Gerenciamento proativo de redes de sensores na Ubicomp

Rodrigo Santos de Souza¹, João Lopes¹, Gizele Gadotti², Ana Marilza Pernas³, Adenauer Yamin³, Cláudio Geyer¹

Resumo: Um dos aspectos centrais da Computação Ubíqua (Ubicomp) está associado a uma forte integração entre os sistemas computacionais e o meio ambiente. Um dos principais desafios da pesquisa na área é fornecer mecanismos para a consciência de contexto que promovam o desenvolvimento de aplicações que reajam de acordo com a dinâmica do ambiente de interesse do usuário. Com essa motivação foi concebido o EXEHDA-SN, um novo serviço para o middleware EXEHDA. O Middleware EXEHDA vem sendo desenvolvido para promover a execução distribuída adaptativa, a coleta e o reconhecimento do contexto, armazenamento e acesso ubíquo e comunicações assíncronas. Nesse sentido, a principal contribuição deste artigo é a proposição do EXEHDA-SN como um serviço para o gerenciamento proativo das interações do Middleware com o ambiente físico, por meio de regras, empregando redes de sensores e atuadores. Para avaliar as funcionalidades do serviço proposto, apresentamos um estudo de caso na área agrícola, com destaque para os protótipos e testes realizados, cujos resultados se mostraram promissores.

Palavras-chave: Ubicomp. Consciência do Contexto. Rede de Sensores.

Abstract: One of the central aspects of Ubiquitous Computing (Ubicomp) is associated with a strong integration of computer systems and the environment. One of the main challenges of research in the area is to provide mechanisms for context-awar to promotes the development of applications that react according to the dynamics of user interest environment. With this motivation is designed the EXEHDA-SN, a new service for EXEHDA middleware. The EXEHDA Middleware has been developed to promote adaptive distributed execution, collect and recognition of context, storage and ubiquitous access and asynchronous communications. In this sense, the main contribution of this paper is the proposition of EXEHDA-SN as a service to the proactive management of the Middleware interactions with the physical environment, through rules, using sensors and actuators networks. To evaluate the functionalities of the proposed service, we present a case study in agriculture, highlighting the prototypes and tests performed. The results were promising.

Keywords: Ubicomp. Context Awareness. Sensor Networks.

1 Introdução

Uma das premissas da Computação Ubíqua é a busca pela integração dos seus serviços com os aspectos do ambiente [1]. Isso tem surgido com a intenção de viabilizar a reação das aplicações do usuário aos seus contextos de interesse [2]. Esse modelo de computação, conhecido como computação consciente do contexto, vem sendo considerado um dos pilares da Computação Ubíqua. A Ubicomp, proposta por Mark Weiser na década de 1990, indica que as interações do usuário com os sistemas computacionais devem ocorrer de maneira pouco intrusiva de modo que estes possam se concentrar em suas atividades de interesse [3].

Na consolidação da consciência do contexto um aspecto importante é o acesso às informações contextuais que devem ser coletadas em ambientes largamente distribuídos [4]. Isso aponta para a utilização de sensores de

http://dx.doi.org/10.5335/rbca.2015.4090

¹Instituto de Informática, Universidade Federal do Rio Grande do Sul, Caixa Postal: 15064 - 91501-970, Porto Alegre (RS) - Brasil {rssouza, geyer, jlblopes}@inf.ufrgs.br

²Centro de Engenharias, Universidade Federal de Pelotas, Caixa Postal: 354 - 96010-610, Pelotas - RS, Brasil gizele.gadotti@ufpel.edu.br

³Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Caixa Postal: 354 - 96010-610, Pelotas - RS, Brasil {marilza, adenauer}@ufpel.edu.br

diferentes naturezas alocados no ambiente, de forma a viabilizar a captura dessas informações. Grande parte dos desafios de pesquisa relacionados ao uso de redes de sensores na obtenção de informações contextuais, estão associados com a diferença entre os requisitos de alto nível das aplicações ubíquas e as operações básicas das redes de sensores [5]. A contribuição central deste artigo visa preencher essa lacuna, por meio da concepção do EXEHDA-SN (EXEHDA - Sensor Networks), um serviço para o Middleware EXEHDA (Execution Environment for Highly Distributed Applications) que tem por objetivo o tratamento de redes de sensores e atuadores. O EXEHDA-SN foi concebido para ser autonômico, gerenciado por regras e, capaz de agir de forma proativa, tanto na captura de informações contextuais do ambiente físico, como na atuação remota sobre esse.

A organização do texto contempla, na segunda seção, uma descrição do escopo de pesquisa deste trabalho. Na terceira seção, é apresentada a arquitetura proposta, detalhando os módulos que a constituem. O protótipo bem como os testes realizados são tratados na quarta seção. Na quinta seção são discutidos os trabalhos relacionados e, na sexta seção são apresentadas as considerações finais.

2 EXEHDA-SN: escopo da pesquisa

O Middleware EXEHDA vem sendo desenvolvido pelo grupo de pesquisa para dar suporte à execução das aplicações da Ubicomp. Sua arquitetura, baseada em serviços, visa criar e gerenciar um ambiente ubíquo, bem como promover a execução de aplicações sobre esse ambiente. Seu foco é permitir que as aplicações possam obter informações de seus contextos de interesse e reagir às variações que acontecem nos mesmos.

A estrutura de software do EXEHDA contempla um núcleo mínimo e serviços carregados sob demanda, os quais estão organizados nos seguintes subsistemas: (i) Reconhecimento do Contexto e Adaptação, (ii) Acesso Ubíquo, (iii) Execução Distribuída e (iv) Comunicação (vide Figura 1) [6]. As contribuições centrais do trabalho apresentado neste artigo estão associadas ao Subsistema de Reconhecimento do Contexto e Adaptação.

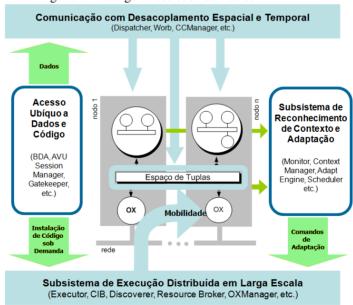


Figura 1: Visão geral dos subsistemas do EXEHDA

Na abordagem de gerenciamento de contexto proposta para o EXEHDA, as responsabilidades são organizadas em dois tipos de servidores: Servidor de Contexto e Servidor de Borda. O software do Servidor de Borda, elemento central da especificação do EXEHDA-SN, se destina a gerenciar a interação com o meio físico por intermédio de sensores e atuadores. O Servidor de Contexto, por sua vez, atua no armazenamento e processamento das informações contextuais [7].

3 EXEHDA-SN: concepção e modelagem

Na proposta do EXEHDA-SN o ambiente computacional é constituído por células em que se distribuem os dispositivos computacionais, conforme mostra a Figura 2. Os componentes básicos desse ambiente são: (*i*) o EXEHDAbase que consiste no elemento central da célula responsável por todos serviços básicos e referência para os demais elementos; (*ii*) o EXEHDAnodo que corresponde aos dispositivos computacionais responsáveis pela execução das aplicações; (*iii*) o EXEHDAnodo móvel, um subcaso do anterior, que corresponde aos dispositivos tipicamente móveis que podem se deslocar entre as células do ambiente ubíquo, como *notebooks*, *tablets* ou *smartphones*, por exemplo; e (*iv*) o EXEHDAborda que consiste nos dispositivos computacionais de borda do ambiente ubíquo, responsáveis por fazer a conexão física e lógica das redes de sensores e atuadores com os demais elementos que constituem a célula.

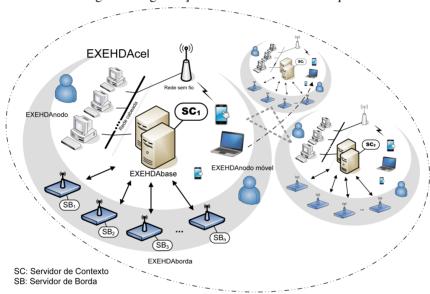


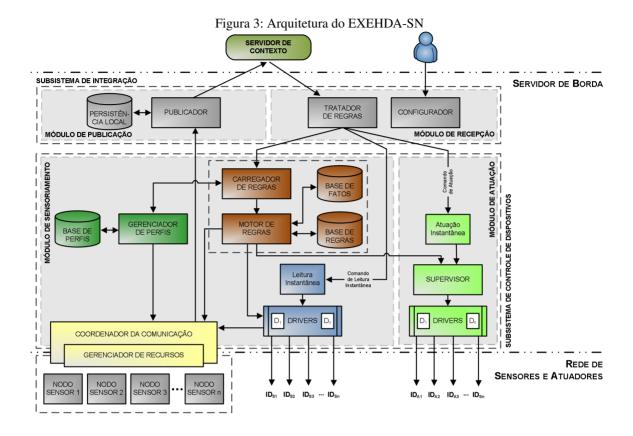
Figura 2: Organização celular do ambiente Ubíquo

O EXEHDA-SN tem sua arquitetura apresentada na Figura 3 com destaque para o Servidor de Borda. Nessa figura, além dos diversos módulos que constituem o Servidor de Borda está identificada a relação desses com o Servidor de Contexto e com os dispositivos que interagem com o ambiente físico. O Servidor de Borda é elemento central nos procedimentos de coleta e atuação do Middleware, tendo seu software instanciado em um equipamento do tipo EXEHDAborda. Sua arquitetura permite gerenciar diferentes dispositivos, como nodos sensores heterogêneos (sensores programáveis), sensores não programáveis e atuadores.

Os nodos sensores, além da capacidade de mensurar grandezas físicas, têm capacidade computacional, dispõem de Sistemas Operacionais embarcados e *frameworks* de programação. Tipicamente, os nodos sensores possuem severas restrições de recursos, como processamento, memória, banda de rede e energia [8]. Além disso, muitos nodos sensores diferem entre si em aspectos de hardware e software. Portanto, as estratégias de gerenciamento adotadas na arquitetura do Servidor de Borda procuram considerar as características intrínsecas de cada dispositivo que constitui a rede de sensores alvo.

Os sensores não programáveis são suportados pelo Servidor de Borda por meio de *drivers* que encapsulam os aspectos tecnológicos envolvidos no seu acesso. Para permitir a modificação de estados indesejados do meio físico, a arquitetura proposta também contempla o gerenciamento de atuadores, os quais são acessados pelo Servidor de Borda por meio de *drivers* específicos.

A arquitetura proposta tem por premissa atuar de forma autonômica tanto na captura e no processamento dos dados de contexto quanto na atuação sobre o meio, visto que os dados contextuais continuam a ser gerenciados mesmo nos períodos nos quais as aplicações interessadas no seu uso estejam inoperantes.



3.1 Gerenciamento Multinível

Grande parte dos desafios enfrentados ao longo dos anos quando da utilização de redes de sensores como provedoras de informações contextuais, estão associados à lacuna existente entre os requisitos de alto nível das aplicações ubíquas e as operação de gerência das redes de sensores [5]. Nesse sentido, na concepção do EXEHDA-SN foi adotada uma estratégia de gerenciamento em dois níveis a fim de promover a separação das preocupações.

O primeiro nível de gerenciamento é direcionado ao desenvolvedor de aplicações para o usuário final. Esse primeiro nível foi concebido tendo como premissa promover a abstração das características estruturais da rede de sensores a fim de oportunizar ao mesmo uma visão *top-down* do gerenciamento. Isso possibilita que o desenvolvedor possa concentrar-se nas demandas do usuário quando da implementação das funcionalidades da aplicação.

A estratégia de gerenciamento adotada nesse nível teve como base o modelo de regras proposto em [9], o qual foi concebido para tratar eventos considerando uma especificação do comportamento autônomo esperado por parte da rede de sensores. Esses eventos podem ser produzidos por alterações no tráfego de dados, frequência de aquisição e valores contextuais aquisitados. Todas estas informações, de forma combinada ou não, são tratadas por meio de regras, a quais são elaboradas considerando as necessidades da aplicação destinada ao usuário final. Nesse primeiro nível de gerenciamento todo processamento acontece no Servidor de Borda.

Os nodos sensores típicos possuem recursos escassos, o que implica na necessidade da utilização de códigos otimizados para a plataforma que se destinam [8]. Além disso, as redes de sensores podem ser constituídas por uma elevada quantidade de dispositivos heterogêneos distribuídos pelo ambiente a ser monitorado. Portanto, considerando a complexidade inerente a combinação de elevadas escalabilidade e heterogeneidade, seria oneroso ao desenvolvedor das aplicações ubíquas envolver-se diretamente na especificação dos códigos que devem estar em execução nos diferentes nodos sensores ativos no momento. Esses aspectos motivaram a concepção do segundo nível de gerenciamento que é direcionado ao usuário administrador do EXEHDA-SN.

Esse segundo nível está associado ao suporte automatizado necessário à execução dos processos nos nodos

sensores. Por meio desse nível de gerenciamento são especificados quais códigos devem ser disparados nos nodos sensores considerando sua atuação combinada com as correspondentes regras no Servidor de Borda. Portanto, tendo por base a necessidade de integração dos dois níveis de gerenciamento, o administrador do EXEHDA-SN quando da implementação dos códigos para os nodos sensores, deve considerar as relações existentes entre estes códigos e as regras correspondentes.

3.2 Tratamento da distribuição

O Subsistema de Integração do EXEHDA-SN foi concebido com o objetivo de promover a interoperação entre o Servidor de Borda e os demais serviços do *Middleware*. Além disso, por meio desse subsistema são oferecidos mecanismos para a configuração dos componentes da arquitetura. Suas tarefas são operacionalizadas pelos módulos de Publicação e de Recepção. O Módulo de Publicação é formado pelos componentes Publicador e Persistência Local, enquanto o Módulo de Recepção pelo Tratador de Regras e pelo Configurador.

O *Publicador* consiste no elemento que realiza o envio dos dados para as demais camadas do *middleware*, interoperando com a interface de aquisição do Servidor de Contexto. Considerando as possíveis falhas de comunicação entre o Servidor de Borda e o Servidor de Contexto se faz necessário um mecanismo de persistência para garantir o armazenamento temporário dessas informações até que sejam publicadas. O elemento da arquitetura desenvolvido para prover esse armazenamento é o componente *Persistência Local*. O componente *Tratador de Regras*, por sua vez, é responsável pelo recebimento dos comandos e das regas provenientes do Servidor de Contexto bem como pelo encaminhamento desses aos respectivos componentes da arquitetura.

Todas as configurações necessárias para o funcionamento do Servidor de Borda são operacionalizadas por meio do componente *Configurador*. Por intermédio deste componente é disponibilizada uma interface Web através da qual é possível gerenciar a remoção e inclusão de sensores e atuadores, configurar *drivers* de dispositivos, gerenciar as regras de controle, configurar o endereço do Servidor de Contexto para a publicação dos dados sensoreados, entre outros.

3.3 Tratamento de sensores e atuadores por meio de regras customizáveis

O Subsistema de Controle de Dispositivos viabiliza a operação da rede de sensores e atuadores considerando as especificidades de cada dispositivo, tendo suas funcionalidades organizadas nos Módulos de Sensoriamento e de Atuação.

O *Motor de Regras* é o componente responsável pelo processamento das regras submetidas pelo Servidor de Contexto. As regras suportadas pelo Servidor de Borda são armazenadas na Base de Regras. A arquitetura do Servidor de Borda não se se restringe a um conjunto específico de regras. Conforme a necessidade das aplicações ubíquas o conjunto de regras pode ser expandido através da inclusão de novas elementos na Base de Regras e na Base de Perfis. Para servir de suporte ao processamento das regras, o componente Base de Fatos tem por função registrar alguns aspectos necessários às execuções das mesmas, como por exemplo o estado dos sensores e os eventos que já ocorreram.

Para adequar-se ao dinamismo do ambiente que se reflete no comportamento das aplicações da Ubicomp, na concepção do Servidor de Borda foram utilizados mecanismos de reprogramação dinâmica. Por meio dessa abordagem pretende-se adaptar o comportamento dos processos em execução nos nodos sensores conforme as demandas das aplicações. Na arquitetura proposta, esse processo é realizado de forma coordenada entre os componentes Carregador de Regras, Gerenciador de Perfis e Base de Perfis. Toda regra submetida ao Servidor de Borda é avaliada pelo componente *Carregador de Regras* o qual tem a função de ativar na arquitetura os elementos necessários ao seu processamento. Se a regra envolver apenas sensores não programáveis, os quais são acessados por meio de *drivers* específicos, ela é imediatamente encaminhada ao *Motor de Regras* para o seu processamento. No caso da regra envolver nodos sensores, o Carregador de Regras informa o componente *Gerenciamento de Perfis* quanto aos códigos disponíveis na *Base de Perfis* que devem ser distribuídos aos nodos sensores para dar suporte a tal regra. Assim que todos os códigos estiverem em execução nos nodos sensores, o Carregador de Regras submete a nova regra ao Motor de Regras para que este possa processá-la.

3.4 Tratamento da heterogeneidade

Os *Drivers*, como componentes arquiteturais, são responsáveis pelo acesso aos valores das grandezas físicas medidas pelos sensores bem como pelos comandos de atuação enviados aos atuadores. Eles encapsulam e controlam os sensores e atuadores de maneira individualizada, o que evita que as diferenças operacionais desses dispositivos se projetem nos demais componentes da arquitetura.

O componente *Leitura Instantânea* tem o objetivo de permitir a leitura de um determinado sensor sob demanda das aplicações a qualquer momento. O componente faz a recepção assíncrona das solicitações e, a partir do ID do sensor, dispara o *driver* correspondente.

Pela natureza das redes de sensores, é esperado que sejam executadas atividades colaborativas entre os processos em execução nos nodos [10][11]. Por isso, torna-se importante a existência de um mecanismo que simplifique a comunicação entre os nodos sensores e desses com o Servidor de Borda. No modelo do Servidor de Borda, essa funcionalidade é operacionalizada através do componente *Coordenador da Comunicação*. Esse componente implementa um modelo de coordenação baseado na abstração de Espaço de Tuplas [12] que simplifica a programação das aplicações, abstraindo aspectos de baixo nível associados com a troca de informações entre os dispositivos envolvidos.

Em redes de sensores os dispositivos estão constantemente entrando e saindo da rede, seja por decorrência de término de energia, perda de sinal de rádio ou pela inserção de novo dispositivo, devido à reposição por avaria ou necessidade de adicionar outro ponto de monitoramento [13]. Neste sentido, o componente *Gerenciador de Recursos* tem a função de administrar esses eventos, mantendo a consistência da infraestrutura da rede como um todo.

O Módulo de Atuação do Servidor de Borda, cuja estrutura é apresentada na Figura 3, agrupa os componentes que fazem o controle dos atuadores. O componente *Atuação Instantânea* tem um funcionamento análogo ao componente de Leitura Instantânea. Ele recebe comandos com o ID do atuador e os correspondentes padrões de operação (tempo de duração, potência de ativação, etc.), os quais são repassados ao componente Supervisor para tratamento.

O componente *Supervisor* aglutina os comandos de atuação. Uma vez recebidos os parâmetros para controle da atuação, o componente Supervisor, após avaliar eventuais conflitos entre as regras oriundas das diferentes fontes, ativa o respectivo *driver* do atuador envolvido.

4 Estudo de caso

Um dos critérios considerados na concepção do serviço EXEHDA-SN foi o não comprometimento com um cenário de uso específico. Desse modo, embora o cenário de uso tratado neste artigo seja relacionado com a área da agricultura, entende-se que o EXEHDA-SN pode ser utilizado em outros domínios de aplicação.

Esta seção resume os principais aspectos do estudo de caso realizado para avaliar as funcionalidades do EXEHDA-SN. O estudo de caso inclui tarefas relativas à detecção, coleta, processamento e notificação de contexto. Segundo [14], a transparência introduzida pelo uso de middleware em aplicações conscientes do contexto implica uma dificuldade quanto à avaliação das funcionalidades propostas. Para o autor, uma das principais estratégias utilizadas para avaliar a capacidade do middleware é através de aplicações experimenteis em que as experiências dos usuários podem ser exploradas de maneira explicita ou implícita. Nesse sentido, as funcionalidades do EXEHDA-SN foram avaliadas pelos utilizadores por meio do método TAM [15].

O estudo de caso realizado foi conduzido procurando atender as demandas do Projeto AMPLUS ⁴ (*Automatic Monitoring and Programable Logging Ubiquitous System*) que tem por finalidade fornecer serviços conscientes do contexto para os equipamentos do Laboratório Didático de Análise de Sementes (LDAS) ⁵. O LDAS integra o Programa de Pós Graduação em Ciência e Tecnologia de Sementes da Universidade Federal de Pelotas (UFPEL).

A qualidade das análises realizadas no LDAS depende diretamente da utilização correta dos equipamentos e da aplicação de métodos e procedimentos uniformes. Porém, grande parte dos equipamentos do Laboratório

⁴http://amplus.ufpel.edu.br

⁵http://amplus.ufpel.edu.br/ldas

tem volume interno relativamente pequeno (em torno de 340 litros), portanto se existir uma grande diferença entre o ambiente interno e externo, a variação das condições internas é rápida devido a pouca inércia térmica. Isso exige, entre outros aspectos, o monitoramento e controle da temperatura e umidade das sementes ao longo de todo período de análise e ação rápida no caso de algum dos valores excederem as faixas especificadas. Nessa perspectiva, o Projeto AMPLUS mostrou-se oportuno para avaliar características importantes do *CA Service*. Os aspectos avaliados nesse cenário foram: (*i*) coleta de dados do ambiente; (*ii*) atuação pró-ativa sobre o meio; e (*iii*) publicação dos dados sensoreados no Servidor de Contexto.

4.1 Prototipação e testes

O protótipo do Servidor de Borda foi escrito em Phyton sobre o Sistema Operacional Raspbian, sendo usado como hardware a Raspberry PI ⁶. A tecnologia central utilizada na implementação do Tratador de Regras e do Publicador foi XML-RPC⁷. O Motor de Regras executa em *loop* todas as regras existentes na Base de Regras. A leitura de um dado sensor é disparada sempre que uma das condições definidas nas regras é satisfeita. Esta leitura é efetivada por *drivers* específicos que tratam cada sensor individualmente, segundo as particularidades tecnológicas de cada um. As regras de gerenciamento do Servidor de Borda, que controlam a leitura dos sensores, a publicação dos seus valores no Servidor de Contexto bem como a supervisão dos atuadores, também são escritas em Phyton.

A fim de avaliar as funcionalidades da arquitetura proposta, optou-se por utilizar no LDAS um conjunto de dispositivos constituído por nodos sensores e por sensores não programáveis. Os sensores não programáveis selecionados para o estudo de caso são baseados na tecnologia 1-Wire⁸. Essa tecnologia caracteriza-se como uma rede de transmissão de dados, baseada em dispositivos eletrônicos endereçáveis, e tem se destacado por sua versatilidade e facilidade de implementação. Por sua vez, os nodos sensores adotados nos testes são do tipo Telos, revisão B com sistema operacional Contiki ⁹, os quais têm sido amplamente utilizados em pesquisas envolvendo redes de sensores. O Telos faz parte da linha de *motes* desenvolvida pela Universidade da Califórnia, Berkeley, e vem sendo fabricado pela empresa Memsic ¹⁰.

Para monitorar as variáveis físicas mais importantes para a operação do LDAS foram utilizados 15 sensores 1-Wire e 3 nodos sensores Telos B. Para explorar a característica reativa da arquitetura, foi utilizado um alerta luminoso, que é acionado sempre que, no processamento de uma regra de contexto de interesse, é identificada a necessidade da atenção dos laboratoristas para com algum dos equipamentos. Conforme as regras estabelecidas à operação do LDAS, os dados sensoreados são publicados periodicamente no Servidor de Contexto para que sejam armazenados para posterior utilização. Os processamentos realizados pelo Servidor de Borda se dão de forma autônoma, mantendo-se operacional independentemente das aplicações interessadas estarem em execução.

Nesse cenário de testes, foram utilizadas regras que tratam condições que contemplam dois critérios distintos: (i) critério de tempo, em que a ação é disparada em função da passagem de um tempo especificado, o qual é usado para publicar dados contextuais periodicamente para registro histórico; (ii) critério de valor, em que uma ação é disparada quando um contexto de interesse (temperatura) extrapolou uma determinada faixa de valores, sendo que nesse caso a ação consiste no acionamento de um alerta luminoso seguido da publicação do valor sensoreado e o envio de e-mail ou SMS.

Para acompanhar o comportamento da arquitetura, foi desenvolvida uma ferramenta para a visualização dos valores das variáveis físicas coletados pelos sensores gerenciados pelo Servidor de Borda. A interface da ferramenta possibilita a seleção do contexto de interesse a ser exibido, disponibilizando um relatório textual com os dados coletados pertinentes a última semana (vide Figura 4). Na parte superior da ferramenta desenvolvida é disponibilizado um menu que permite selecionar os outros modos de visualização oferecidos.

Na Figura 5, é apresentado o modo gráfico da ferramenta desenvolvida, em que é possível visualizar simultaneamente as curvas de variação dos valores de vários sensores utilizados no LDAS. A seleção dos sensores a serem visualizados é feita a partir de um menu com suporte a múltipla seleção. Também é disponibilizado um recurso de inspeção que permite a comparação dos valores em um determinado instante do tempo. A janela de

⁶http://www.raspberrypi.org

⁷http://www.xmlrpc.com

⁸http://www.maximintegrated.com

⁹http://www.contiki-os.org/

¹⁰http://www.memsic.com/

Figura 4: Projeto AMPLUS: relatório textual dos dados sensoriados



tempo dos dados que estão sendo visualizados pode ser definida pelo usuário por meio da mesma interface gráfica que exibe os valores sensoreados. Esse módulo da ferramenta possibilita ao pesquisador do laboratório comparar as variações de temperatura e umidade ocorridas durante o processo de germinação das sementes.

Projeto AMPLUS LDAS Variável de Contexto Controlada: Comparação entre sensores Início Tabela Estatísticas Temperatura Zoom 1d 1s 1m 1a All Início Out 2, 2013 Fim Out 7, 2013 do, Out 5, 18:15-18:29 30.36°C 4: 28.40°C +22.5°C +17.5°C 12:00 00:00 12:00 00:00 00:00 12:00 00:00 12:00

Figura 5: Projeto AMPLUS: visualização gráfica dos dados sensoriados

Por demandas das pesquisas realizadas no LDAS foi concebido um módulo para a consulta de dados contextuais a partir do cruzamento de múltiplos sensores através de diferentes regras. A manipulação dessas regras e de seus parâmetros é feita por meio do módulo conforme é apresentado na Figura 6, sendo disponibilizado como resultado os valores que satisfazem as condições submetidas e alguns dados estatísticos a eles associados.

A comparação dos contextos de interesse registrados, visualizados por intermédio da ferramenta desenvolvida, com os demais eventos produzidos pelo protótipo do Servidor de Borda (acionamento do alerta luminoso,



Figura 6: Projeto AMPLUS: cruzamento de dados sensoriados

envio de e-mail e SMS recebidos) possibilitou avaliar o comportamento da arquitetura. Durante o período analisado, considerando as definições estabelecidas nas regras utilizadas, foi possível observar que todos os eventos monitorados do ambiente tem sido identificados pela arquitetura.

4.2 Avaliação de aceitação

Esta seção apresenta um detalhamento do experimento e os resultados obtidos com a avaliação de aceitação da ferramenta desenvolvida. O estudo envolveu dez voluntários, entre professores, alunos e técnicos, com atividades relacionadas ao LDAS. Cada participante utilizou um desktop para acessar a ferramenta. Após a realização de um treinamento básico, os participantes utilizaram a ferramenta de visualização e responderam um questionário de avaliação, considerando a experiência de uso.

O questionário foi construído com base no Modelo de Aceitação de Tecnologia (TAM), usando uma escala de Likert [15]. Para a aceitação da ferramenta, o modelo TAM considera: (i) Facilidade de uso: grau em que o usuário avalia que a ferramenta pode reduzir seu esforço; e (ii) Percepção de utilidade: grau em que o usuário avalia que a ferramenta pode melhorar a sua experiência.

As Tabelas 1 e 2 contêm, respectivamente, o questionário aplicado aos usuários e as respostas obtidas para facilidade de uso e percepção de utilidade. Em ambas as tabelas, a primeira coluna corresponde a questão, as seguintes cinco colunas apresentam os resultados obtidos em cada escala, em graus relativos e absolutos, e a última coluna mostra a média consolidada da percentagem, variando de 0 a 5.

Analisando os resultados, pode-se observar que a solução desenvolvida para o LDAS obteve aprovação, tanto para facilidade de uso, como para percepção de utilidade. Entretanto, ocorreram resultados na escala "indiferente" nas duas últimas questões da percepção de utilidade. Isso pode ser interpretado como uma preocupação com o controle da qualidade dos experimentos desenvolvidos no LDAS, em função do uso de mecanismos autônomos, sem a usual intervenção humana, para a emissão de alertas para estados contextuais que exijam uma atuação imediata. Nesse caso, uma estratégia que pode ser adotada é intensificar os testes e validações com os usuários e

Tabela 1: Avaliação da facilidade de uso

Questão	Discordo	Discordo	Indiferente	Concordo	Concordo	Média
	Totalmente	Parcial-		Parcial-	Totalmente	
		mente		mente		
 A ferramenta é fácil de entender. 	0,0%(0)	0,0%(0)	0,0%(0)	40,0%(4)	60,0%(6)	4,6
2. A ferramenta é fácil de usar.	0,0%(0)	0,0%(0)	0,0%(0)	30,0%(3)	70,0%(7)	4,7
3. As opões são claras e objetivas.	0,0%(0)	0,0%(0)	10,0%(1)	20,0%(2)	70,0%(7)	4,6
4. Com pouco esforço consigo se-	0,0%(0)	0,0%(0)	0,0%(0)	20,0%(2)	80,0%(8)	4,8
lecionar um contexto de interesse.						
5. Com pouco esforço consigo	0,0%(0)	0,0%(0)	0,0%(0)	30,0%(3)	70,0%(7)	4,7
acessar os relatórios gráficos.						

Tabela 2: Avaliação da percepção de utilidade

Questão	Discordo	Discordo	Indiferente	Concordo	Concordo	Média
	Totalmente	Parcial-		Parcial-	Totalmente	
		mente		mente		
As opções apresentadas são rele-	0,0%(0)	0,0%(0)	0,0%(0)	30,0%(3)	70,0%(7)	4,7
vantes.						
2. A ferramenta facilita a obtenção	0,0%(0)	0,0%(0)	0,0%(0)	40,0%(4)	60,0%(6)	4,6
de dados de contexto envolvendo						
múltiplos sensores.						
3. A ferramenta facilita a atuação	0,0%(0)	0,0%(0)	30,0%(3)	30,0%(3)	40,0%(4)	4,1
imediata a partir da emissão de um						
alerta ou mensagem.						
4. Eu usaria essa ferramenta no meu	0,0%(0)	0,0%(0)	30,0%(3)	20,0%(2)	50,0%(5)	4,2
trabalho.						

iniciar uma implantação gradativa das aplicações.

5 Trabalhos Relacionados

Tendo como base as premissas que motivaram o desenvolvimento da arquitetura proposta nesse artigo, o estudo da literatura da área proporcionou identificar alguns trabalhos relacionados, dentre os quais foram selecionados os seguintes: CARE [16], CoCA [17], HiCon [18], Solar [19], WComp [20]. Os aspectos considerados importantes na seleção de tais trabalhos foram: (i) suporte a redes de sensores e atuadores; (ii) aquisição autônoma dos dados de contexto; (iii) suporte ao tratamento de regras; e (iv) suporte à atuação distribuída sobre o meio.

A arquitetura proposta para o EXEHDA-SN foi concebida de forma a gerenciar redes de sensores e atuadores. Com isso, pode ser otimizado o gerenciamento tanto da aquisição dos dados de contexto a partir de vários tipos de sensores, usual nos ambientes computacionais para provimento de aplicações ubíquas, como na atuação distribuía sobre o meio físico. Tal característica é encontrada, em parte, nos projetos CoCA e HiCon, que têm suporte a redes de sensores. O projeto WComp, por sua vez, permite atuação sobre o meio, entretanto, não suporta o gerenciamento de redes de atuadores.

Com exceção dos projetos CARE e Solar, os demais preveem o emprego de mecanismos específicos para aquisição do contexto. Esses mecanismos adotam uma estratégia de separação entre a obtenção e o uso do contexto. Essa estratégia também é adotada no trabalho apresentado neste artigo, pois a arquitetura proposta para o Servidor de Borda gerencia a coleta e atuação sobre o meio físico de maneira integrada ao Servidor de Contexto, porém é auto gerida. Desta maneira, proporciona outro diferencial em relação aos projetos relacionados, pois atua de forma autônoma na aquisição dos dados de contexto, ou seja, de forma independente das aplicações interessadas.

O suporte ao tratamento de regras é encontrado na maioria dos trabalhos identificados na literatura, porém distribuição desse tratamento entre os Servidores de Contexto e de Borda é um diferencial em relação aos demais projetos. Enquanto no Servidor de Contexto são tratadas regras elaboradas que fazem o cruzamento de diferentes informações, incluindo dados históricos, os quais demandam maior poder computacional, o Servidor de Borda trata as regras de contingência. Esta funcionalidade de processamento de contexto, nos trabalhos relacionados,

usualmente está restrita a um único equipamento.

6 Considerações finais

Este artigo resume os esforços de pesquisa associados à concepção do EXEHDA-SN, um novo serviço do Middleware EXEHDA para coleta e pré-tratamento das informações contextuais com suporte à respectiva atuação sobre o meio. O serviço proposto é baseado em regras e atua de forma proativa em relação aos eventos dos contextos de interesse.

A principal contribuição deste trabalho é a proposição de um serviço para a coleta e atuação, que possibilita gerenciar de maneira autonômica e através de regras, dispositivos de diferentes naturezas (sensores não programáveis, nodos sensores e atuadores) e tecnologias (de hardware, software básico e comunicação). A estratégia adotada para o EXEHDA-SN resultou em um serviço que ampliou o escopo de uso do Middleware EXEHDA, possibilitando sua utilização em diferentes cenários.

A avaliação da arquitetura proposta se deu por meio de um protótipo desenvolvido para suprir demandas do Projeto Amplus, desenvolvido utilizando o EXEHDA, o qual vem sendo adotado pelo Laboratório Didático de Análise de Sementes da Faem/Ufpel. O estudo de caso vem apresentando resultados promissores tanto no que tange a concepção do modelo arquitetural, como às tecnologias utilizadas. Os resultados positivos apresentados na avaliação de aceitação da proposta, realizada com os usuários do projeto Amplus, estão sendo fundamentais para a consolidação da pesquisa desenvolvida.

Como trabalho futuro, um segundo estudo de caso, envolvendo viticultura de precisão, está sendo avaliado. Nesse sentido, foi realizado um estudo bibliográfico a fim de identificar as principais demandas atuais dessa área. Baseado no estudo realizado, um cenário de testes vem sendo concebido. As observações iniciais vêm indicando que as demandas da viticultura de precisão vão ao encontro das funcionalidades propostas para a arquitetura do EXEHDA, apontando para resultados promissores na continuidade do trabalho que vem sendo realizado.

Referências

- [1] CACERES, R.; FRIDAY, A. Ubicomp Systems at 20: Progress, Opportunities, and Challenges. *IEEE Pervasive Computing*, v. 11, n. 1, p. 14–21, 2012. ISSN 1536-1268. Disponível em: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6109229>. Acesso em 30 out. 2015.
- [2] KRUMM, J. et al. Ubiquitous Computing Fundamentals. [S.1.]: Chapman & Hall/CRC, 2010.
- [3] WEISER, M. The Computer for the 21st Century. Scientific American, v. 265, n. 3, p. 94–104, sep 1991.
- [4] PERERA, C. et al. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys and Tutorials*, v. 16, n. 1, p. 414–454, jan 2014.
- [5] WANG, M. et al. Middleware for Wireless Sensor Networks: A Survey. *Journal of Computer Science and Technology*, v. 23, n. 3, p. 305–326, jun 2008. ISSN 1000-9000. Disponível em: http://www.springerlink.com/index/10.1007/s11390-008-9135-x. Acesso em 30 out. 2015.
- [6] LOPES, J. L. et al. A model for context awareness in Ubicomp. In: *Proceedings of the 18th Brazilian symposium on Multimedia and the web WebMedia '12*. New York, New York, USA: ACM Press, 2012. p. 161.
- [7] LOPES, J. et al. Towards a Distributed Architecture for Context-Aware Mobile Applications in UbiComp. In: *Brazilian Symposium on Multimedia and the Web (WebMedia)*. Salvador: [s.n.], 2013. p. 43–49.
- [8] FAROOQ, M. O.; KUNZ, T. Operating systems for wireless sensor networks: a survey. *Sensors (Basel, Switzerland)*, v. 11, n. 6, p. 5900–30, jan 2011. ISSN 1424-8220. Disponível em: . Acesso em 30 out. 2015.">http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3231431{&}tool=pmcentrez{&}rendertype=ab>. Acesso em 30 out. 2015.
- [9] TERFLOTH, K. A Rule-Based Programming Model for Wireless Sensor Networks. Tese (Doutorado) Freie Universität Berlin, 2009.

- [10] CHOE, Y.-H.; KELLY, T.; ADOLPH, M. Ubiquitous Sensor Networks (USN) ITU-T Technology Watch Report 4. [S.1.], 2008.
- [11] MOTTOLA, L.; PICCO, G. P. Programming Wireless Sensor Networks: Fundamental Concepts and State of the Art. *ACM Computing Surveys*, v. 43, n. 3, p. 1–51, apr 2011. ISSN 03600300. Disponível em: http://portal.acm.org/citation.cfm?doid=1922649.1922656>. Acesso em 30 out. 2015.
- [12] SOUZA, R. S. D. et al. Um Modelo de Coordenação Escalável e Proativo para Aplicações Ubíquas. In: *Simpósio Brasileiro de Computação Ubíqua e Pervasiva (SBCUP)*. [S.l.: s.n.], 2012.
- [13] YICK, J.; MUKHERJEE, B.; GHOSAL, D. Wireless sensor network survey. *Computer Networks*, v. 52, n. 12, p. 2292–2330, aug 2008. ISSN 13891286. Disponível em: http://linkinghub.elsevier.com/retrieve/pii/S1389128608001254>. Acesso em 30 out. 2015.
- [14] KNAPPMEYER, M. et al. Survey of Context Provisioning Middleware. *IEEE Communications Surveys & Tutorials*, v. 15, n. 3, p. 1492–1519, jan 2013.
- [15] YOON, C.; KIM, S. Convenience and TAM in a ubiquitous computing environment: The case of wireless LAN. *Electronic Commerce Research and Applications*, v. 6, n. 1, p. 102–112, mar 2007.
- [16] AGOSTINI, A.; BETTINI, C.; RIBONI, D. Hybrid reasoning in the CARE middleware for context awareness. *International Journal of Web Engineering and Technology*, v. 5, n. 1, p. 3–23, 2009.
- [17] EJIGU, D.; SCUTURICI, M.; BRUNIE, L. Hybrid Approach to Collaborative Context-Aware Service Platform for Pervasive Computing. *Journal of Computers*, v. 3, n. 1, p. 40–50, jan 2008.
- [18] CHO, K. et al. HiCon: a hierarchical context monitoring and composition framework for next-generation context-aware services. *IEEE Network*, v. 22, n. 4, p. 34–42, jul 2008.
- [19] CHEN, G.; LI, M.; KOTZ, D. Data-centric middleware for context-aware pervasive computing. *Pervasive and Mobile Computing*, v. 4, n. 2, p. 216–253, apr 2008.
- [20] TIGLI, J.-Y. et al. WComp middleware for ubiquitous computing: Aspects and composite event-based Web services. *annals of telecommunications annales des télécommunications*, v. 64, n. 3-4, p. 197–214, jan 2009.