

uMED: Uma Arquitetura de Software Direcionada à Medicina Ubíqua

Sérgio Rodrigues¹, Luthiano Venecian¹, João Ladislau¹, Adenauer Yamin^{1 2}

¹PPGINF-Centro Politécnico
Universidade Católica de Pelotas (UCPeL)
Rua Félix da Cunha, 412 – 96010-000 – Pelotas – RS – Brazil

²Centro de Desenvolvimento Tecnológico
Universidade Federal de Pelotas (UFPeL)
Caixa Postal 354 – 96001-970 – Pelotas – RS – Brazil

{sergior, venecian, joaolopes, adenauer}@ucpel.tche.br

Resumo. *A medicina ubíqua tem como premissa disponibilizar serviços de saúde a qualquer hora, sem restrições de localização potencializando a mobilidade dos profissionais de saúde. Estes profissionais necessitam de ferramentas de entrega e de acesso a informações tanto no local onde encontra-se o paciente, como não. O objetivo deste trabalho é propor uma infraestrutura de software que integre sensores e dispositivos computacionais tanto móveis quanto fixos, em um ambiente ubíquo, oferecendo serviço de sensibilidade ao contexto. Esta arquitetura denominada uMED, e suas respectivas aplicações alvo irão contemplar o monitoramento de sinais vitais de pacientes considerando um ambiente típico de trabalho de equipes médicas.*

1. Introdução

A computação ubíqua vem conquistando espaços com a disseminação de dispositivos portáteis, sobretudo após a ampliação das tecnologias de redes sem fio [Costa et al. 2008]. Dessa forma, o uso desta computação tem conduzido a aplicações de diversas áreas do mundo real, onde sua usabilidade nos moldes da computação tradicional não poderia ser tão ampla. Cenários na área médica, podem estender-se a diversas atividades, como telemonitoramento através do uso de sensores de medicina ubíqua, diagnósticos remotos, segunda opinião, juntas médicas, *Home Care* viabilizando a integração de dispositivos móveis como *Personal Digital Assistants* (PDAs), *Tablets*, *Smartphones* com os sistemas de prontuário eletrônico existentes nos hospitais, entre outros cenários, desde residências inteligentes a trabalhos colaborativos [Diniz 2009].

Como uma das motivações para medicina ubíqua, temos a necessidade que ambientes computacionais ofereçam suporte à mobilidade dos profissionais de saúde, tendo em vista que a mobilidade dos médicos é inerente à própria profissão. Além desse caráter nômade do médico, é importante considerar que a atividade médica é bastante fragmentada [Tentori and Favela 2008], ou seja, está 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 de atividades dos profissionais, mesmo em virtude de seus constantes deslocamentos, tendem a melhorar a produtividade dos mesmos [Diniz 2009].

Ao se construir e executar aplicações para medicina ubíqua, há uma série de funcionalidades que devem ser providas, envolvendo desde a aquisição de informações contextuais, a partir do conjunto de fontes heterogêneas e distribuídas, até a representação dessas informações, seu processamento, armazenamento, e a realização de inferências para seu uso em tomadas de decisão. Em vez dessas funcionalidades ficarem a cargo da aplicação, e deste modo incorporandas ao código do negócio, são utilizadas plataformas ou *middlewares* de provisão de contexto [Diniz 2009].

Na perspectiva de suprir estas funcionalidades, este trabalho tem como objetivo principal propor um *framework* denominado uMED direcionado para demandas da medicina ubíqua.

2. uMED: Concepção e Modelagem

O *framework* uMED está inserido nos esforços de pesquisa do projeto PERTMED (Sistema de TeleMedicina Móvel) [PERTMED 2007], no qual vem sendo desenvolvido por três universidades do sul do Brasil (Universidade Federal de Santa Maria, Universidade Católica e Federal de Pelotas) contando com a participação dos grupos de pesquisa (GMob e G3PD), e com a colaboração de equipes médicas ligadas a essas universidades. O uMED é uma arquitetura de *software* direcionada para aplicações de medicina ubíqua.

2.1. Arquitetura de Software do uMED

O *framework* uMED tem como premissa receber as informações coletadas pelos sensores, notificar as partes interessadas conforme regras de monitoramento pré-definidas e possibilitar que os sensores/atuadores possam ser controlados (ativados, desativados, regulados) e consultados a qualquer hora de forma ubíqua, além de permitir a alteração de regras de monitoramento em tempo de execução.

Uma visão geral da arquitetura de *software* do uMED é apresentado na Figura 1. A mesma é composta por quatro serviços: Gerente de Atuação, Gerente de Aplicações, Gerente de Borda e o Servidor de Contexto. Esses serviços são autônomos e cooperantes e possibilitam o monitoramento, assim como controle (ativação, desativação, regulação) dos atuadores e consultas em seus repositórios.

O *framework* uMED, emprega o ambiente ubíquo definido no projeto ISAM [ISAM 2009] e provido pelo *middleware* EXEHDA [Yamin 2004].

O uMED irá contemplar o uso de tecnologias de processamento semântico nas funcionalidades de: (i) aquisição, (ii) atuação e (iii) emprego das informações contextuais direcionadas as aplicações ubíquas.

Para que a arquitetura de *software* proposta, para o uMED, possa realizar tarefas de manipulação e dedução sobre o contexto, a mesma é alimentada por contextos de interesses, os quais são responsáveis por caracterizar os aspectos que devem ser considerados nos procedimentos de monitoração do ambiente ubíquo, de interpretação dos dados capturados e das respectivas notificações assim como as possíveis atuações no ambiente ubíquo. No Contexto de Interesse das Aplicações ficam definidas quais informações contextuais serão adquiridas pelos sensores, traduzidas e deduzidas, bem como quais comandos serão empregados no controle dos atuadores. Essas especificações são relacionadas por aplicação, componente e adaptador, considerando os seguintes parâmetros e regras:

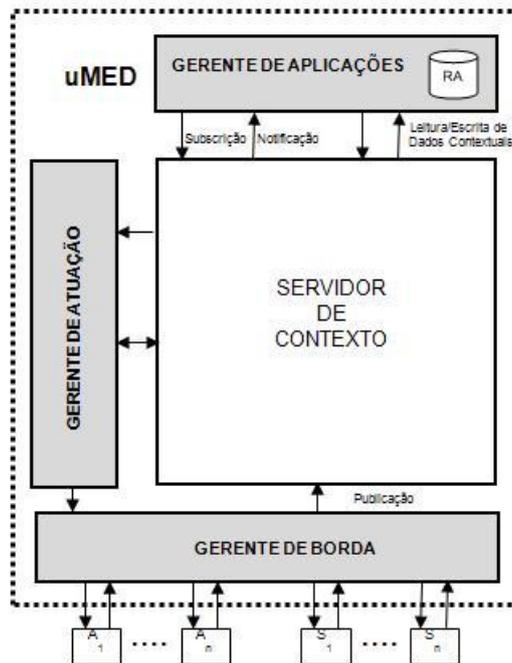


Figura 1. Arquitetura de Software Proposta para uMED

- instanciação dos sensores/atuadores que participam das aplicações;
- parâmetros operacionais para ativação/desativação e publicação dos sensores;
- parâmetros operacionais para controle (ativação, desativação, regulagem, etc.) dos atuadores;
- especificação do número máximo de registros pelos sensores a serem armazenados no repositório contextual. Ultrapassando este número, os valores mais antigos serão apagados;
- regras de deduções sobre os dados coletados pelos sensores;
- regras de deduções sobre os dados utilizados para controle dos atuadores (dados de ativação).

Na seção 2.1.1 é abordado os gerentes que compõem o modelo do uMED.

2.1.1. uMED: Principais Gerentes

O uMED é composto por três gerentes: Gerente de Atuação, Gerente de Aplicações e o Gerente de Borda. Além desses gerentes compõem a arquitetura de software do uMED o Servidor de Contexto. A seguir são abordados os três gerentes assim como o Servidor de Contexto utilizados na arquitetura de *software* prevista para uMED.

Gerente de Atuação

Responsável pelo controle (ativação, desativação e regulagem) dos sensores/atuadores, após ser notificado pelo Servidor de Contexto no qual informa os parâmetros operacionais a serem configurados, bem como quais sensores/atuadores deve ser interpelado(s) repassando essas informações ao Gerente de Borda para que ative os sensores/atuadores

de interesse da aplicação.

Além disto o Gerente de Atuação possui as seguintes atribuições:

- processar os contextos de interesse da aplicação, extraindo as informações para sua operação;
- repassar ao Gerente de Borda os parâmetros operacionais para ativação, desativação, regulagem dos sensores e atuadores;
- notificar ao Gerente de Borda quais sensores/atuadores serão necessários para atender as demandas das aplicações em uso;
- disparar no ambiente ubíquo uma verificação do status de sensores;
- disparar no ambiente ubíquo uma verificação do status de atuadores.

Gerente de Aplicações

O Gerente de Aplicações o qual trabalha com duas abordagens distintas: componentes de software das aplicações assíncronos ou síncronos ao contexto, os quais possuem características complementares quando da definição das aplicações ubíquas do uMED.

Para este trabalho compreende-se como um componente síncrono ao contexto da aplicação aquele que é programado para executar uma determinada ação caso ocorra algum evento de interesse do contexto da aplicação, já por componente assíncrono ao contexto da aplicação entende-se aquele cuja execução ocorre por intervenção do usuário. A seguir é detalhado a utilização da abordagem síncrona e assíncrona pelo Gerente de Aplicações.

Os componentes das aplicações que são síncronos ao contexto se inscrevem ao Servidor de Contexto e aguardam as notificações geradas por este servidor. Por exemplo, uma aplicação se inscreve informando se a temperatura corporal ou a pressão arterial de um determinado paciente alterar 10%, automaticamente deseja-se que essa variação seja notificada a um determinado profissional de saúde, essa notificação pode ser através de envio de mensagem de SMS (*Short Message Service*) ou pode ser um disparo de um alerta em uma determinada enfermaria, ou ambas as notificações.

Os componentes das aplicações que são assíncronos ao contexto, por sua vez, permitem que os profissionais de saúde (médicos, enfermeiros, entre outros) tenha a possibilidade de consultar informações sobre os sinais vitais de um determinado paciente a qualquer tempo, em qualquer lugar (acesso ubíquo), podendo inclusive serem gerados relatórios personalizado por este profissional de saúde. Outra funcionalidade prevista, é que o usuário a partir dos dados coletados, ou seja, baseado nas informações obtidas pela consulta, possa controlar (ativar, desativar, regular) os atuadores.

O Gerente de Aplicações contempla um repositório de dados denominado RA (Repositório de Aplicações), o qual contém o código fonte, os parâmetros operacionais das aplicações administradas por este Gerente. Este repositório funciona como um *container* WEB disponibilizando as aplicações para os usuários.

Gerente de Borda

Este Gerente é responsável pela coleta de dados capturados pelos sensores conforme parâmetros operacionais enviados pelo Gerente de Atuação, assim como, pelo controle dos atuadores. Os dados coletados pelos sensores passam por várias etapas, nas quais é feito o tratamento apropriado a esses dados capturados, de modo que seja possível então a persistência destes dados. Este mesmo dado persistido, é enviado ao Servidor de Contexto para ser analisado.

O controle dos atuadores depende dos parâmetros operacionais feitas pelo Gerente de Atuação no qual informa as preferências do Contexto de Interesse das Aplicações.

No Gerente de Borda, sensores/atuadores podem ser implantados por meio de arquivos de configuração. Desta forma, um sensor/atuador pode ser adicionado, removido ou substituído por outra implementação mais adequada ao dispositivo onde está sendo empregado no uMED.

Para a aquisição dos contextos através de sensores se faz necessário:

- especificar intervalos de tempo entre medições;
- registrar flutuação mínima para que aconteça a publicação;
- definir a faixa na qual os valores dos sensores deverão ser publicados.

O Gerente de Atuação envia parâmetros operacionais ao Gerente de Borda, através de um arquivo XML (*eXtensible Markup Language*) o que torna a arquitetura mais flexível. Conforme pode ser observado na Figura 2 o arquivo chamado de `SensorConfiguracao`, contém parâmetros para inicialização e publicação de contextos pelos sensores ao Gerente de Aquisição.

O arquivo de configuração deverá ser utilizado para configurar os sensores de acordo com seus parâmetros operacionais definidos pelo desenvolvedor no Contexto de Interesse das Aplicações. Esses parâmetros são especificados com informações para publicação, tais como: identificador do sensor, frequência de publicação, flutuação mínima, faixa mínima e máxima.

```
<?xml version="1.0" encoding="UTF-8"?>
<Sensors>
<Sensor identificador="codigo_sensor1" intMed="frequencia_publicacao"
flut="flutuacao_minima" valInf="faixa_minima" valSup="faixa_maxima"/>
<Sensor identificador="codigo_sensor2" intMed="frequencia_publicacao"
flut="flutuacao_minima" valInf="faixa_minima" valSup="faixa_maxima"/>
</Sensors>
```

Figura 2. Exemplo de Configuração do Arquivo `SensorConfiguracao`

De forma análoga, o atuador também pode ser configurado por um arquivo XML. O Gerente de Atuação envia parâmetros operacionais para o Gerente de Borda, em um arquivo XML contendo parâmetros operacionais definidos pelo desenvolvedor no Contexto de Interesse das Aplicações. Esses parâmetros são especificados com informações para atuação, tais como: identificador do atuador, ativa/desativa atuador, potência de atuação, intervalo de atuação.

Servidor de Contexto

Aplicações ubíquas executam em ambientes instrumentados com sensores, geralmente dotados de interfaces de redes sem fio, nos quais dispositivos, agentes de *software* e serviços são integrados e cooperam para atender os objetivos dos usuários finais. Essa categoria de aplicações caracteriza-se por constantes mudanças em seu contexto de execução, geradas pelos ambientes altamente dinâmicos em que tipicamente executam.

Deste modo, a sensibilidade ao contexto refere-se à capacidade de um sistema computacional perceber características de seu ambiente, e é um requisito fundamental para permitir a adaptação em resposta às mudanças ambientais. Aplicações sensíveis ao contexto conhecem o ambiente no qual estão sendo utilizadas e tomam decisões de acordo com mudanças no seu próprio ambiente. Ou seja, reagem a ações executadas por outras entidades, podendo essas ser pessoas, objetos ou até mesmo outros sistemas, que modifiquem o ambiente. Essas aplicações, de um modo geral, tomam ciência das alterações que venham a acontecer no ambiente. Tais modificações denominam-se alterações nas informações de contexto [Venecian 2010].

O mecanismo concebido utiliza o ambiente ubíquo promovido pelo *middleware* EXEHDA [Yamin 2004]. Este se propõem a integração de tecnologias de suporte semântico em mecanismo de sensibilidade ao contexto, desde a sua aquisição, processamento e distribuição das informações contextuais, direcionado a computação ubíqua. Neste será mantido o Contexto de Interesse das Aplicações. E denomina-se EXEHDA-SS (Suporte Semântico) [Venecian 2010].

O EXEHDA-SS [Venecian 2010] foi concebido para ser responsável pelo 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 será possível um refinamento qualificado dessas informações capturas e distribuídas nas células de execução do EXEHDA. O mesmo será parte importante na arquitetura de software do uMED.

3. Trabalhos Relacionados

Considerando aspectos de afinidade foram selecionado cinco trabalhos relacionados, (PERTMED [PERTMED 2007], ABC [Bardram 2009], Awareness [Wegdam 2005], UbiDoctor [Diniz 2009] e ClinicSpace [da Silva 2010]). Observa-se que os cinco consideram soluções de *middleware* e serviços para ambientes de medicina ubíqua. Alguns, utilizam como cenário a rotina diária de um hospital, inclusive utilizando sensores, em alguns casos, equipamentos ubíquos espalhados em alguns setores hospitalares, outros utilizam como cenário o monitoramento de pacientes à distância. Também estão associados a interações entre médicos e pacientes ou entre grupos de médicos, de modo a prover, em alguns casos, a participação do paciente no tratamento.

Na tabela 1 e verificado se determinadas características estão ou não presentes nos trabalhos relacionados:

- abrangência: os trabalhos relacionados contemplam mobilidade remota (no qual é atendida uma área extensa por exemplo, região metropolitana) ou mobilidade local (no qual atende uma área restrita por exemplo, um hospital);

- adaptação: preocupa-se com o desconforto no uso de alguns dispositivos portáteis permitindo a migração das sessões para dispositivos mais adequados. Para oferecer suporte a essa multiplataforma, existe o serviço de adaptação de conteúdo;
- sensibilidade: um ambiente ubíquo contém diferentes dispositivos, tais como (sensores, atuadores, eletroeletrônicos, computadores, etc.) que interagem. Afim de coletar informações sobre o ambiente onde está sendo utilizado, existe o serviço de sensibilidade ao contexto;
- intervenção: consiste em uma funcionalidade, que permite realizar ações sobre o meio físico. No entanto, estas ações podem ser realizadas de acordo com políticas e parâmetros como: tipo de ação (ex. ligar, desligar dispositivos), tempo de duração (ex. ligar por dez minutos), potência (ex. ligar na velocidade máxima), entre outras variações específicas de cada atuador;
- automática: consiste na coleta automatizada de dados sensorizados e seu respectivo processamento. Em função do processamento dos dados no Servidor de Contexto podem ser disparados alertas para os profissionais de saúde.

Tabela 1. Síntese de Trabalhos Relacionados

Projetos	Abrangência	Adaptação	Sensibilidade	Intervenção	Automática
PERTMED	Remota	Sim	Sim	Sim	Sim
ABC	Local	Sim	Sim	Sim	Sim
Awareness	Remota	Sim	Sim	Sim	Sim
UbiDoctor	Remota	Sim	Sim	Não	Não
ClinicSpace	Remota	Sim	Sim	Sim	Sim

Conforme pode ser observado na tabela 1 alguns projetos analisados possuem características em comum, contudo com a pesquisa destes trabalhos foi possível conhecer um pouco sobre a área de medicina ubíqua seus objetivos e desafios a serem vencidos. O estudo destes projetos foi fundamental para proposta do uMED.

4. Considerações

Uma das formas de prestar suporte ao nomadismo dos profissionais de saúde é permitir que ele continue executando sua atividade, sabendo que determinados pacientes estão sobre monitoramento e as informações decorrentes estão disponíveis de qualquer lugar bastando para isso ter acesso a um dispositivo computacional, como por exemplo, PDAs, *Notebooks*, e *Desktops* entre outros. Sendo assim, este profissional de saúde não necessariamente precisa se deslocar até determinado paciente para saber o estado de saúde que ele se encontra, otimizando assim suas rotinas de trabalho. Com o objetivo de colaborar para superar alguns desafios da área de medicina ubíqua, este trabalho está propondo uma arquitetura sensível ao contexto que gerencia sinais vitais de pacientes, com possibilidade de alteração de regras de monitoramento 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 uso de tal arquitetura em um sistema de apoio a vida irá possibilitar uma diminuição no nomadismo e fragmentação de atividades exercidas pelos profissionais de saúde e com isso a possibilidade de um ganho de produtividade, já que haverá menos investimento de tempo em deslocamentos.

A proposta do uMED contemplará além do monitoramento de paciente a possibilidade de intervenção no meio, por exemplo, gerar alertas conforme regras definidas pelos próprios profissionais de saúde. Além de permitir a criação de relatórios personalizáveis sobre os dados monitorados de um determinado paciente.

Referências

- Bardram, J. (2009). Activity-based computing for medical work in hospitals. *ACM Transactions on Computer-Human Interaction*, 16(2).
- Costa, C. A., Yamin, A. C., and Geyer, C. F. R. (2008). Toward a general software infrastructure for ubiquitous computing. *IEEE Pervasive Computing*, 7(1):64–73.
- da Silva, F. L. (2010). 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. Tese de mestrado em ciência da computação, PPGI/UFSM, Santa Maria, RS.
- Diniz, J. (2009). *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.
- ISAM (2009). Infraestrutura de suporte às aplicações móveis. Disponível em: <<http://www.inf.ufrgs.br/isam/index.html>>. Acesso em março de 2010.
- PERTMED (2007). Pertmed - sistema de telemedicina móvel. Disponível em: <<http://pertmed.wkit.com.br/pertmed/doku.php>>. Acesso em novembro de 2009.
- Tentori, M. and Favela, J. (2008). Activity-aware computing for healthcare. *IEEE Pervasive Computing*, 7(2):51–57.
- Venecian, L. R. (2010). EXEHDA-CS: um mecanismo para sensibilidade ao contexto com suporte semântico. Tese de mestrado em ciência da computação, PPGINF/CPOLI/UCPEL, Pelotas, RS.
- Wegdam, M. (2005). 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.
- Yamin, A. (2004). *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.