

Uma Arquitetura Distribuída para Internet das Coisas na Medicina Ubíqua

Rodrigo Souza¹, João Lopes¹, Alexandre Souza²,
Patrícia Davet², Adenauer Yamin², Cláudio Geyer¹

¹ Universidade Federal do Rio Grande do Sul – Porto Alegre – RS

² Universidade Federal de Pelotas – Pelotas – RS

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

{arrdsouza, ptdavet, marilza, adenauer}@inf.ufpel.edu.br

Abstract. *Recent advances in the Internet of Things (IoT) area which have provided an increasing availability of networked sensors and actuators, has given a new perspective to research in the context-awareness in Ubicomp. In this sense, the main contribution of this paper is proposition of CoIoT, an architecture for IoT designed with the aim of providing, through rules, the proactive management of the EXEHDA Middleware interactions with the physical environment. To evaluate the functionalities of the proposed architecture we present a case study in the medical area. The results were promising.*

Resumo. *Os recentes avanços na área da Internet das Coisas (IoT) que têm proporcionado uma crescente disponibilidade de sensores e atuadores conectados em rede, vem dando uma nova perspectiva às pesquisas em consciência de contexto na Ubicomp. Neste sentido, a principal contribuição deste artigo consiste na proposição do CoIoT, uma arquitetura para a IoT concebida com o objetivo de proporcionar, através de regras, o gerenciamento proativo das interações do Middleware EXEHDA com o ambiente físico. Para avaliar as funcionalidades da arquitetura proposta, apresentamos um estudo de caso na área da medicina cujos resultados se mostraram promissores.*

1. Introdução

Na Ubicomp, os sistemas computacionais devem ser capazes de reagir às mudanças do estado das diferentes variáveis contextuais de seu interesse, as quais devem ser coletadas em ambientes largamente distribuídos [Knappmeyer et al. 2013]. Nesse sentido, os recentes avanços científicos e tecnológicos na área da Internet das Coisas (*Internet of Things* – IoT) têm proporcionado o uso de sensores em larga escala os quais constituem fontes geradoras de informações contextuais para as aplicações Ubíquas conscientes do contexto [Perera et al. 2014].

Grande parte dos desafios de pesquisa relacionados ao uso da Internet das Coisas na obtenção de informações contextuais, estão associados com a diferença entre requisitos de alto nível das aplicações ubíquas e as operações básicas dos dispositivos da IoT [Perera et al. 2014]. A contribuição central deste artigo visa suprir essa lacuna, através da concepção do CoIoT (*Context* + IoT), uma arquitetura integrada ao Middleware EXEHDA que tem por objetivo o tratamento de sensores e

atuadores. O EXEHDA (*Execution Environment for Highly Distributed Applications*) [Lopes et al. 2012] provê uma arquitetura de software baseada em serviços que visa criar e gerenciar um ambiente ubíquo, bem como promover a execução, sob este ambiente, das aplicações da computação ubíqua.

O CoIoT foi concebido para ser autônomo, gerenciado através de regras e, capaz de agir de forma proativa, tanto na captura de informações contextuais do ambiente físico, como na atuação remota sobre o mesmo.

O artigo está estruturado da seguinte forma. A seção 2 apresenta o modelo proposto para o CoIoT, detalhando sua arquitetura e funcionalidades. Na Seção 3, apresentamos um estudo de caso concebido na área da Medicina. Os trabalhos relacionados são apresentados na seção 4 e, na seção 5, são feitas as considerações finais.

2. CoIoT: Concepção e Modelagem

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 Servidor de Borda se destina a gerenciar as interações com o meio físico, enquanto o Servidor de Contexto atua no armazenamento e processamento das informações contextuais [Lopes et al. 2014].

O CoIoT tem sua arquitetura apresentada na Figura 1 com destaque para o Servidor de Borda, a qual foi concebida com o objetivo de gerenciar diferentes dispositivos da IoT, como nodos sensores heterogêneos (sensores programáveis), sensores não programáveis e atuadores. A arquitetura proposta tem por premissa atuar de forma autônoma uma vez que os dados contextuais continuam a ser coletados e processados mesmo nos períodos nos quais as aplicações interessadas no seu uso estejam inoperantes.

No CoIoT, com o intuito de atender o perfil inerentemente distribuído das aplicações ubíquas, o Subsistema de Integração foi concebido para promover a interoperação entre o Servidor de Borda e os demais serviços do Middleware em execução em outros equipamentos. 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 foi concebido o componente *Persistência Local* cuja função é realizar o armazenamento temporário dessas informações até que as mesmas sejam publicadas. O componente *Receptor de Comandos*, por sua vez, é responsável pelo recebimento dos comandos e das regras provenientes do Servidor de Contexto bem como pelo encaminhamento dos mesmos aos respectivos componentes da arquitetura. Todas as configurações necessárias para o funcionamento do Servidor de Borda são operacionalizadas através do componente *Configurador*. Através do mesmo é disponibilizada uma interface Web pela qual é possível gerenciar a remoção e inclusão de sensores e atuadores, configurar *drivers* de dispositivos, gerenciar as regras de controle, entre outros.

O *Motor de Regras* é o componente responsável pelo processamento das regras, as quais são armazenadas na Base de Regras. A arquitetura do Servidor de Borda não se restringe a um conjunto fixo de regras. Conforme as demandas das aplicações ubíquas o conjunto de regras pode ser expandido através da inclusão de novos elementos na Base de Regras e na Base de Perfis. Para servir de suporte ao processamento das regras, o com-

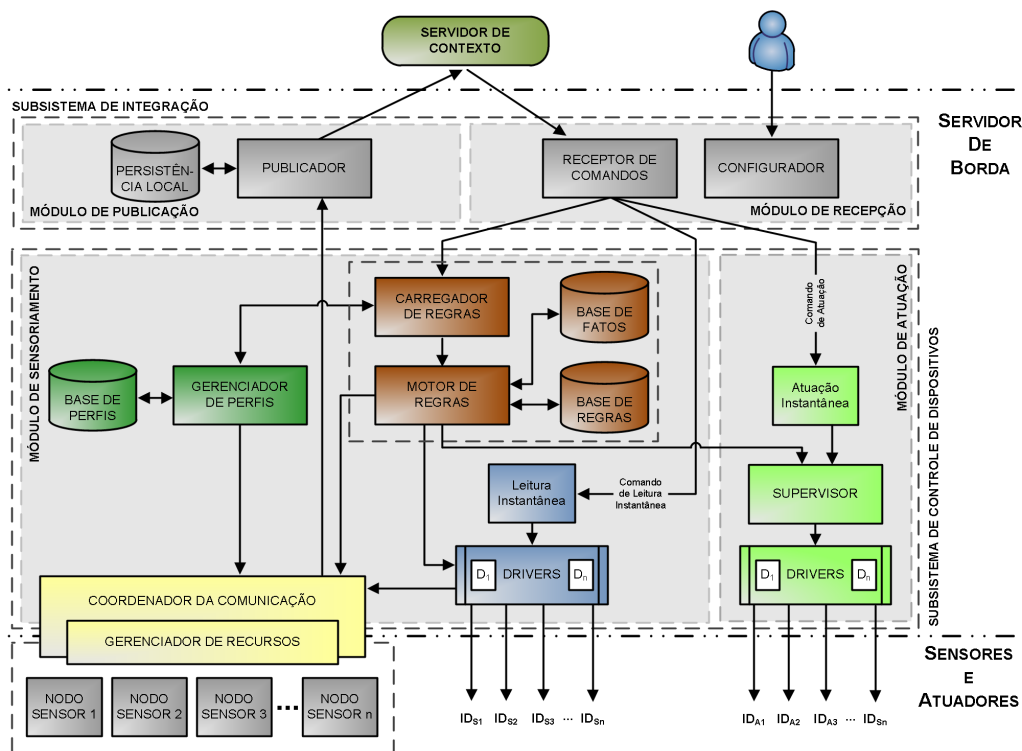


Figura 1. Arquitetura do CoIoT

ponente 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 CoIoT foram utilizados mecanismos de reprogramação dinâmica. Esta abordagem tem por objetivo adaptar o comportamento dos processos em execução nos nodos sensores conforme as demandas das aplicações. Quando uma regra é submetida ao Servidor de Borda, o componente *Carregador de Regras* informa o *Gerenciador 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 esses 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.

Os *Drivers*, enquanto 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*, por sua vez, 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.

Os componentes arquiteturais Coordenador da Comunicação e Gerenciador de Recursos foram concebidos com o objetivo de gerenciar os aspectos associados à comunicação entre os nodos sensores e o Servidor de Borda. No componente *Coorde-*

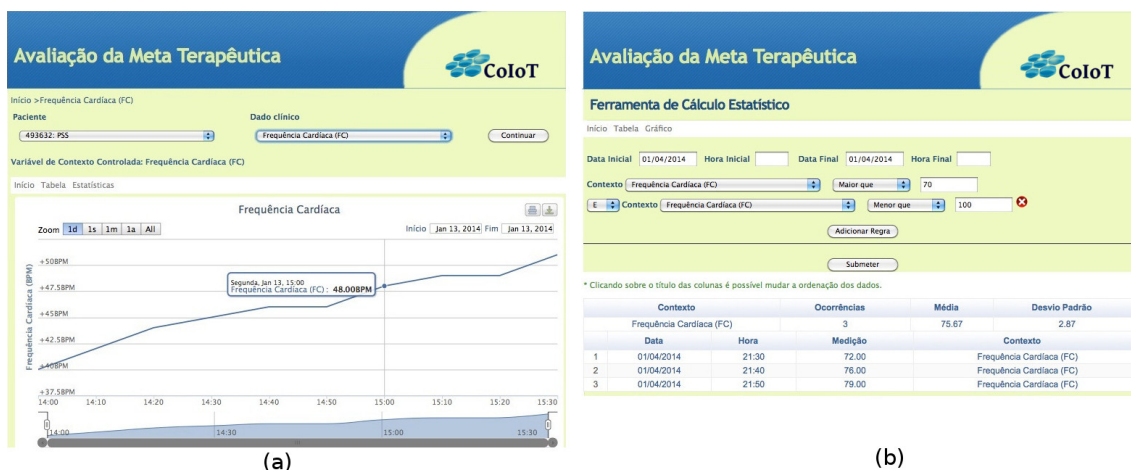


Figura 2. Aplicação Web: (a) Interface Gráfica; (b) Informações Estatísticas

nador da Comunicação foi adotado o modelo de coordenação baseado em Espaço de Tuplas a fim de promover a abstração dos aspectos de baixo nível associados à troca de informações entre os dispositivos. O *Gerenciador de Recursos*, por sua vez, atua na administração da entrada e saída de dispositivos da rede, ocorrências típica na IoT, mantendo assim a consistência da infraestrutura como um todo [Perera et al. 2014].

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.

3. Estudo de Caso

O estudo de caso foi desenvolvido com a expectativa de atender demandas da área Médica o qual possibilitou avaliar os seguintes aspectos da arquitetura do CoIoT: (i) sensoria-mento e a coleta de informações contextuais, (ii) avaliação e notificação dos dados de contexto aos demais serviços do Middleware e (iii) publicação dos dados sensoriados no Servidor de Contexto.

A bomba de infusão e o monitor de sinais vitais utilizados neste estudo de caso possuem comunicação de dados via *WiFi*. Com isso, é possível remotamente parametrizar os dados de uma nova infusão, fazer a leitura dos sinais vitais coletados, verificar o estado dos sensores/alarmes e o andamento da infusão.

O estudo de caso realizado consiste no acompanhamento da evolução dos sinais vitais do paciente, permitindo que o médico possa observar se está sendo atingido o efeito desejado com a administração dos medicamentos. Esse efeito procurado pelos médicos nos parâmetros vitais é, neste estudo de caso, denominado de *Meta Terapêutica*. Com o objetivo de verificar o comportamento da arquitetura do CoIoT através do acompanha-mento e avaliação das Metas Terapêuticas foi desenvolvida uma aplicação direcionada para uso com navegadores Internet.

Na tela inicial da aplicação é possível fazer a seleção de pacientes e os dados clínicos de interesse do profissional de saúde. Na Figura 2a é apresentado um gráfico típico da evolução dos sinais vitais. Através do mesmo é possível acompanhar a variação dos sinais vitais em função da medicação administrada ao paciente. Posicionando o *mouse* no gráfico é detalhado o valor e o momento (data e horário) de coleta do mesmo. Também é possível visualizar mais de um sinal vital simultaneamente em um mesmo gráfico, sendo viabilizada assim a comparação entre os mesmos. Na opção “Estatística” (vide Figura 2b) é apresentada uma funcionalidade da aplicação *Web* que viabiliza o cruzamento de dados contextuais envolvendo múltiplos sinais vitais a partir de diferentes regras. Esse recurso permite a adição, remoção e edição de regras e parâmetros.

Como estratégia de geração de alerta quando algum dos sinais vitais monitorado indicar que o paciente não está respondendo adequadamente ao tratamento foi utilizado o envio de e-mail e/ou SMS de forma automática aos profissionais de saúde.

4. Trabalhos Relacionados

O estudo da literatura da área proporcionou identificar alguns trabalhos relacionados, dentre os quais foram selecionados os seguintes: CARE [Agostini et al. 2009], CoCA [Ejigu et al. 2008], HiCon [Cho et al. 2008], Solar [Chen et al. 2008], WComp [Tigli et al. 2009]. 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 a atuação distribuída sobre o meio.

A arquitetura proposta para o CoIoT possibilita otimizar o gerenciamento tanto da aquisição dos dados de contexto a partir de vários tipos de sensores como na atuação distribuída 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 para o tratamento do contexto que adotam uma estratégia de separação entre a obtenção e o uso dos dados contextuais. Essa estratégia também é adotada no CoIoT, porém as tarefas de coleta e atuação operam de forma autônoma, uma vez que atuam de maneira independente das aplicações interessadas. Isto proporciona outro diferencial em relação aos projetos relacionados.

O suporte ao tratamento de regras é encontrado na maioria dos trabalhos identificados na literatura, porém a distribuição deste tratamento entre os Servidores de Contexto e de Borda é um diferencial em relação aos demais projetos, uma vez que esta funcionalidade usualmente fica restrita a um único equipamento.

5. Considerações Finais

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

A avaliação realizada na área Médica através de uma aplicação concebida para o acompanhamento dos sinais vitais e o respectivo envio de alertas aos profissionais de saúde, demonstra que o suporte arquitetural do CoIoT constitui uma ferramenta oportuna para avaliar a resposta do paciente ao tratamento. Este pronto aviso à equipe médica de que a Meta Terapêutica não foi atingida, também constitui uma das contribuições do CoIoT, uma vez que demonstra potencial no que tange à redução do tempo de internação.

Como trabalhos futuros, os seguintes aspectos deverão ser considerados: (i) realizar a validação clínica no ambiente hospitalar; (ii) explorar o emprego de regras de processamento contextual que utilizem outros mecanismos de inferência de mais alto nível; e (iii) dar continuidade aos procedimentos de integração do CoIoT com os diferentes serviços e funcionalidades do Middleware EXEHDA.

Referências

- Agostini, A., Bettini, C., and Riboni, D. (2009). Hybrid reasoning in the CARE middleware for context awareness. *International Journal of Web Engineering and Technology*, 5(1):3–23.
- Chen, G., Li, M., and Kotz, D. (2008). Data-centric middleware for context-aware pervasive computing. *Pervasive and Mobile Computing*, 4(2):216–253.
- Cho, K., Hwang, I., Kang, S., Kim, B., Lee, J., Lee, S., Park, S., Song, J., and Rhee, Y. (2008). HiCon: a hierarchical context monitoring and composition framework for next-generation context-aware services. *IEEE Network*, 22(4):34–42.
- Ejigu, D., Scuturici, M., and Brunie, L. (2008). Hybrid Approach to Collaborative Context-Aware Service Platform for Pervasive Computing. *Journal of Computers*, 3(1):40–50.
- Knappmeyer, M., Kiani, S. L., Reetz, E. S., Baker, N., and Tonjes, R. (2013). Survey of Context Provisioning Middleware. *IEEE Communications Surveys & Tutorials*, 15(3):1492–1519.
- Lopes, J. a. L., de Souza, R. S., Pernas, A. M., Yamim, A., and Geyer, C. (2014). A Distributed Architecture for Supporting Context-Aware Applications in UbiComp. In *IEEE International Conference on Advanced Information Networking and Applications (AINA)*, Victória, Canada.
- Lopes, J. a. L., Souza, R. S., Geyer, C. R., Costa, C. A., Barbosa, J. V., Gusmão, M. Z., and Yamin, A. C. (2012). A model for context awareness in UbiComp. In *Proceedings of the 18th Brazilian symposium on Multimedia and the web - WebMedia '12*, page 161, New York, New York, USA. ACM Press.
- Perera, C., Zaslavsky, A., Christen, P., and Georgakopoulos, D. (2014). Context Aware Computing for The Internet of Things: A Survey. *IEEE Communications Surveys & Tutorials*, 16(1):414–454.
- Tigli, J.-Y., Lavirotte, S., Rey, G., Hourdin, V., Cheung-Foo-Wo, D., Callegari, E., and Riveill, M. (2009). WComp middleware for ubiquitous computing: Aspects and composite event-based Web services. *annals of telecommunications - annales des télécommunications*, 64(3-4):197–214.