

Explorando o Processamento Semântico para Sensibilidade ao Contexto na Computação em Grade

Luthiano Venecian, Sérgio Rodrigues,
Adenauer Yamin
PPGINF-Centro Politécnico
Universidade Católica de Pelotas
{venecian,sergior,adenauer}@ucpel.tche.br

Adenauer Yamin
Centro de Desenvolvimento Tecnológico
Universidade Federal de Pelotas
adenauer@ufpel.edu.br

João Ladislau Lopes
Instituto Federal Sul-rio-grandense
jlopes_cavg@ifsul.edu.br

Claudio Geyer
Instituto de Informática
Universidade Federal do Rio Grande do Sul
geyer@inf.ufrgs.br

Abstract

The diversity of devices and information in a grid computing environment introduces different challenges for interoperability between the parties involved. When you build and run context-aware applications, emerging requirements should be fulfilled, involving the acquisition of contextual information, representation, and processing. With the aim of meet these requirements of the large scale distributed computing, we designed the EXEHDA-SS, which is responsible for processing the contextual information, performing reasoning and context manipulation, using ontologies in the semantic representation and processing of contextual information. The proposed solution was evaluated on a case study aimed at the medical field whose features have been established in order to explore the proposed mechanism of context-awareness with semantic support.

1. Introdução

A Computação em Grade é uma forma de computação onde o processamento está espalhado entre os dispositivos que integram o ambiente computacional, que executam tarefas bem definidas dependendo de sua natureza, interligados de forma que essa estrutura exija pouco gerenciamento por parte do usuário final. Para isto, as aplicações executam em ambientes distribuídos instrumentados com sensores, nos quais dispositivos, agentes de software e serviços são integrados de forma transparente e cooperam para atender aos objetivos das aplicações. As aplicações da Computação em Grade caracterizam-se por constantes mudanças em

seu estado de execução, geradas pelos ambientes altamente dinâmicos em que executam [11].

Um ambiente de grade computacional tem uma natureza dinâmica, devido a variedade de dispositivos e tecnologias existentes, assim como às mudanças constantes na disponibilidade de equipamentos no ambiente computacional [21]. Ao se construir e executar aplicações sensíveis ao contexto na Computação em Grade, há uma série de funcionalidades que devem ser providas, envolvendo desde a aquisição de informações contextuais, a partir de um conjunto de fontes heterogêneas e distribuídas, até a representação dessas informações, seu processamento, e a realização de inferências para seu uso em tomadas de decisão.

Este artigo apresenta uma proposta de um mecanismo de sensibilidade ao contexto, desde a sua aquisição, processamento e distribuição das informações contextuais, direcionado à ambientes de computação distribuída, empregando para isto tecnologias de suporte semântico.

O EXEHDA (*Execution Environment for Highly Distributed Applications*) [20] é um *middleware* baseado em serviços que visa criar e gerenciar um ambiente distribuído, bem como promover a execução, sob este ambiente, das aplicações que expressam a semântica siga-me. Estas aplicações são distribuídas e adaptativas ao contexto em que seu processamento ocorre, estando disponíveis a partir de qualquer lugar, todo o tempo [10].

O EXEHDA-SS abrange o tratamento das informações contextuais, realizando tarefas de manipulação e dedução sobre o contexto, utilizando ontologias para suporte a representação e processamento das informações contextuais. Através do uso de inferências espera-se garantir um refinamento qualificado dessas informações capturadas e dis-

tribuídas nas células de execução do EXEHDA.

O artigo está organizado nas seguintes seções: a seção 2 trata da concepção e modelagem do EXEHDA-SS; a seção 3 apresenta os trabalhos relacionados; a seção 4 descreve um estudo de caso; e a seção 5 apresenta as considerações finais.

2 Concepção e Modelagem do EXEHDA-SS

Esta seção apresenta as linhas gerais do mecanismo de sensibilidade ao contexto proposto para o *middleware* EXEHDA. Este mecanismo está integrado ao Subsistema de Reconhecimento de Contexto e Adaptação do *middleware* [21], e tem como contribuição central oferecer um servidor de contexto com suporte semântico para o processamento das informações contextuais.

As seções subsequentes apresentam, respectivamente, o modelo de representação de contexto do ambiente de grande computacional e a arquitetura de software proposta para sensibilidade ao contexto personalizável por componentes de software das aplicações com suporte a processamento semântico do EXEHDA-SS.

2.1 Modelo de Representação de Contexto

Na concepção do EXEHDA-SS foi selecionado o uso de ontologias como tecnologia para representação e processamento das informações contextuais, atendendo especificamente as demandas do servidor de contexto do EXEHDA-SS [16]. As ontologias são empregadas sob duas perspectivas: (i) representação semântica dos dados contextuais e (ii) o estabelecimento de relações entre os mesmos, possibilitando a realização de inferências. Através do uso de inferências espera-se contribuir para qualificação das informações contextuais a serem entregues aos demais serviços do EXEHDA e/ou aplicações que tenham interesse em manipulação de dados contextuais.

O núcleo do modelo semântico para representação de contexto é a ontologia “OntContext”, definida para ser responsável pela representação dos contextos coletados, notificados e das instâncias dos Contextos de Interesse das Aplicações. A OntContext, figura 1, é empregada pelo servidor de contexto do EXEHDA-SS para prover suporte semântico ontológico nas tarefas de dedução, utilizando informações contextuais.

A linguagem OWL foi escolhida para construção do modelo ontológico da OntContext, utilizando a sub-linguagem OWL-DL [9]. Os principais aspectos considerados para seleção é seu suporte a metadados RDF, abstrações de classes, generalização, agregação, relações de transitividade, simetria e detecção de inconsistências [17].

Na estrutura de classes da OntContext foi definida a classe ‘Sensor’, a qual contém as instâncias dos sensores que integram o monitoramento do ambiente de grade computacional provido pelo *middleware* EXEHDA. A seguir são detalhadas as demais classes e sub-classes da OntContext.

Classes, Subclasses e Atributos para Coleta e Notificação de Contextos

- Contexto: a classe ‘contexto’ armazena os dados coletados pelo sensores como: identificador do nodo, data/hora, usuário, identificador do contexto de interesse, sensor e valor coletado pelo sensor. Este armazenado está localizado no RIC - Repositório de Informações Contextuais.
- Contexto_Notificado: contém os seguintes atributos: identificador, aplicação, componente, adaptador e usuário, no qual as informações contextuais processadas e deduzidas pelo Motor de Inferência são persistentes no RCN - Repositório de Contextos Notificados. Compõe o relacionamento desta classe as subclasses: ContextoNotificado_Sensor e ContextoNotificado_Deduzido.
- ContextoNotificado_Sensor: esta subclasse possui identificação e o sensor que gerou uma notificação, além do valor coletado pelo respectivo sensor, identificador do nodo e valor traduzido.
- ContextoNotificado_Deduzido: são armazenados nesta subclasse os identificadores das regras de dedução e valores deduzidos.

Classes e Atributos do Contexto de Interesse das Aplicações

- Contexto_Interesse: contém o identificador, aplicação, componente e adaptador.
- Sensor_Public: define os parâmetros operacionais dos sensores, tais como: intervalo do tempo de medicação, taxa de flutuação mínima dos dados a serem publicados, valor inferior e superior, valor *default* do sensor e número máximo de dados contextuais armazenados. Também ficam instanciados nesta classe as regras de tradução a serem processadas pelo Gerente de Aquisição.
- Contexto_Deduzido: compõem esta classe os atributos identificador e a regra de dedução. Essas regras serão processadas pelo Motor de Inferência.

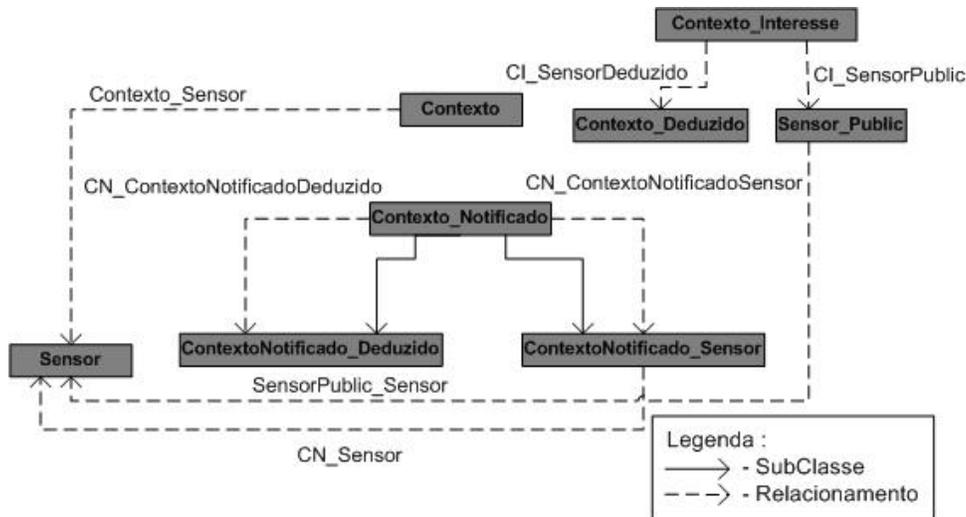


Figura 1. Classes da OntContext

2.2 Arquitetura de Software Proposta

O EXEHDA-SS é alimentado por Contextos de Interesses das Aplicações, os quais são responsáveis por caracterizar os aspectos que devem ser considerados nos procedimentos de monitoração do ambiente de grade computacional, de interpretação destes dados capturados e das respectivas notificações.

Uma visão geral da arquitetura de software do EXEHDA-SS é ilustrada na figura 2. O servidor de contexto é composto por três serviços: Gerente de Aquisição, Gerente de Interpretação e Gerente de Notificação. Esses serviços são autônomos e cooperantes para a realização de tarefas de manipulação e dedução sobre o contexto.

A descrição da arquitetura proposta para o EXEHDA-SS é apresentada a seguir.

Gerente de Aquisição

O Gerente de Aquisição tem a função central de prover a captura de informações de contexto, disponibilizando as mesmas em um formato adequado para que o Gerente de Interpretação possa implementar suporte semântico. Para a aquisição de contextos através de sensores e publicação de dados a partir dos mesmos se faz necessário: (i) especificar intervalos de tempo entre medições; (ii) flutuação mínima para que aconteça a publicação; (iii) definir a faixa na qual os valores dos sensores deverão ser publicados.

Neste sentido, foi definido na OntContext, a classe *Contexto_Interesse*, onde cada contexto de interesse possui, por aplicação, uma relação de sensores com seus parâmetros operacionais, regras para dedução, tradução das informações coletadas por estes sensores e o máximo de

instâncias armazenadas dos sensores no RIC.

Quando de uma publicação, o serviço Tradutor e Instanciador Contextual presente neste gerente, identifica os contextos de interesse pertencentes ao sensor em execução, processa as regras de tradução definidas pelo programador da aplicação e armazena no RIC, representado ontologicamente pela classe *Contexto*.

No RIC são instanciadas as informações coletadas dos sensores, tais como: identificador do sensor, valor coletado pelo sensor, valor que tenha ocorrido tradução contextual, contexto de interesse pertencente ao sensor, nome do nodo, usuário e horário da coleta.

Por fim, após obter os contextos de interesse pertencentes a uma determinada publicação, o Gerente de Interpretação é acionado disponibilizando os contextos de interesse para a execução do Motor de Inferência.

Gerente de Interpretação

As principais funções previstas para este gerente são: (i) manter o RIC, que armazena os contextos capturados pelo Gerente de Aquisição; (ii) utilizar um Motor de Inferência para processamento e dedução sobre as informações de contexto mantidas no RIC e nos Contextos de Interesses das Aplicações; (iii) alimentar o RCN, que armazena os estados dos contextos disponibilizados pelo Gerente de Notificação.

O Motor de Inferência, identifica os sensores e as regras definidas pertencentes ao contexto de interesse repassado pelo Gerente de Aquisição e executa dois possíveis processamentos:

- Leitura no RIC dos valores coletados pelos sensores envolvidos do respectivo contexto de interesse em

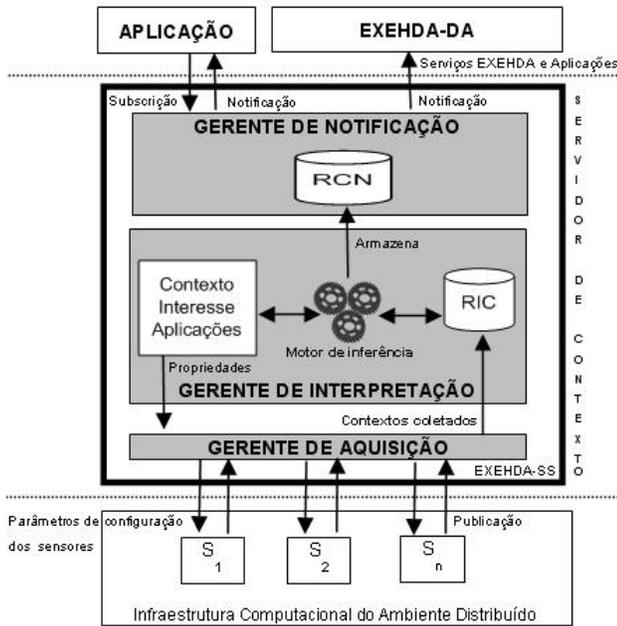


Figura 2. Visão Geral da Arquitetura de Software do EXEHDA-SS

processamento - inferência baseada em Ontologias - Máquina de Inferência *OWLReasoner* [8]. Caso ainda não tenha ocorrido uma publicação de informação contextual pelos sensores envolvidos do respectivo contexto de interesse, é passado o valor *default* do sensor, definido na classe *Sensor_Public*;

- Dedução de regras definidas pelo desenvolvedor na classe *Contexto_Deduzido* da *OntContext* produzindo desta forma contextos deduzidos - inferência baseada em regras - Máquina de Inferência *Generic rule reasoner* [8].

Finalizando o processo, o Gerente de Interpretação, instancia no RCN localizado no Gerente de Notificação, os valores dos contextos e as deduções processadas pelo Motor de Inferência. A figura 3 apresenta o fluxo de interpretação contextual do motor de inferência do EXEHDA-SS.

Gerente de Notificação

Este gerente é responsável por disponibilizar os contextos processados pelo Gerente de Interpretação. As principais funções deste gerente são: (i) notificação de informações contextuais ao serviço de adaptação dinâmica do EXEHDA - EXEHDA-DA [18]; (ii) receber subscrições pelas aplicações e notificá-las de acordo com as solicitações realizadas.

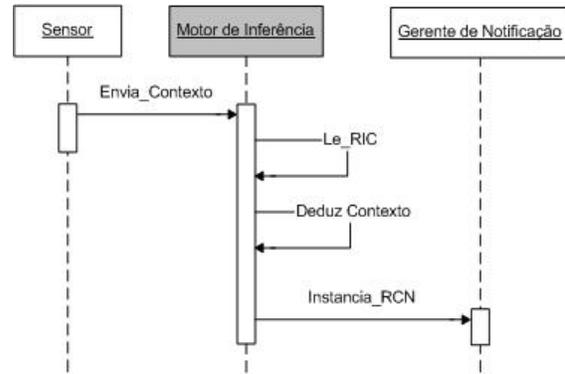


Figura 3. Fluxo do Motor de Inferência do EXEHDA-SS

A notificação para o serviço de adaptação dinâmica do *middleware* EXEHDA ocorre através do envio da instância do contexto notificado ao mesmo. O serviço de adaptação ao receber a instância que contém alterações contextuais no ambiente de grade computacional, realiza uma leitura dos demais valores no contexto notificado da *OntContext* e inicia as tarefas de execução previstas em seu processamento.

Outra funcionalidade prevista para o EXEHDA-SS diz respeito à realização de subscrições no Gerente de Notificação pelas aplicações. Nesse sentido, a aplicação deve invocar o método *ExehdaSS_Subscricao(app,userId,idSubscricao)*, passando três parâmetros: código da aplicação, usuário e o identificador do nó que solicitou a subscrição. O subscritor presente no Gerente de Notificação, recebe esses parâmetros e dispara uma leitura no RCN encaminhando para a aplicação as informações da respectiva subscrição - vide figura 4. Esta forma de execução definida para o **Subscritor** faz com que o serviço de sensibilidade ao contexto do EXEHDA-SS possa ser utilizado diretamente pelas aplicações.

```
<?xml version="1.0" encoding="UTF-8"?>
<EXEHDA-SS>
<Contexto_Notificado UsuarioId="AFD10-4589E">
<Aplicacao AplicacaoId="100">
<Sensor SensorId="100" SensorValor="180" />
<Sensor SensorId="101" SensorValor="28°" />
<Sensor SensorId="102" SensorValor="190110" />
</Aplicacao>
</Contexto_Notificado>
</EXEHDA-SS>
```

Figura 4. Exemplo de Notificação as Aplicações

3 Trabalhos Relacionados

Esta seção apresenta trabalhos relacionados ao EXEHDA-SS. Foram avaliadas arquiteturas de onze projetos, cujas características constituem um conjunto representativo do que vêm sendo desenvolvido nos últimos anos na direção de infraestruturas de suporte a sensibilidade ao contexto em ambientes distribuídos. Sendo eles: (A) *Context Management System* [22]; (B) *Context Toolkit* [5]; (C) *Middleware* de Contexto do Gaia [14]; *Social Philanthropic Information Environment* [1]; (E) *Context Aware Mobile Networks and Services* [19]; (F) *Service-Oriented Context-Aware Middleware* [6]; (G) *Context Broker Architecture* [4]; (H) *Mobile Collaboration Architecture* [15]; (I) *Framework* de Contexto [7]; (J) *Semantic Context Kernel* [3]; e (K) *Infraware* [12].

Em linhas gerais, o estudos dos trabalhos relacionados além de prover uma aproximação com as tecnologias empregadas na concepção do EXEHDA-SS, possibilitou uma sistematização das características a serem consideradas quanto da concepção dos componentes da arquitetura, em particular seus serviços, pelos seguintes aspectos apresentados na tabela 1.

Assim como o EXEHDA-SS, oito dos trabalhos relacionados avaliados empregam ontologias para representação das informações contextuais, caracterizando o uso de ontologias como uma tendência na área.

O emprego de ontologias permitiu ao EXEHDA-SS, enquanto servidor de contexto, realizar o processamento de regras de dedução dos dados contextuais, sendo que quatro dos trabalhos avaliados possuem um mecanismo de dedução em suas arquiteturas, porém nenhum deles contempla a definição de regras externas à aplicação pelo desenvolvedor como empregado pelo EXEHDA-SS.

A adoção da linguagem de consulta SPARQL no EXEHDA-SS se mostrou peça central na proposição do mecanismo para processamento das informações contextuais coletadas e armazenadas no RIC. Dentre os trabalhos avaliados, os *middlewares Context Broker Architecture*, *Semantic Context Kernel* e *Infraware*, utilizam esta mesma estratégia no processamento das informações coletadas.

O mecanismo de dedução baseado em ontologias para o processamento contextual proposto para o EXEHDA-SS é baseado na API JENA [8]. A seleção da API JENA é decorrente de sua significativa presença na literatura especializada, sendo também utilizada em três *middlewares* entre os trabalhos relacionados, o *Context Broker Architecture*, *Middleware* de Contexto do Gaia e *Semantic Context Kernel*.

Outrossim, a utilização da API JENA no EXEHDA-SS, que atua como interpretador, permite que as regras para tradução e dedução possam ser modificadas dinamicamente, sem interromper a execução da aplicação. Esta característica diferencia o EXEHDA-SS por não ser con-

templada nos trabalhos relacionados considerados.

4 Estudo de Caso: Acompanhamento de Pacientes

O EXEHDA-SS está inserido nos esforços de pesquisa do Projeto PertMed (Sistema de TeleMedicina Móvel) [13], o qual prevê que os procedimentos da área médica sejam atendidos por dispositivos distribuídos, bem como por dispositivos concebidos especificamente para medicina, como por exemplo os *holters* [2], que tem a função especializada de coleta de dados.

Este estudo de caso consiste em uma aplicação sintética direcionada à área médica denominada AP - Acompanhamento de Pacientes, cujas funcionalidades foram concebidas com o intuito de explorar o mecanismo de sensibilidade ao contexto com suporte semântico em um ambiente hospitalar.

Os objetivos contemplados na aplicação AP são: (i) exibir dados de pacientes adquiridos dinamicamente por mecanismo de sensoriamento de sinais (pressão alta e frequência cardíaca); (ii) emitir, de forma automatizada, diferentes níveis de alertas, em função dos dados sensorados, para os agentes de saúde (médicos, enfermeiros); e (iii) permitir acesso ao histórico dos dados sensorados dos pacientes por agentes de saúde. Compreende-se que na AP, os sensores assim como os profissionais de saúde podem estar localizados nos mais diversos lugares.

A AP foi projetada para ser executada com o suporte do serviço de adaptação dinâmica do *middleware*. Nesse cenário, o EXEHDA-SS através de seus gerentes, recebe os dados publicados pelos sensores, realiza o processamento e notifica aos demais serviços do *middleware* para que possam ser realizadas as adaptações inerentes a execução da aplicação.

A seguir serão apresentadas as configurações da AP na OntContext e a publicação e processamento dos contextos sensorados pelo EXEHDA-SS.

4.1 Configuração da AP na OntContext

Ao configurar a aplicação no modelo ontológico do EXEHDA-SS, o desenvolvedor necessita habilitar seus contextos de interesse, especificando (i) os parâmetros operacionais e as regras de tradução dos sensores de monitoramento de pacientes (pressão alta e frequência cardíaca), e (ii) as regras de dedução, para a funcionalidade de Envio Automático de Mensagens.

Neste estudo de caso foram definidos dois parâmetros operacionais para publicação dos sensores de monitoramento de pacientes. O primeiro relacionado ao intervalo de medição: pressão alta a cada duas horas e frequência cardíaca a cada cinco minutos. E o segundo a faixa de flutuação mínima: pressão alta um percentual e frequência cardíaca cinco percentuais.

Tabela 1. Comparação entre o EXEHDA-SS e os Trabalhos Relacionados

	A	B	C	D	E	F	G	H	I	J	K	EXEHDA-SS
Aquisição	X	X	X	X	X	X	X	X	X	X	X	X
Compartilhamento			X			X	X					X
Contexto de Interesse das Aplicações	X											X
Interpretação								X		X	X	X
Mecanismo de Dedução			X			X	X			X		X
Notificação		X	X	X	X	X			X			X
Ontologias			X		X	X	X	X	X	X	X	X

As regras de tradução e dedução empregadas utilizam uma sintaxe de construção de regras adotada pelo subsistema de inferência da API JENA. O raciocinador baseado em regras *Generic rule reasoner* suporta a criação de regras definidas pelo desenvolvedor da aplicação.

Deste modo, foram definidas regras de tradução para caracterizar que a informação sensorada apresenta sinais normais ou não. Portanto, pressão alta superior a 180090 (180 por 90 mmHg) gera um valor traduzido para 1, caso inferior para 0. Frequência cardíaca superior a 100 (100 bpm), é traduzido para 1 ou inferior, para 0 - vide figura 5 e 6 respectivamente.

Sensor de Pressão Alta
[PA_ForaFaixa: (?c rdf:type ont: Contexto) (?c ont:Contexto_Sensor 102) (?c ont:Contexto_Valor ?cv) (greater (?cv,180090)) -> (?n rdf:type ont 1)]
[PA_DentroFaixa: (?c rdf:type ont: Contexto) (?c ont:Contexto_Sensor 102) (?c ont:Contexto_Valor ?cv) (lessThan (?cv,180090)) -> (?n rdf:type ont 0)]

Figura 5. Regra de Tradução - Sensor de Pressão Alta

Sensor de Frequência Cardíaca
[[FC_ForaFaixa: (?c rdf:type ont: Contexto) (?c ont:Contexto_Sensor 100) (?c ont:Contexto_Valor ?cv) (greaterThan (?cv,100)) -> (?n rdf:type ont 1)]
[FC_DentroFaixa: (?c rdf:type ont: Contexto) (?c ont:Contexto_Sensor 100) (?c ont:Contexto_Valor ?cv) (lessThan (?cv,101)) -> (?n rdf:type ont 0)]

Figura 6. Regra de Tradução - Sensor de Frequência Cardíaca

Para o Envio Automático de Mensagens definiu-se uma regra de dedução para risco de infarto, onde foi considerado os seguintes parâmetros: sensor de frequência cardíaca com batimentos superiores a 180, e pressão arterial superior a 240100, conforme apresentado na figura 7.

Regra de Dedução - Risco de Infarto
[FC: (?c rdf:type ont: Contexto) (?c ont:Contexto_Sensor 100) (?c ont:Contexto_Valor ?cv) (greaterThan (?cv,180)) -> (?c rdf:type ont FCsup)]
[PA: (?c rdf:type ont: Contexto) (?c ont:Contexto_Sensor 102) (?c ont:Contexto_Valor ?cv) (greaterThan (?cv,240100)) -> (?c rdf:type ont PAsup)]
[RI: (?c rdf:type ont:FCsup) (?c rdf:type ont:PAsup) -> (?c rdf:type ont Risco_Infarto)]

Figura 7. Regra de Dedução para Risco de Infarto

4.2 Publicação e Processamento de Contextos Sensorados pelo EXEHDA-SS

Para disparo dos sensores de monitoramento de pacientes definidos para a AP, o Gerente de Aquisição processa um arquivo de configuração, que contém seus parâmetros operacionais (intervalo de medição e faixa de flutuação - descrito na seção 4.1) para publicação de informações a serem tratadas pelo EXEHDA-SS.

Esta publicação ocorre através da conexão periódica dos sensores ao Gerente de Aquisição, permitindo a instanciação das informações contextuais no RIC de acordo com que os dados sensorados. Desta forma, ao armazenar esses valores coletados, o Gerente de Aquisição também processa as regras de tradução apresentadas na seção 4.1 deste artigo.

O motor de inferência ao ser acionado, realiza uma lei-

tura dos demais dados dos sensores especificados no monitoramento de pacientes e instancia no RCN. Este por sua vez, notifica o serviço de adaptação dinâmica para que possam ser produzidos os diferentes níveis de alertas na AP.

A associação entre os dados coletados e os níveis de alertas em função do sensoramento são resumidos a seguir:

- Nível de Alerta Verde: sinais normais;
- Nível de Alerta Laranja: início de problema, frequência cardíaca ou pressão arterial fora do normal;
- Nível de Alerta Vermelho: alerta máximo, frequência cardíaca e pressão arterial fora do normal.

A figura 8 mostra as telas de acordo com nível de alerta.

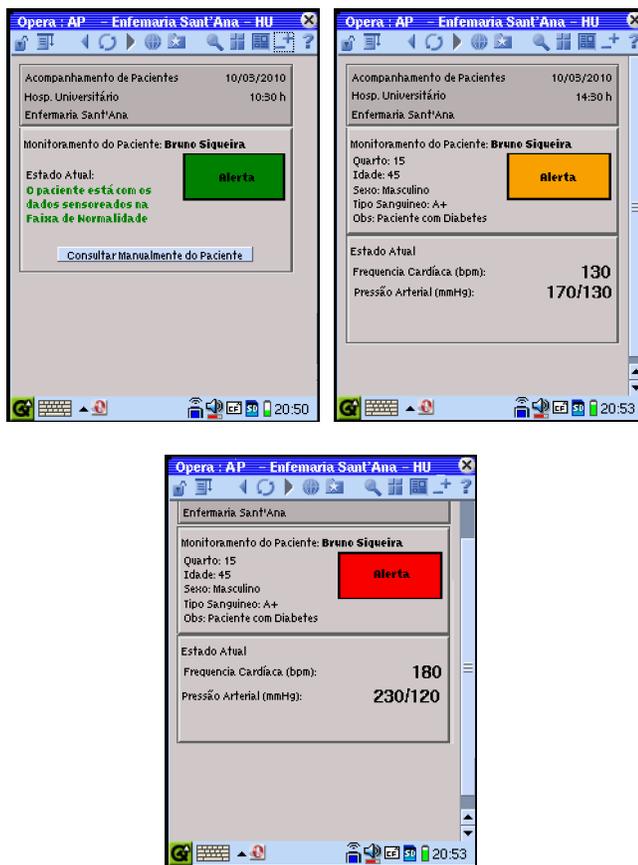


Figura 8. Níveis de Alertas: Verde, Laranja e Vermelho, respectivamente

Uma das características presentes na arquitetura de software do EXEHDA-SS é que as aplicações podem efetuar uma subscrição e serem notificadas diretamente. Neste sentido, para que a AP possa disparar Envio Automático de Mensagens (SMS, email, etc), o usuário interessado, deve se registrar para receber notificações produzidas pelo processamento da regra de dedução descrita na seção 4.1.

5. Considerações Finais

Na computação distribuída em larga escala um aspecto fundamental relaciona-se ao monitoramento e a manipulação das informações contextuais. Deste modo, a Computação Sensível ao Contexto é um paradigma computacional que se propõe a permitir que as aplicações tenham acesso e tirem proveito de informações contextuais que digam respeito às computações que realizam.

Neste sentido, um sistema é dito sensível ao contexto, se ele usa o mesmo para prover informações ou serviços ao usuário. Suas aplicações são capazes de modificar seu comportamento baseado nas informações de contexto.

A utilização de tecnologias para processamento semântico no tratamento de informações de contexto traz como características: (i) a descrição formal, padrão e estruturada de cada dimensão semântica de informação de contexto; (ii) o suporte à interoperabilidade sintática, estrutural e, principalmente, semântica entre aplicações sensíveis ao contexto; e (iii) a capacidade de interpretar e inferir inter-relacionamentos com base nos conteúdos e descrições semânticas das entidades envolvidas.

Considerando estas premissas, foi concebido um modelo de representação ontológico das informações de contexto, que foi modelado para ser expansível, tanto no que diz respeito a especificação dos aspectos pertinentes a captura de dados do ambiente de grade computacional, como quanto ao registro dos possíveis consumidores de contextos de interesse. Os consumidores podem ser as aplicações que se subscrevem no servidor de contexto, bem como outros serviços do *middleware* EXEHDA, dentre os quais destaca-se o serviço de adaptação dinâmica ao contexto.

As atividades desenvolvidas ao longo deste trabalho, tendo por base os estudos realizados nos trabalhos relacionados, a construção de um modelo de representação ontológica, a concepção de um mecanismo sensível ao contexto, permitiram a obtenção das seguintes contribuições: (i) concepção de um servidor de contexto, que contempla três serviços para atender as tarefas de aquisição de informações contextuais, sua interpretação e notificação de contextos de interesse; (ii) definição de parâmetros operacionais para ativação e publicação de informações pertinentes as aplicações; (iii) permitir a especificação de contextos de interesse individualizado por aplicação, os quais são representados ontologicamente na OntContext; (iv) expansibilidade do modelo ontológico a outros domínios; e (v) manipulação e dedução sobre dados contextuais através do Motor de Inferência baseado em regras definidas pelo desenvolvedor das aplicações, em tempo de execução, o que diferencia o EXEHDA-SS dos demais projetos relacionados analisados. Entende-se por tempo de execução a flexibilidade da arquitetura de software reconfigurar as regras para manipulação e processamento das informações contextuais

dinamicamente.

Assim, o EXEHDA-SS mostra-se alinhado com o objetivo de prover suporte semântico nas tarefas de aquisição, interpretação e notificação das informações contextuais aos demais serviços do EXEHDA e/ou as aplicações. O emprego de ontologias se mostrou oportuno para prover o suporte semântico necessário a proposta do EXEHDA-SS.

Dentre os aspectos levantados para continuidade do trabalho destacam-se: (i) proposição de um mecanismo baseado em banco de dados para persistência de dados factíveis de serem coletados; (ii) inclusão de outros parâmetros para controle operacional dos sensores levando em conta o poder computacional dos mesmos, e suas restrições quanto ao consumo de energia; e (iii) expandir o mecanismo para construção de contextos que englobem várias células de execução do ambiente de grade computacional provido pelo *middleware* EXEHDA.

Referências

- [1] R. Belotti. *Sophie - Context Modelling and Control*. Diploma thesis, Swiss Federal Institute of Technology Zurich, 2004.
- [2] F. S. Brito. *Eletrocardiografia ambulatorial sistema holter, fibrilação atrial, depressão do segmento ST e isquemia*. Reblampa, 2007.
- [3] R. F. Bulcao Neto and M. G. C. Pimentel. *Toward a Domain-Independent Semantic Model for Context-Aware Computing*. In: Proceedings of the 3rdIW3C2 Latin American Web Congress, IEEE Computer Society, 2005.
- [4] H. Chen, T. Finin, A. Joski, and Y. Peng. *UMBC eBiquity Project: Context Broker Architecture (CoBrA)*. <http://ebiquity.umbc.edu/project/html/id/1/> - Acesso em 11/2009, 2005.
- [5] A. Dey. Understanding and using contex. pages 04–07. Personal UbiComp, 2001.
- [6] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang. *An Ontology-based Context Model in Intelligent Environments*. In: Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, 2005.
- [7] K. Henriksen and J. Indulska. *Developing Context-Aware Pervasive Computing Applications: Models and Approach*. In: Pervasive and Mobile Computing Journal, 2005a.
- [8] Jena. Jena - a semantic web framework for java, 2009. Disponível em: < <http://jena.sourceforge.net/> >. Acesso em maio de 2010.
- [9] H. Knublauch, A. L. Rector, N. Drummond, and M. HorridgeE. Owl pizzas-practical experience of teaching owl-dl, 2004. Common Errors Common Patterns. Springer, 2004. 63-81p.
- [10] J. L. Lopes, M. L. Pilla, and A. C. Yamin. Exehda: a middleware for complex, heterogeneous and distributed applications. *Iberian-American Conference on Technology Innovation and Strategic Areas*, Maio 2007.
- [11] F. Magoules, J. Pan, K. Tan, and A. Kumar. *Introduction to Grid Computing*, volume 10. CRC Press, 2009.
- [12] J. G. Pereira Filho, R. M. Pessoa, C. Z. Calvi, N. Q. Oliveira, R. R. M. Carmo, A. C. P. Barbosa, C. R. G. Farias, and M. M. Leite. *Infraware: um Middleware de Suporte a Aplicações Móveis Sensíveis ao Contexto*. In: SBRC - Simpósio Brasileiro de Redes de Computadores, 2006.
- [13] PERTMED. Pertmed - sistema de telemedicina móvel, 2010. Disponível em: <<http://gmobserver02.inf.ufsm.br:8082/features/projetos.html>>. Acesso em maio de 2010.
- [14] M. Roman, C. K. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahrstedt. *Gaia: A middleware infrastructure to enable active spaces*. IEEE Pervasive Computing, 2003.
- [15] V. Sacramento and M. e. a. Endler. Moca: A middleware for developing collaborative applications for mobile users. *IEEE Distributed Systems Online*, 5(10), 2004.
- [16] L. R. Venecian. *EXEHDA-SS: Um Mecanismo para Sensibilidade ao Contexto com Suporte Semântico*. In: <http://olaria.ucpel.tche.br/luthiano/> - Acesso em 03/10, 2010.
- [17] W3C. *OWL Web Ontology Language*. In: <http://www.w3.org/TR/owl-guide/> - Acesso em 02/10, 2010.
- [18] N. Warken. *Uma Contribuição ao Controle da Adaptação na Computação Ubíqua*. In: <http://olaria.ucpel.tche.br/nelsiw/> - Acesso em 07/09, 2009.
- [19] M. Wegdam. *AWARENESS: a project on Context AWARE Networks and ServiceS*. Proceedings of the 14th Mobile & Wireless Communications Summit 2005, 19-23, 2005.
- [20] A. Yamin. *Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva*. Tese (doutorado em ciência da computação), Instituto de Informática, UFRGS, Porto Alegre, RS, 2004.
- [21] A. Yamin and I. Augustin. Exehda: Adaptive middleware for building a pervasive grid environment. pages 203–219. Frontiers in Artificial Intelligence and Applications - Self-Organization and Autonomic Informatics, 2005.
- [22] A. Zimmermann, M. Specht, and A. Lorenz. *Personalization and Context Management*. User Modeling and User-Adapted Interaction, 2005b.