregada para viabilizar a interação dinâmica dos componentes de software das aplicações, bem como a manutenção dos módulos (inclusão, alteração e atualização), sem que sua execução necessite ser interrompida.

### Cenário 1: Coleta, Atuação e Suporte a Falhas

O teste para as funcionalidades de coleta de dados de contexto e atuação sobre o meio pelo Servidor de Borda consistiu no monitoramento da temperatura na sala do Laboratório Didático de Análise de Sementes<sup>9</sup> e o acionamento de um diodo emissor de luz na

---

<sup>4</sup><http://amplus.ufpel.edu.br>

<sup>5</sup><http://plenus.ucpel.tche.br>

<sup>6</sup><http://www.java.com>

<sup>7</sup><http://incubator.apache.org/jena/>

<sup>8</sup><http://www.osgi.org>

<sup>9</sup><http://amplus.ufpel.edu.br/ldas>

sala do responsável pelo laboratório, para alertar que temperatura estava fora da faixa entre 15°C e 27°C. Para o processo de coleta e atuação foi utilizado, respectivamente, um sensor de temperatura digital e uma chave eletrônica endereçável, ambos baseados na tecnologia 1-Wire<sup>10</sup>. O módulo Publicador do Servidor de Borda foi configurado para realizar a publicação das temperaturas a cada 30 minutos no Servidor Celular.

Considerando o modelo de contexto e a arquitetura de software, a aplicação foi estruturada conforme o seguinte perfil: (i) Ambiente: Laboratório Didático de Análise de Sementes (LDAS); (ii) Coletor: Servidor de Borda LDAS; (iii) Aplicação: Monitoramento das Condições do Ambiente Físico do LDAS; (iv) Componente: Controle da Temperatura na Sala do LDAS; (v) Contexto de Interesse: Temperatura capturada por sensores digitais; (vi) Estados de Contexto: valores de temperatura coletados (limites para publicação:  $\geq 27^{\circ}\text{C}$  ou  $\leq 15^{\circ}\text{C}$ ); (vii) Evento: temperatura fora do limite; (viii) Regra: dispara o componente de software responsável pelo acionamento da luz de alerta na sala do responsável pelo LDAS.

Ainda, uma interrupção na conexão entre os servidores de Borda e Celular foi realizada com o intuito de testar a persistência no Servidor de Borda e a publicação dos dados temporariamente armazenados quando restabelecida a conexão.

## **Cenário 2: Comunicação entre os Servidores**

A comunicação entre os componentes de software distribuídos nos servidores de Borda e Celular ocorre através de serviços Web, implementados com XML-RPC<sup>11</sup>. Esta abordagem tem como principais características a simplicidade, interoperabilidade, facilidade de integração entre arquiteturas e estabilidade. Esses aspectos podem ser relevantes para aplicações ubíquas, devido a elevada heterogeneidade de plataformas encontradas nesse tipo de ambiente [Engelen et al. 2007].

## **Cenário 3: Interpretação do Contexto**

Com relação à interpretação do contexto, os testes realizados tiveram o objetivo de avaliar a representação formal do contexto através de ontologias, a capacidade de realizar inferências e a possibilidade de armazenar o histórico do contexto e fazer consultas sobre o mesmo.

A ferramenta Protégé<sup>12</sup> foi utilizada para modelar a ontologia, sendo esta representada em OWL-DL (*Description Logics*). Para verificar a consistência da ontologia e classificar a taxonomia foi utilizado o reasoner Pellet<sup>13</sup>, integrado ao próprio ambiente do Protégé. Como consequência, verificou-se que a ontologia não possuía fatos contraditórios, sendo gerada uma correta hierarquia de classes, inferindo superclasses e classes equivalentes de cada classe modelada.

Utilizando o módulo de persistência de dados da API Jena, o modelo ontológico foi armazenado no banco de dados PostgreSQL<sup>14</sup>, viabilizando a realização de consultas

---

<sup>10</sup><http://ubiq.inf.ufpel.edu.br/1-wire/>

<sup>11</sup><http://www.xmlrpc.com>

<sup>12</sup><http://protege.stanford.edu/>

<sup>13</sup><http://pellet.owldl.com/>

<sup>14</sup><http://www.postgresql.org/>

sobre o mesmo com a linguagem SPARQL<sup>15</sup>. Um cenário de uso foi construído para testar essa funcionalidade, relacionado ao projeto plenUS. O objetivo deste cenário foi realizar consultas na base de dados, empregando SPARQL. O contexto de interesse definido foi a disponibilidade de *slots* (canteiros em casas de vegetação, livres para instalação de um experimento) na Embrapa Clima Temperado em Pelotas. Essa informação de contexto tem como fonte uma aplicação responsável pela atualização das propriedades das instâncias da ontologia, utilizada por um usuário quando este libera o *slot*. Esse tipo de fonte de contexto caracteriza um sensoriamento virtual.

No modelo ontológico, Embrapa Clima Temperado é uma instância da classe “Lugar”, que descreve a instituição no município de Pelotas, e os *slots* são instâncias da classe “Recurso”. Assim, como uma consulta SPARQL retorna apenas dados explícitos, não tendo capacidade dedução, o resultado mostrou somente os *slots* livres associadas à Embrapa Clima Temperado e não a suas unidades específicas, que são instanciadas na classe “Célula”. Essa estrutura da ontologia modela exatamente o conhecimento do mundo real, por isso um processo de inferência é oportuno para obter informações que estão implícitas. Nesse sentido, foram criadas e executadas regras de inferência, utilizando a linguagem de regras e o *reasoner* disponibilizados pela API Jena. A regra de inferência criada para esse cenário de uso deduziu, em função dos relacionamentos e propriedades das classes na ontologia, que o contexto de interesse deveria ser expandido, incluindo a classe “Célula”, já que esta é uma especialização de “Lugar”. Com isso, as instâncias de outras unidades da Embrapa em Pelotas, instanciadas na classe “Célula”, como as estações experimentais Terras Baixas e Cascata, foram incluídas na consulta, logo o resultado da consulta pode mostrar mais *slots* disponíveis para o experimento.

#### **Cenário 4: Composição de Contextos Dinâmicos**

O cenário para avaliação desta funcionalidade da proposta DynamiCC foi construído no âmbito do projeto AMPLUS, tendo por base a mesma aplicação descrita no Cenário 1.

Para testar a composição dinâmica, o contexto de interesse foi alterado para incluir os fatores umidade e temperatura do ambiente externo. Este último fator é publicado por um Servidor de Borda, gerenciado pelo Servidor Celular localizado na célula da Embrapa-Terras Baixas, responsável pela estação agroclimatológica localizada no Campus Capão do Leão (campus da Universidade Federal de Pelotas onde está localizado o LDAS). Com isso, caracteriza-se a coleta de dados de contexto de forma distribuída, em diferentes células do ambiente ubíquo. Além disso, o cenário de uso buscou validar a alteração da regra de processamento em tempo de execução. Nesse caso, quando ocorreu um estado de contexto que dispara o evento relacionado a regra de processamento “dispara o componente de software responsável pelo acionamento da luz de alerta na sala do responsável pelo LDAS”, fora do horário de atividade do laboratório, essa regra foi substituída, em tempo de execução pela regra “dispara o componente de software para envio de e-mail para o responsável pelo laboratório”.

Nesse cenário, o Servidor de Borda LDAS publica os dados de contexto (temperatura e umidade) coletados na sala do laboratório e o módulo “Compositor de Contextos” do Servidor Celular, agrega a temperatura externa publicada pelo Servidor de Borda da

---

<sup>15</sup><http://www.w3.org/TR/rdf-sparql-query>

Estação Agroclimatológica e realiza a composição, considerando os estados de contexto, eventos e regras relacionados.

## 7. Considerações Finais

O principal diferencial da proposta DynamiCC para consciência do contexto em relação aos trabalhos relacionados diz respeito a maior expressividade na representação do contexto, pelo emprego de uma abordagem híbrida para modelagem do mesmo, bem como o suporte à interpretação e à composição de contextos dinâmicos, o que viabiliza a construção de contextos complexos, em tempo de execução das aplicações.

Dentre outros, na continuidade da pesquisa, os seguintes aspectos deverão ser explorados em trabalhos futuros: (i) revisar e ampliar os aspectos considerados nas aplicações alvo da proposta; (ii) especificar a API e os protocolos entre a camada de aplicação e o *middleware*; (iii) dar continuidade a especificação dos módulos da arquitetura.

## Referências

- Agostini, A., Bettini, C., and Riboni, D. (2009). Hybrid reasoning in the care middleware for context awareness. *Int. J. Web Eng. Technol.*, 5(1):3–23.
- Augustin, I., Yamin, A. C., and Silva, L. C. d. (2008). Building a smart environment at large-scale with a pervasive grid middleware. In Wong, J., editor, *Grid Computing Research Progress*, volume 1, chapter 10, pages 323–344. Nova Science, New York, NY, USA.
- Baldauf, M., Dustdar, S., and Rosenberg, F. (2007). A survey on context-aware systems. *Int. J. Ad Hoc Ubiquitous Comput.*, 2(4):263–277.
- Bettini, C., Brdiczka, O., Henriksen, K., Indulska, J., Nicklas, D., Ranganathan, A., and Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive Mob. Comput.*, 6(2):161–180.
- Chen, G., Li, M., and Kotz, D. (2008). Data-centric middleware for context-aware pervasive computing. *Pervasive Mob. Comput.*, 4(2):216–253.
- Cho, K., Hwang, I., Kang, S., Kim, B., Lee, J., Lee, S., Park, S., Song, J., and Rhee, Y. (2008). Hicon: a hierarchical context monitoring and composition framework for next-generation context-aware services. *Network, IEEE*, 22(4):34–42.
- Costa, C. A. (2008). *Continuum: A Context-aware Service-based Software Infrastructure for Ubiquitous Computing*. Tese (doutorado em ciência da computação), UFRGS, Porto Alegre, RS.
- Costa, C. A., Yamin, A. C., and Geyer, C. F. R. (2008). Toward a general software infrastructure for ubiquitous computing. *IEEE Pervasive Computing*, 7(1):64–73.
- Dey, A., Salber, D., and Abowd, G. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16.
- Dilli, R. M. (2010). *Uma Proposta para Descoberta de Recursos na Computação Ubíqua com Suporte Semântico*. Dissertação (mestrado em ciência da computação), UCPEL, Pelotas, RS.

- Engelen, R. V., Gallivan, K., Gupta, G., and Cybenko, G. (2007). Xml-rpc agents for distributed scientific computing. pages 168–182.
- Fedeli, R. D., Polloni, E. G. F., and Peres, F. E. (2002). *Orientação a Objeto com Prototipação*. Thomson.
- Ferry, N., Hourdin, V., Lavirotte, S., Rey, G., Riveill, M., and Tigli, J.-Y. (2011). Wcomp, a middleware for ubiquitous computing, ubiquitous computing. In Babkin, E., editor, *Ubiquitous Computing*, volume 1, chapter 8, pages 171–176. InTech.
- Ferry, N., Lavirotte, S., Tigli, J.-Y., Rey, G., and Riveill, M. (2009). Context adaptive systems based on horizontal architecture for ubiquitous computing. In *Proceedings of the 6th International Conference on Mobile Technology, Application and Systems, Mobility '09*, pages 10:1–10:8, New York, NY, USA. ACM.
- Kjeldskov, J. and Skov, M. B. (2007). Exploring context-awareness for ubiquitous computing in the healthcare domain. *Personal Ubiquitous Comput.*, 11(7):549–562.
- Lopes, J. L. B. (2008). *EXEHDA-ON: Uma Abordagem Baseada em Ontologias para Sensibilidade ao Contexto na Computação Pervasiva*. Dissertação (mestrado em ciência da computação), UCPEL, Pelotas, RS.
- Lopes, J. L. B., Pilla, M. L., and Yamin, A. C. (2007). Exehda: a middleware for complex, heterogeneous and distributed applications. *Iberian-American Conference on Technology Innovation and Strategic Areas*.
- Loureiro, A. A. F., Oliveira, R. A. R., de Moura Braga Silva, T. R., Júnior, W. R. P., de Oliveira, L. B. R., Moreira, R. A., Siqueira, R. G., Rocha, B. P. S., and Ruiz, L. B. (2009). Computação ubíqua ciente de contexto: Desafios e tendências. In *27º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC 2009 - Minicurso*, pages 99–149.
- Soldatos, J., Stamatis, K., Azodolmolky, S., Pandis, I., and Polymenakos, L. (2007). Semantic web technologies for ubiquitous computing resource management in smart spaces. *Int. J. Web Eng. Technol.*, 3(4):353–373.
- Tigli, J.-Y., Lavirotte, S., Rey, G., Hourdin, V., Cheung-Foo-Wo, D., Callegari, E., and Riveill, M. (2009). Wcomp middleware for ubiquitous computing: Aspects and composite event-based web services. *Annales des Télécommunications*, 64(3-4):197–214.
- Venecian, L. R. (2010). *EXEHDA-SS: Um Mecanismo para Sensibilidade ao Contexto com Suporte Semântico*. Dissertação (mestrado em ciência da computação), UCPEL, Pelotas, RS.
- Venturi Filho, S. N., Moura, A. M. d. C., and Cavalcanti, M. C. R. (2010). Armazenamento e manipulação de ontologias utilizando sistemas gerenciadores de banco de dados. Relatório técnico, Instituto Militar de Engenharia - IME.
- Warken, N. (2010). *Uma Proposta de Controle da Adaptação Dinâmica ao Contexto na Computação Ubíqua*. Dissertação (mestrado em ciência da computação), UCPEL, Pelotas, RS.
- Weiser, M. (1991). The computer for the 21st century. *Scientific American*, 3(265):94–104.