

A Semantic-Based Software Architecture for Processing Context Information in Ubiquitous Medicine

Sérgio Rodrigues, Renato Dilli
Instituto Federal Sul-rio-grandense
Pelotas, RS
{sergio,dilli}@ifsul.edu.br

Luthiano Venecian, Adenauer Yamin
Universidade Católica de Pelotas
Pelotas, RS
{venecian,adenauer}@ucpel.tche.br

Cristiano Costa, Jorge Barbosa
Universidade do Vale do Rio dos Sinos
São Leopoldo, RS
{cac,jbarbosa}@unisinis.br

João Lopes, Rodrigo Souza, Cláudio Geyer
Universidade Federal do Rio Grande do Sul
Porto Alegre, RS
{jlblopes,rssouza,geyer}@inf.ufrgs.br

Abstract

Ubiquitous medicine aims to provide healthcare services at any time, without restriction of location, enhancing the mobility of health professionals. Considering this premise, in this paper we propose a software architecture model that integrates sensors and computing devices (mobile or fixed) in a ubiquitous environment, providing context-awareness service. This model, called uMED¹, uses semantic processing technologies in the contextual information acquisition and processing. To evaluate the uMED, we present a case study related to clinical monitoring of patients.

1 Introdução

A computação ubíqua (Ubicomp) prevê acesso a sistemas de informação, independente de tempo, localização e equipamento. Essa premissa vem sendo materializada pela disseminação de dispositivos portáteis, com acesso a redes sem-fio. Nesse sentido, o uso da Ubicomp vem sendo potencializado em diversas áreas do mundo real, ampliando a usabilidade das aplicações tradicionais. Na área médica, diversas atividades podem ser atendidas por aplicações ubíquas, como telemonitoramento, diagnósticos remotos, segunda opinião, juntas médicas, *Home Care*, sistemas de prontuário eletrônico, etc. Esse cenário caracteriza uma integração entre medicina e Ubicomp, e vem sendo denominado de medicina ubíqua [5].

Uma das motivações centrais para medicina ubíqua é a necessidade de ambientes computacionais que ofereçam suporte à mobilidade dos profissionais de saúde. A mobilidade dos médicos, especialmente, é algo inerente à própria profissão. Além desse caráter nômade do médico, a atividade médica é bastante fragmentada, estando sujeita a interrupções durante sua execução, uma vez que médicos passam pouco tempo em cada local ou atividade. Dessa forma, mecanismos que facilitem a continuidade das atividades durante seus constantes deslocamentos tendem a melhorar a produtividade dos mesmos [17].

As aplicações ubíquas na área de medicina são caracterizadas por apresentarem contextos de execução dinâmicos e de composição heterogênea, com elevado grau de mobilidade dos seus principais atores (médicos, pacientes, paramédicos, entre outros). Nesse sentido, 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. A construção de sistemas sensíveis ao contexto permanece uma tarefa complexa com custo elevado para desenvolvimento, manutenção e reaproveitamento de código. Dentre os motivos que levam a essa situação destaca-se a falta de uma apropriada infraestrutura de suporte às aplicações que atendam a estas demandas [7].

Na construção de aplicações ubíquas sensíveis ao contexto uma série de funcionalidades devem ser providas, abrangendo desde a aquisição de informações de contexto, a partir de um conjunto de fontes heterogêneas e distribuídas, até a representação dessas informações, armazenamento, processamento e raciocínio para seu uso em tomadas de decisão. Para reduzir essa complexidade, tem-se obser-

¹Trabalho integrado ao Projeto Pertmed (*Pervasive Telemedicine*), financiado pelo Finep/MCT

vado o emprego de infra-estruturas de suporte às aplicações, que passam a ser responsáveis pelo provimento destas funcionalidades [2].

Assim, na perspectiva de suprir estas funcionalidades, a arquitetura de software proposta neste artigo, denominada uMED (*ubiquitous MEDicine*), visa prover suporte às aplicações ubíquas sensíveis ao contexto da área de medicina através do gerenciamento das informações contextuais do ambiente, coletando dados de contexto de diferentes naturezas, analisando essas informações como variáveis independentes ou combinadas e disponibilizando-as para que as aplicações possam utilizá-las, em função de seu contexto de interesse, nos processos de reação às variações no meio. O modelo proposto tem como premissa central a integração de tecnologias semânticas nas tarefas de processamento das informações de contexto. O uMED está inserido nos esforços de pesquisa do projeto Pertmed e desenvolve-se no âmbito do grupo de pesquisa G3PD².

O artigo está organizado da seguinte forma: a seção 2 apresenta a concepção do uMED. A seção 3 mostra a modelagem do uMED, destacando o modelo proposto para representação do contexto e a arquitetura de software prevista. A seção 4 destaca um estudo de caso. A seção 5 sintetiza os trabalhos relacionados e os compara com o uMED. Por fim, a seção 6 apresenta as considerações finais.

2 Concepção da Proposta

A concepção do uMED considera a premissa do mesmo ser integrado ao *middleware* EXEHDA [9]. 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: (i) um EXEHDAbase, responsável por todos os serviços básicos da célula; (ii) EXEHDA nodos, são os equipamentos de processamento disponíveis no ambiente; e (iii) EXEHDA mob-nodos, são os nós do sistema passíveis de portabilidade.

As tecnologias de Web Semântica são empregadas pelo uMED nas funcionalidades de aquisição, processamento e disponibilização das informações contextuais, considerando as premissas de elevada heterogeneidade e escalabilidade da computação ubíqua.

A decisão pelo emprego de tecnologias de Web Semântica, particularmente ontologias, na concepção do uMED se deve ao fato destas: (i) propiciarem uma maior expressividade na definição do contexto; (ii) possibilitarem a realização de inferências a partir das informações de contexto; (iii) viabilizarem facilidades de reutilização e extensibilidade por novas aplicações ou novas situações

de contexto. Além disso, a literatura registra um crescimento na utilização de tecnologias de Web Semântica no desenvolvimento de aplicações ubíquas, com o objetivo de modelar e processar as informações referentes a essas aplicações [15, 18, 16, 8, 1].

As principais tecnologias de Web Semântica utilizadas pelo uMED são: (i) linguagem OWL-DL (*Web Ontology Language - Description Logics*) [10] para definir e instanciar ontologias, com máxima expressividade, mantendo a computabilidade e a decidibilidade; (ii) API Jena [4] para manipulação de ontologias e realização de inferências sobre as mesmas; e (iii) linguagem SPARQL (*SPARQL Protocol And RDF Query Language*) [13] para consulta em ontologias.

3 Modelagem do uMED

Esta seção apresenta o modelo proposto para representação de contexto do ambiente ubíquo, bem como a arquitetura de software concebida para o uMED.

3.1 Representação do Contexto

Para a modelagem das informações contextuais processadas pelo uMED são utilizadas ontologias [6]. Estas são empregadas sob duas perspectivas: (i) representação semântica dos dados contextuais e; (ii) estabelecimento de relações entre os mesmos, possibilitando a realização de inferências. Com isso, pretende-se contribuir para qualificação das informações contextuais adquiridas e notificadas pelo uMED.

A construção da ontologia utilizada pelo uMED emprega a metodologia proposta por [11]. O modelo ontológico inclui a ontologia denominada OntContext (Ontologia da Situação do Contexto), que representa os contextos coletados, os contextos notificados e as instâncias dos contextos de interesse das aplicações (vide Figura 1).

A arquitetura de software prevê gerentes autônomos e cooperantes para a realização de tarefas de manipulação e dedução sobre o contexto. Na perspectiva da arquitetura, as ontologias são usadas para representar o ambiente ubíquo, descrevendo as entidades envolvidas e suas respectivas propriedades. Neste sentido, elas definem, principalmente, os diferentes tipos de aplicações, serviços e dispositivos. Além disso, as ontologias são utilizadas na descrição de padrões para informações médicas, como por exemplo: temperatura corporal e pressão arterial.

A OntContext contempla a seguinte estrutura de classes e sub-classes:

- Classe 'Sensor': contém as instâncias dos sensores que participam do sensoriamento do ambiente ubíquo.

²Grupo de Pesquisa em Processamento Paralelo e Distribuído que abrange Instituições da Região Sul

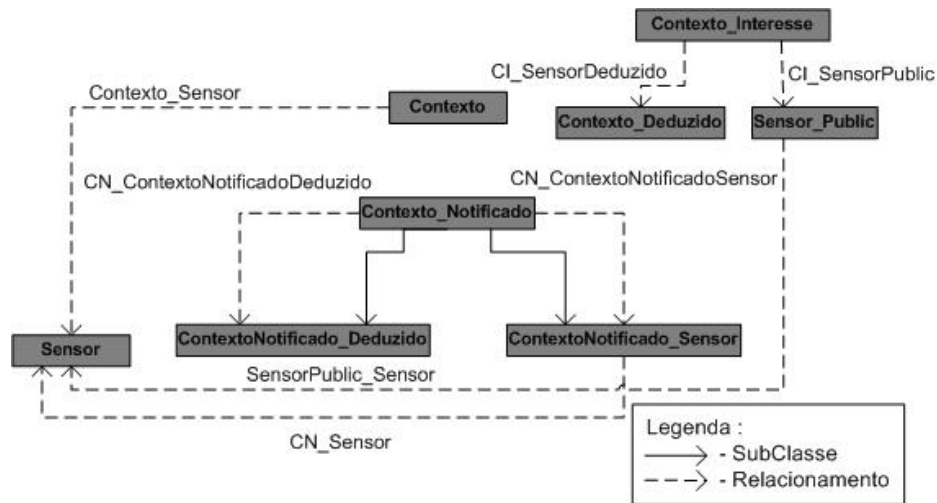


Figura 1. ontologia OntContext

- Classe 'Contexto': armazena os dados coletados pelos sensores no Repositório de Informações Contextuais (RIC), localizado na arquitetura de software do uMED.
- Classe 'Contexto_Notificado': persiste os contextos notificados, processados e deduzidos pelo Motor de Inferência da arquitetura de software.
- Sub-classes 'ContextoNotificado_Sensor' e 'ContextoNotificado_Deduzido': registram seus dados no Repositório de Contextos Notificados (RCN).
- Classe 'Contexto_Interesse' e sub-classes 'Sensor_Public' e 'Contexto_Deduzido': instanciam as aplicações, os parâmetros operacionais de publicação dos sensores e as regras de dedução definidas pelo desenvolvedor para processamento no Motor de Inferência.

3.2 Arquitetura de Software

O uMED tem como premissa tratar as informações adquiridas por sensores, notificar as partes interessadas conforme as regras de tratamento de contexto utilizadas pelas diferentes aplicações, e possibilitar que os sensores e/ou os atuadores possam ser ativados, desativados, configurados e consultados a qualquer hora de forma ubíqua, além de permitir alterações das regras de tratamento de contexto durante a execução das aplicações.

A arquitetura de software proposta para o uMED, apresentada na Figura 2, é dirigida por contextos de interesses, os quais são responsáveis por especificar os aspectos que devem ser considerados nos procedimentos de monitoração do ambiente ubíquo, de interpretação dos dados capturados e os respectivos procedimentos associados.

Nos contextos de interesse das diversas aplicações, que alimentam a arquitetura de software do uMED, estão definidas as informações contextuais a serem adquiridas pelos sensores e processadas (traduzidas e deduzidas) pela arquitetura, bem como os comandos empregados no controle dos atuadores. Essas especificações são relacionadas por aplicação, componente e adaptador, considerando os seguintes parâmetros e regras: (i) instanciação dos atuadores e sensores que participam das aplicações; (ii) parâmetros operacionais para controle da ativação, da desativação, bem como da configuração dos sensores e/ou atuadores; (iii) especificação do número máximo de registros pelos sensores a serem armazenados no repositório contextual. Ultrapassando este número, os valores mais antigos coletados pelos sensores são apagados; (iv) regras de deduções sobre os dados adquiridos pelos sensores; e (v) regras de deduções sobre os dados utilizados para controle dos atuadores (dados de ativação).

A arquitetura de software é composta por cinco módulos: Gerente de Atuação, Gerente de Aplicações, Gerente de Borda, Gerente de Comunicação e Servidor de Contexto. Os serviços da arquitetura são autônomos e cooperantes e possibilitam a coleta de dados, assim como o controle da ativação, desativação e configuração de atuadores. A arquitetura conta com diversos repositórios de dados, os quais são atualizados e/ou consultados durante a operação da mesma.

Gerente de Atuação

O Gerente de Atuação é responsável pelo controle da ativação, desativação e configuração dos atuadores empregados pelas aplicações gerenciadas pelo uMED. Esse gerente trabalha com duas abordagens distintas: Atuação

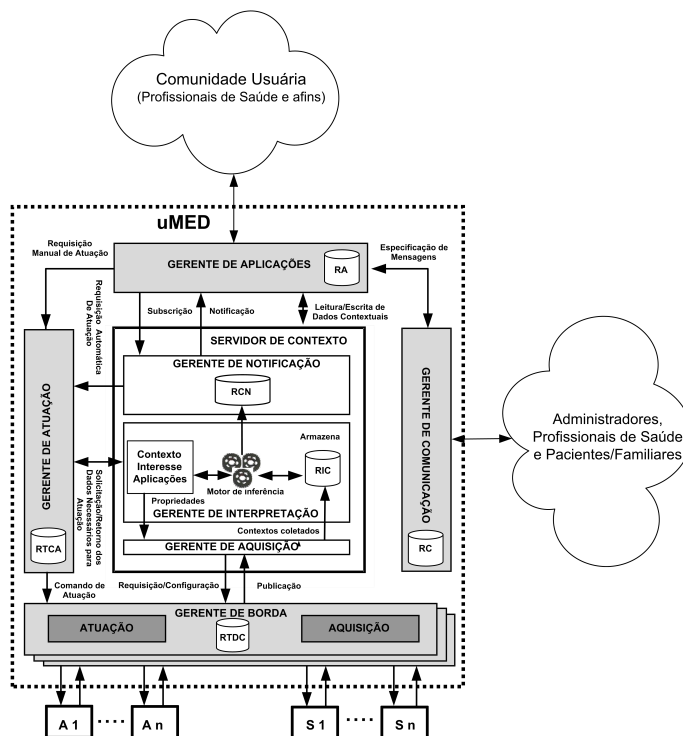


Figura 2. Arquitetura de Software

Manual e Atuação Automática.

Compreende-se como uma atuação manual aquela que é disparada pelo usuário da aplicação a qualquer instante, por exemplo, um profissional de saúde deseja ativar um dispositivo qualquer ou alterar algum parâmetro operacional do mesmo. Os dispositivos podem ser de gerência ambiental, tais como: aquecedores, umidificadores, alarmes, luzes de sinalização, bem como relacionados a demandas médicas: aparelho de oxigênio, bomba de infusão, ventilador pulmonar, dentre outros.

A atuação automática é disparada através de notificações produzidas pelo Servidor de Contexto, ou seja, o processamento de um determinado contexto de interesse pode implicar a ativação e/ou desativação e/ou configuração de um dispositivo. Os parâmetros operacionais necessários para atuação são definidos no próprio Servidor de Contexto.

Para tratamento da perda de conexão entre o Gerente de Atuação e o Gerente de Borda, existe um mecanismo de *bufferização* dos comandos emitidos (RTCA - Repositório Temporário de Comandos de Atuação), conforme mostra a Figura 2, o qual se vale de um parâmetro do tipo *time-to-live* para avaliar quais dos comandos ainda tem sentido serem tratados pelo Gerente de Borda.

Gerente de Aplicações

O Gerente de Aplicações é responsável por disponibilizar ao usuário final as aplicações que estão sendo gerenciadas pelo uMED. O Gerente de Aplicações trabalha com duas abordagens distintas: (i) componentes de software das aplicações assíncronas ao contexto; (ii) componentes de software das aplicações síncronas ao contexto. Ambas abordagens possuem características complementares na definição das aplicações ubíquas do uMED.

Os componentes de software das aplicações síncronas ao contexto são destinados ao monitoramento de sinais vitais de pacientes ou do ambiente relacionado aos mesmos, emitindo de forma automatizada, diferentes níveis de alertas, em função dos dados coletados, para os profissionais de saúde. Por sua vez, os componentes assíncronos ao contexto viabilizam as seguintes funcionalidades: (i) manipulação instantânea de sensores e atuadores; (ii) atuação no ambiente ubíquo através de alterações nas configurações operacionais dos atuadores; (iii) concessão ao usuário da aplicação da possibilidade de criar relatórios personalizados.

Gerente de Borda

O Gerente de Borda é responsável pelo primeiro processamento dos dados capturados pelos sensores, bem como o tratamento final das informações utilizadas para controle dos atuadores. Os dados adquiridos pelos sensores passam por várias etapas, nas quais são feitos tratamentos em níveis crescentes de abstração, culminando com a persistência dos mesmos no RTDC (Repositório Temporário de Dados Coletados). O dado persistido é publicado no Servidor de Contexto para ser analisado [14].

A arquitetura permite que os contextos possam ser capturados por diferentes Gerentes de Borda, cada um deles com um conjunto particular de sensores e/ou atuadores, com regimes operacionais individualizados. Esses Gerentes de Borda podem se valer da Internet e operar em diferentes instituições.

O controle dos atuadores coordenado pelo Gerente de Atuação, emprega os parâmetros operacionais especificados nos contextos de interesse das aplicações.

No Gerente de Borda, sensores e/ou atuadores podem ser implementados por meio de arquivos de configuração independentes. Desta forma, são facilitadas as operações de adição e/ou substituição de sensores e atuadores.

Gerente de Comunicação

O Gerente de Comunicação centraliza a relação do uMED com os diferentes segmentos de usuários através de interfaces pró-ativas. Seu objetivo é notificar informações

dos pacientes que estão sob monitoramento, através de mecanismos de uso rotineiro pelos envolvidos.

Com este serviço é possível notificar os profissionais de saúde, pacientes e seus familiares através de três tipos de mensagens: (i) Mensagens instantâneas através da rede celular, SMS - *Short Message Service (on-line)*; (ii) Mensagens através do *Google Talk (on-line)*; (iii) Mensagens por e-mail (*off-line*).

Com intuito de caracterizar para o profissional de saúde responsável o estado das mensagens enviadas, o Gerente de Comunicação contempla um repositório (RC - Repositório de Comunicações), no qual é registrado o sucesso ou não do envio dos três tipos de mensagem.

Servidor de Contexto

O servidor de contexto empregado pelo uMED é composto por três serviços: Gerente de Aquisição, Gerente de Interpretação e Gerente de Notificação (vide Figura 2).

Uma descrição resumida dos mesmos é feita a seguir:

- Gerente de Aquisição: tem a função central de prover a captura de informações de contexto, disponibilizando-as no RIC 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) especificar a flutuação mínima para que aconteça a publicação; (iii) definir a faixa na qual os valores dos sensores deverão ser publicados.
- Gerente de Interpretação: possui as seguintes funções: (i) consultar no RIC, os dados contextuais coletados 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.
- Gerente de Notificação: é responsável por disponibilizar os contextos processados pelo Gerente de Interpretação às aplicações.

O Servidor de Contexto é responsável pelo processamento das informações de contexto, empregando suporte semântico na execução destas funcionalidades [19].

No uMED, de forma específica, o Servidor de Contexto realiza as seguintes tarefas: (i) receber informações de contexto do Gerente de Borda; (ii) processar as regras de deduções; (iii) notificar ao Gerente de Aplicações conforme o contexto de interesse das aplicações; (iv) interagir com Gerente de Atuação.

No Servidor de Contexto existe a possibilidade das regras de tratamento contextual serem modificadas durante a execução. Dentre outros aspectos este perfil operacional garante uma operação ininterrupta, o que se mostra oportuno, considerando que um mesmo Servidor de Contexto pode atender simultaneamente a várias aplicações.

4 Estudo de Caso

O estudo de caso consiste em uma aplicação direcionada à área de medicina ubíqua denominada ACP (Acompanhamento Clínico de Pacientes), cujas funcionalidades são concebidas com o objetivo de explorar o suporte oferecido pelo uMED. Para ter liberdade na manipulação dos dados durante os testes é utilizada uma base de dados sensorizados criada pelo G3PD especificamente para este estudo de caso.

A premissa buscada é qualificar o acompanhamento clínico dos pacientes, que não estejam internados em UTI (Unidade de Tratamento Intensivo), com o intuito de aumentar o período de tempo que o paciente permaneça sob cuidado e evitar deslocamentos desnecessários pelos profissionais de saúde para saber o estado em que se encontra um paciente. Nesta perspectiva, os objetivos contemplados na aplicação ACP são:

- exibir dados de pacientes adquiridos dinamicamente por mecanismo de coleta de sinais vitais;
- emitir, de forma automatizada, diferentes níveis de alertas, em função dos dados coletados, para os profissionais de saúde;
- integrar o serviço de alertas da aplicação a rede aberta de comunicação *Google Talk*, SMS e e-mail;
- prover possibilidade de uso, tanto a partir de dispositivo móveis, como de mesa;
- permitir acesso ubíquo ao histórico dos dados coletados dos pacientes pelos profissionais de saúde;
- permitir acesso ubíquo ao histórico dos dados de atuação nos pacientes pelos profissionais de saúde;
- atuar remotamente em dispositivos médicos que estejam associados ao paciente.

Inicialmente, o desenvolvedor deve configurar a aplicação na ontologia “OntContext” para habilitar seus contextos de interesse, especificando: (i) os parâmetros operacionais e as regras de tradução dos sensores de monitoramento de pacientes (frequência cardíaca, pressão arterial e temperatura corporal); e (ii) as regras de dedução, para a funcionalidade de Envio Automático de Mensagens

(SMS, email, entre outros), sendo que os atores interessados devem se registrar para que possam receber notificações produzidas pelo processamento desta regra.

As regras de tradução e dedução empregadas utilizam uma sintaxe de construção adotada pelo subsistema de inferência da API Jena [4], baseado no raciocinador *Generic Rule Reasoner*, o qual permite a criação de regras definidas pelo desenvolvedor da aplicação.

A forma como está organizada a aplicação ACP permite sua integração aos demais sistemas de informações hospitalares. As informações cadastrais são de quatro naturezas: (i) dados cadastrais de pacientes; (ii) dados dos profissionais de saúde (médicos e enfermeiros); (iii) dados adquiridos dinamicamente pela rede de sensores de sinais vitais de pacientes; (iv) dados de ativação de atuadores.

Os profissionais de saúde, se desejarem, podem disparar o Módulo Aquisição do Gerente de Borda e capturar em determinado instante os dados correspondentes aos sinais vitais dos pacientes, bem como ativar o Módulo Atuação do Gerente de Borda e atuar remotamente em dispositivos associados ao tratamento do paciente. Este procedimento manual não interfere nas rotinas automáticas de captura de dados.

Os sinais capturados para este estudo de caso são frequência cardíaca, pressão arterial e temperatura corporal. Em função dos valores dos sinais coletados são produzidos diferentes níveis de alertas aos profissionais de saúde. Dependendo do nível de alerta é disponibilizada a opção de envio de mensagens aos profissionais de saúde ou, ainda, no caso de alerta máximo, o envio incondicional de mensagens.

Na aplicação ACP é possível monitorar de forma pró-ativa os pacientes, observando os últimos alertas disparados (vide Figura 3). Os níveis de alerta são classificados pela cor do alerta, da seguinte forma: (i) Alerta Verde: sinais vitais normais. Todos os sinais monitorados se encontram na faixa aceitável; (ii) Alerta Amarelo: sinal vital irregular. Um dos sinais monitorados se encontra em uma faixa irregular; (iii) Alerta Vermelho: sinais vitais críticos. Dois ou mais sinais monitorados se encontram em uma faixa irregular, este alerta é considerado o alerta máximo.

Os sensores associados ao paciente podem ser configurados através de uma interface disponibilizada aos profissionais de saúde. Através da mesma são selecionados o tipo de sensor e determinados seus parâmetros operacionais.

Com as informações armazenadas dos sensores e atuadores é possível a geração de relatórios de histórico clínico, proporcionando estudos mais aprofundados dos casos clínicos dos pacientes (vide Figura 4).

Na aplicação ACP é possível configurar o envio de notificações aos profissionais de saúde para cada nível de alerta. São explorados os três tipos de notificações existentes, sendo que no alerta vermelho é obrigatória a escolha



Figura 3. Monitoramento Pró-ativo de Pacientes

de pelo menos um tipo.

5 Trabalhos Relacionados

No levantamento do estado-da-arte foram considerados projetos que constituem uma parcela representativa do que vêm sendo desenvolvido nos últimos anos para atender as demandas da medicina ubíqua, e cuja arquitetura de software contempla o emprego de um *middleware*. Os projetos ABC [12], Awareness [20], UbiDoctor [5] e ClinicSpace [3] são sintetizados na tabela 1. A revisão sistematizada destes projetos, foi central na definição das principais características previstas na modelagem do uMED.

A concepção do uMED prevê uma arquitetura que contempla serviços cooperantes para atender as tarefas de monitoramento de dados contextuais, de atuação no meio e emissão de alertas aos profissionais de saúde. Esse modelo arquitetural não é encontrado nos projetos pesquisados, os quais, ao contrário do uMED, empregam serviços específicos no atendimento das diferentes funcionalidades.

Assim como o uMED, os demais projetos estudados contemplam uma arquitetura para suporte à sensibilidade ao contexto, modelada para ser expansível, tanto no que diz respeito a captura de dados do ambiente ubíquo, bem como quanto aos possíveis consumidores de contextos de interesse.

Nos projetos pesquisados, ao contrário do uMED, não

Tabela 1. Síntese dos Trabalhos Relacionados

Características	ABC	Awareness	UbiDoctor	ClinicSpace
Mobilidade Remota		X	X	X
Adaptação de Conteúdo	X		X	X
Sensibilidade ao Contexto	X	X	X	X
Processamento Semântico				X
Intervenção no Ambiente Ubíquo				X
Notificação Automática	X	X		X
Descoberta de Recursos	X	X		
Personalização de Regras				X

Acompanhamento Clínico de Pacientes

Monitoramento Configurações Relatórios

Relatório de Histórico Clínico

Código do Paciente: 367712
Nome do Paciente: Renato da Silva Oliveira
Leito: 04

Data Início: 08/12/2010
Data Fim: 09/12/2010

Buscar

Dados Coletados dos Sensores

Data / Hora	Sensor	Valor Coletado	Alerta
09/12/2010 22:13	Temperatura	40.2 °C	Vermelho
09/12/2010 22:13	Frequência Cardíaca	120 bpm	Vermelho
09/12/2010 22:13	Pressão Arterial	160 / 120 mmHg	Vermelho
09/12/2010 22:10	Temperatura	39.3 °C	Amarelo
09/12/2010 22:06	Temperatura	38.7 °C	Amarelo
09/12/2010 22:00	Temperatura	38.0 °C	Amarelo
09/12/2010 19:38	Temperatura	36.5 °C	Verde
09/12/2010 19:38	Frequência Cardíaca	70 bpm	Verde
09/12/2010 19:38	Pressão Arterial	120 / 80 mmHg	Verde
09/12/2010 15:22	Temperatura	36.0 °C	Verde

Dados dos Atuadores

Data / Hora	Atuador	Potência de Atuação	Tempo de Atuação
09/12/2010 07:05	Ap. Oxigênio	Normal	720 s
09/12/2010 14:40	Bomba Infusão	Baixa	180 s
09/12/2010 22:10	Ventilador Pulmonar	Baixa	320 s

Imprimir

© 2010 G3PD

Figura 4. Relatório de Histórico Clínico

existem mecanismos para criação de relatórios personalizados pelos usuários, possibilitando a geração de dados para estudo dos casos clínicos.

Ao contrário do uMED, os trabalhos relacionados não facultam a criação de estratégias que permitam a inserção ou remoção de sensores e/ou atuadores de forma que sejam abstraídas as especificidades dos mesmos.

O uMED disponibiliza um mecanismo de tolerância a falhas de comunicação nos gerentes de borda, o qual persiste os dados adquiridos pelos sensores, garantindo sua disponibilização ao servidor de contexto quando forem restabelecidas as comunicações. Somente o projeto Awareness possui um mecanismo similar de tolerância a falhas nos sensores.

O uMED disponibiliza uma interface para os profissio-

nais de saúde especificarem o recebimento de alertas através de mensagens, esta interface permite ao profissional de saúde determinar as suas preferências quanto aos tipos de mensagens. Apenas o projeto ClinicSpace viabiliza funcionalidade análoga ao envio de alertas através de mensagens por e-mail.

Todos os projetos relacionados e o uMED possuem funcionalidades que permitem ao usuário realizar leitura instantânea de sensores e especificar os parâmetros operacionais para publicação dos sensores pertencentes as aplicações.

O uMED permite a especificação de características funcionais que possibilitam ao usuário realizar alterações de regras em tempo de execução, manipular de forma instantânea os atuadores, promover alteração nos parâmetros de configuração dos atuadores remotamente e especificar os parâmetros operacionais para configuração dos atuadores pertencentes as aplicações. Estas características estão presentes somente no projeto ClinicSpace.

6 Considerações Finais

Com o intuito de colaborar para superar alguns dos desafios identificados para a área de medicina ubíqua, propomos uma arquitetura de software para gerenciamento de aplicações ubíquas sensíveis ao contexto direcionadas a esta área. Consideramos que o emprego do uMED em um sistema de apoio a vida pode possibilitar uma diminuição no nomadismo e fragmentação de atividades exercidas pelos profissionais de saúde.

A arquitetura de software proposta contempla o monitoramento dos sinais vitais do paciente e/ou condições do seu ambiente, tendo a possibilidade de gerar alertas conforme regras definidas pelos próprios profissionais de saúde, bem como produzir relatórios personalizáveis, tanto com informações gerenciais como destinadas a estudo de casos clínicos. As regras de monitoramento podem ser alteradas em tempo de execução. Entende-se por tempo de execução a flexibilidade da arquitetura de software em

reconfigurar dinamicamente as regras para manipulação e processamento das informações contextuais.

O uMED ainda possibilita a atuação no ambiente. Com base nos dados obtidos através do monitoramento, o profissional de saúde pode decidir por promover o acionamento, a configuração ou o desligamento de algum equipamento ou dispositivo. A versão atual do protótipo já conta com interface customizada para *smartphone (iPhone)* e computador de mesa.

Resumidamente, entendemos que a contribuição central do trabalho consiste na concepção de uma arquitetura de software direcionada a aplicações da medicina ubíqua, que trata de modo sinérgico quatro frentes: (i) a aquisição de dados monitorados; (ii) a atuação no ambiente ubíquo; (iii) a gerência dos diferentes níveis de alerta, conforme processamento das regras contextuais; (iv) a geração de dados para estudo dos casos clínicos.

Na continuidade da pesquisa do uMED destacam-se os seguintes aspectos: (i) ampliar o estudo de padrões de sensores e atuadores no desenvolvimento de aplicações direcionados a área médica; (ii) utilizar um sistema especialista no processamento de regras contextuais no Servidor de Contexto; (iii) reavaliar os protocolos e procedimentos empregados na troca de informações entre os Gerentes.

Referências

- [1] C. A. Costa. *Continuum: A Context-aware Service-based Software Infrastructure for Ubiquitous Computing*. Tese (doutorado em ciência da computação), UFRGS, Porto Alegre, RS, 2008.
- [2] C. A. Costa, L. C. d. Silva, J. L. V. Barbosa, A. C. Yamin, and C. F. R. Geyer. A primer of ubiquitous computing challenges and trends. In F. M. M. Neto and P. F. R. Neto, editors, *Designing Solutions-Based Ubiquitous and Pervasive Computing: New Issues and Trends*, volume 1, chapter 15, pages 282–303. IGI Global Publishing, Hershey, 2010.
- [3] F. L. da Silva. *ClinicSpace: Modelagem de uma Ferramenta-Piloto para Definição de Tarefas Clínicas em um Ambiente de Computação Pervasiva Baseado em Tarefas e Direcionado ao Usuário-Final*. Dissertação de mestrado em ciência da computação, PPGI/UFSM, Santa Maria, RS, 2010.
- [4] I. Dickinson. Jena ontology api, 2009. Disponível em: <<http://jena.sourceforge.net/ontology/>>. Acesso em abril de 2011.
- [5] J. Diniz. *UbiDoctor: Arquitetura de Serviços para Gerenciamento de Sessão e Adaptação de Conteúdo em Ambientes de Medicina Ubíqua*. Tese (doutorado em ciência da computação), Universidade Federal de Pernambuco, UFPE, Recife, PE, 2009.
- [6] D. Fensel, W. Wahlster, and H. Lieberman, editors. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, Cambridge, MA, USA, 2005.
- [7] K. Henriksen and J. Indulska. Developing context-aware pervasive computing applications: Models and approach. *Pervasive and Mobile Computing*, 2(2):37–64, 2006.
- [8] J. L. B. Lopes. *EXEHDA-ON: Uma Abordagem Baseada em Ontologias para Sensibilidade ao Contexto na Computação Pervasiva*. Dissertação (mestrado em ciência da computação), UCPEL, Pelotas, RS, 2008.
- [9] J. L. B. 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.
- [10] D. L. McGuinness and F. van Harmelen. Owl web ontology language overview, 2009. Disponível em: <<http://www.w3.org/TR/owl-features/>>. Acesso em abril de 2011.
- [11] N. F. Noy and D. L. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05, Knowledge Systems, AI Laboratory, Stanford University, 2001.
- [12] J. B. Pedersen, M. Mogensen, and J. E. Bardram. The abc adaptive fusion architecture. *Proceedings of the 4th international workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC 2006)*, 182, 2006.
- [13] E. Prud'hommeaux and A. Seaborne. Sparql - a query language for rdf, 2008. Disponível em: <<http://www.w3.org/TR/rdf-sparql-query/>>. Acesso em abril de 2011.
- [14] I. Santos, S. Rodrigues, L. Venecian, J. Lopes, and A. Yamin. Exehda-gb: Um mecanismo para captura e modificação do estado de contextos na computação ubíqua. In *Anais da 11ª Escola Regional de Alto Desempenho*, Porto Alegre, RS, 2011.
- [15] V. Suraci, S. Mignanti, and A. Aiuto. Context-aware semantic service. In *16th IST Mobile and Wireless Communications Summit*, pages 1–5, 2007.
- [16] M. Sutterer, O. Droegehorn, and K. David. Upos: User profile ontology with situation-dependent preferences support. In *First International Conference on Advances in Computer-Human Interaction - ACHI 2008*, pages 230–235, 2008.
- [17] M. Tentori and J. Favela. Activity-aware computing for healthcare. *IEEE Pervasive Computing*, 7(2):51–57, Abril 2008.
- [18] A. Toninelli, R. Montanari, L. Kagal, and O. Lassila. Proteus: A semantic context-aware adaptive policy model. *IEEE International Workshop on Policies for Distributed Systems and Networks*, 0:129–140, 2007.
- [19] L. R. Venecian, J. L. B. Lopes, I. Augustin, A. C. Yamin, and C. Geyer. Um mecanismo para sensibilidade ao contexto com suporte semântico na computação ubíqua. In *XXXVI Latin American Informatics Conference - CLEI*, Asuncion, Paraguay, 2010.
- [20] M. Wegdam. Awareness: A project on context aware mobile networks and services medical systems international. In *the Proceedings of the 14th Mobile and Wireless Communications Summit*, pages 19–23, Junho 2005.