

Uma Proposta de Adaptação Dinâmica para Sistemas Ubíquos Baseados em Espaço de Tuplas Distribuído

Mestrando: Benedito José de Almeida Neto*

Orientadora: Rossana M. C. Andrade[‡]

Coorientador: Marcio E. F. Maia

Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat)
Mestrado e Doutorado em Ciência da Computação (MDCC)
Universidade Federal do Ceará (UFC)
Fortaleza – CE – Brasil
{beneditoneto, marcio, rossana}@great.ufc.br

Nível: Mestrado

Ano de Ingresso: 2011

Previsão de Conclusão: Março de 2013

Resumo: *A evolução das tecnologias móveis e da forma como a informação é acessada favorece o surgimento de sistemas capazes de antever as necessidades do usuário e se adaptar às variações de seu contexto de forma imperceptível. Tais sistemas, denominados sistemas ubíquos, apresentam o problema de adaptação dinâmica em um cenário altamente distribuído, heterogêneo e volátil, no qual componentes devem cooperar entre si, de forma descentralizada, para realizar uma determinada atividade. Baseado no levantamento bibliográfico sobre computação ubíqua verificou-se a necessidade da adoção de estratégias de coordenação descentralizada, permitindo que os diversos dispositivos que compõem um sistema ubíquo possam interagir e cooperar de forma eficiente diante de uma estrutura de rede dinâmica e com constantes desconexões. Neste sentido, este trabalho propõe uma estratégia de coordenação com foco em redes ad hoc, baseada em espaço de tuplas distribuídos, que permite a reconfiguração dinâmica de seus componentes.*

Palavras-chave:

Computação pervasiva, computação ubíqua, coordenação, espaço de tuplas distribuído, sensibilidade ao contexto, adaptação dinâmica.

*Bolsista de mestrado (MDCC/UFC) financiado pela CAPES através do programa PROPAG da UFC (Programa REUNI de Orientação e Operacionalização da Pós-Graduação Articulada à Graduação).

[‡]Bolsista de Produtividade Desenvolvimento Técnico e Extensão Inovadora do CNPq – Nível 2.

1. CONTEXTO TEÓRICO

A computação móvel tem se expandido consideravelmente devido a fatores como a popularização dos dispositivos portáteis e a evolução das redes móveis. A forma como as informações são acessadas evolui continuamente, atualmente os dispositivos computacionais estão imersos no ambiente, e oferecem diferentes modalidades de interação (e.g., visual, gestual, auditiva). Assim, surge uma demanda por sistemas rápidos, confiáveis e acessíveis em qualquer lugar e a qualquer momento (ubíquo) [1].

Diante da complexidade do desenvolvimento de aplicações ubíquas, a literatura [3][8][9] propõe o uso de arquiteturas, middlewares e sistemas de suporte que facilitem a ubiquidade, como em [15], que apresenta uma arquitetura para o suporte de aplicações ubíquas em redes domésticas centradas em TV digital.

Um dos desafios desse tipo de sistema é garantir a cooperação entre seus componentes e a adaptação suave a mudanças de contexto. Cooperação necessária, por exemplo, em uma aplicação que realize a transferência automática de um vídeo de um celular para uma TV baseada no contexto do usuário.

1.1 Computação Ubíqua

Historicamente, a computação tem evoluído a forma como os usuários interagem com os computadores. Partindo dos mainframes, passando pelos computadores pessoais e portáteis, até o advento da computação ubíqua, no qual um único usuário possui diversos dispositivos (*smartphones, tablets, laptops*) voltados à suas necessidades. O principal objetivo dos sistemas ubíquos é fornecer acesso à informação a qualquer momento e onde quer que o usuário esteja [3].

Segundo Mark Weiser [2], a computação ubíqua tem o objetivo de melhorar o uso do computador, fazendo muitos computadores disponíveis em todo o lugar, mas tornando-os efetivamente invisíveis para o usuário. Essa definição é considerada visionária e norteou muitas outras pesquisas das quais resultaram novas definições. Diante do atual cenário tecnológico, Lima [4] define computação ubíqua como: “O uso de um conjunto de computadores dos mais variados tamanhos, formatos e funções, que de forma coordenada e autônoma, auxiliam as pessoas na realização das diversas tarefas cotidianas. Esse auxílio é realizado de tal forma que a infraestrutura computacional responsável fica escondida no ambiente”. Assim, a autonomia e a coordenação entre os componentes do sistema tornam-se pontos fundamentais para a concretização da computação ubíqua.

Tais sistemas possuem características próprias que se relacionam à sensibilidade ao contexto, adaptabilidade e mobilidade. Em [4], é apresentada uma compilação dessas características, gerada a partir de uma ampla revisão literária, dentre as quais podemos citar a Captura de experiências e intenções; Comportamento adaptável; Descentralização; Descoberta automática de serviços; Heterogeneidade de dispositivos e serviços; Interoperabilidade espontânea; Mínima intervenção do usuário; Onipresença dos serviços; e Tolerância a falhas.

Diante dos desafios encontrados no desenvolvimento de sistemas ubíquos, esta pesquisa está direcionada ao comportamento adaptativo a partir da coordenação dos componentes que integram esses sistemas. O sistema deve estar apto a se adaptar diante da perda de conexão com algum dispositivo móvel, assim como ao surgimento repentino de vários dispositivos ao mesmo tempo. O

principal objetivo da coordenação é prover a autonomia necessária para a adaptação dinâmica tanto diante de mudanças no contexto quanto diante de falhas no sistema [5].

1.2 Middleware Ubíquo

Middlewares ubíquos devem atender aos requisitos da computação ubíqua, fornecendo abstrações que facilitem o desenvolvimento de aplicações. Na literatura, são encontrados trabalhos como [6], [7], [8] e [16] que elencam tais requisitos. Eles devem prover, a qualquer momento, acesso a informações de contexto heterogêneas, distribuídas e imprevistas em uma escala global e para diferentes cenários, permitindo a descoberta de novos tipos de contexto e informações restritas a um ambiente, garantindo a interoperabilidade semântica de contexto [8].

Segundo Rocha [6], esses middlewares devem suportar importantes requisitos de ubiquidade como: heterogeneidade, descoberta de serviço, coordenação e interoperabilidade. Em [4], são definidos os requisitos básicos comuns a muitos sistemas ubíquos e que devem ser gerenciados pelo middleware, sendo eles: *Coordenação, Descoberta/Descrição de Serviços, Interoperabilidade, Sensibilidade ao Contexto e Adaptabilidade, Invisibilidade e Autonomia*. O requisito de sensibilidade ao contexto está diretamente relacionado ao requisito de *Adaptabilidade*, uma vez que o sistema deve se adaptar a variações de contexto.

1.3 SysSU

SysSU¹ (*System Support for Ubiquity*) [4], é uma infraestrutura de suporte a ubiquidade que tem o objetivo de atender aos principais requisitos do desenvolvimento de sistemas ubíquos. Atualmente, ele suporta satisfatoriamente os requisitos de coordenação, descrição e descoberta de serviços, interoperabilidade e sensibilidade ao contexto. É possível adaptar sua arquitetura para que seja executado totalmente em uma plataforma embarcada e ainda estendê-lo para funcionamento em uma rede *ad hoc*. Ele implementa um modelo de coordenação formado pela composição dos modelos Linda [10] e *Publish/Subscribe* [11] e ainda especifica uma sintaxe para troca de mensagens, independente de linguagem de programação ou plataforma de desenvolvimento. O SysSU é baseado em espaço de tuplas centralizado, como pode ser visto na Figura 1. Dessa forma não é possível que aplicações executem sem conexão com um servidor central.

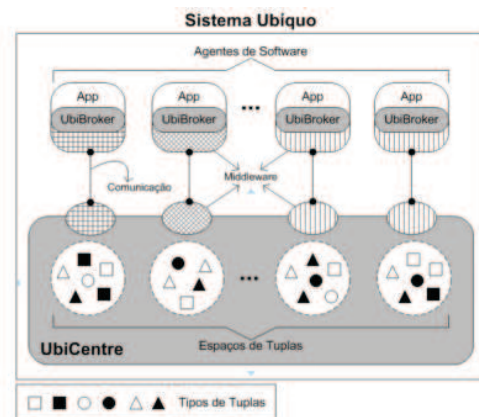


Figura 1. Arquitetura do SysSU [4].

¹<http://code.google.com/p/sysSU/>

As entidades principais do SysSU são *App*, *UbiBroker*, e *UbiCentre* (Figura 1). A *App* representa as aplicações ou serviços que compõem o sistema. O *UbiCentre* é a entidade que representa o espaço de tuplas centralizado, ele permite que as tuplas sejam acessadas concorrentemente pelas *Apps* através de um *UbiBroker*.

2. IDENTIFICAÇÃO DO PROBLEMA

Os sistemas ubíquos devem estar preparados para operar em, pelo menos, dois tipos de ambientes: (i) ambientes voláteis, no qual dispositivos móveis aparecem e desaparecem de maneira repentina; e (ii) ambientes dinâmicos, no qual a estrutura da rede, seus componentes e serviços estão em constante mudança. Para prover um suporte adequado ao desenvolvimento de aplicações ubíquas, surge a necessidade de infraestruturas de software que ofereçam coordenação, interoperabilidade, mobilidade, sensibilidade ao contexto e autonomia [7] [9].

Em um cenário em que há interação entre diversos dispositivos móveis e diferentes provedores de serviços, e em que o usuário pode se mover por diversos contextos e utilizar diferentes dispositivos com diferentes recursos, uma abordagem de coordenação distribuída e desacoplada se torna necessária para garantir adaptação dinâmica [14]. Tratando-se de dispositivos móveis, o desafio a ser enfrentado é a coordenação descentralizada das ações de cada dispositivo para que alcancem um objetivo global.

3. OBJETIVO

O objetivo deste trabalho é propor um sistema que forneça suporte a computação ubíqua, permitindo a comunicação *ad hoc* e adaptação dinâmica ao contexto de seus componentes. Este trabalho propõe uma extensão do SysSU, através do uso de espaço de tuplas distribuído e da implementação de uma camada de comunicação adaptativa que permita acesso a informações contextuais disponíveis em outros dispositivos.

4. CONTRIBUIÇÕES ESPERADAS

Espera-se que, com a utilização de espaço de tuplas distribuído, as aplicações ubíquas construídas sobre o SysSU possam ter mais mobilidade e independência de uma infraestrutura de software centralizada, aumentando assim, a disponibilidade do sistema. Espera-se também que os dispositivos móveis possam se comunicar e se coordenar de forma *ad hoc*, adaptando-se dinamicamente às tecnologias de redes disponíveis.

As tuplas presentes no espaço de tuplas do sistema servem a diferentes propósitos. Elas podem representar informações contextuais, relativas às aplicações e ao próprio sistema de coordenação. Tuplas também representam componentes e serviços disponíveis no sistema, como sensores ou gerenciadores de acesso. As mensagens de coordenação também serão criadas através de tuplas, permitindo que uma aplicação possa ser notificada caso ocorra alguma mudança do espaço de tuplas.

Aplicações embarcadas em um dispositivo poderão ter acesso direto a tuplas criadas por outros dispositivos. Como pode ser visto na Figura 2, uma aplicação que possua acesso local apenas às tuplas T1 e T2, mas necessite de informações contextuais presentes nas tuplas T1, T2, T3 e T4, poderá ter acesso direto às tuplas que estão disponíveis em outro dispositivo, sem a necessidade de um intermediador centralizado.

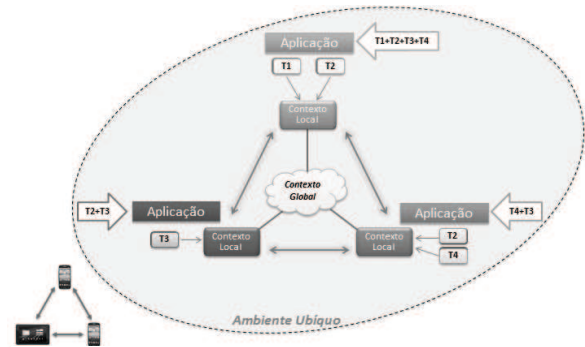


Figura 2. Contexto distribuído.

Com a adoção do modelo de espaço de tuplas distribuído será possível acessar as informações de contexto presentes em vários dispositivos. Isso permitirá a formação de um contexto global (Figura 2), composto por informações contextuais capturadas de forma independente e propagadas de acordo com regras definidas na própria tupla [13], considerando o tempo de vida e o alcance (número de saltos) de cada tupla.

Este trabalho faz parte de um projeto maior chamado LoCCAM (*Loosely Coupled Context Acquisition Middleware*), que está sendo desenvolvido por pesquisadores do GREat² (Grupo de Redes, Engenharia de Software e Sistemas). O LoCCAM é um middleware extensível, configurável, com baixo acoplamento para o gerenciamento de contexto. O LoCCAM é direcionado para dispositivos Android e utiliza o SysSU em sua arquitetura. As contribuições deste trabalho permitirão que o LoCCAM gere informações de contexto distribuídas e troque mensagens entre dispositivos móveis em redes *ad hoc*.

5. METODOLOGIA ADOTADA

Os esforços de estudo e pesquisa deste trabalho visam contemplar as seguintes metas: Revisar a bibliografia relativa à computação ubíqua e aos desafios da coordenação distribuída; Revisar as soluções de ubiquidade oferecidas pelo SysSU; Portar o SysSU para um dispositivo móvel (*smartphone*), permitindo que o *UbiCentre*, com seu espaço de tuplas, opere na plataforma Android (requisito do projeto LoCCAM); Implementar uma camada de comunicação que permita troca de mensagens entre os dispositivos móveis em redes *ad hoc*; Desenvolver um mecanismo de acesso local ao espaço de tupla que funcione como um *UbiBroker* interno; Definir um modelo de representação contextual através de tuplas empregando a especificação semântica padrão do SysSU (requisito do projeto LoCCAM); Definir as regras de propagação das tuplas e regras de inferência baseadas no contexto global; Habilitar o LoCCAM a realizar gerenciamento de contexto distribuído; e Fornecer subsídios para elaboração de trabalhos futuros relacionados ao tema pesquisado.

A metodologia de pesquisa consiste na leitura de artigos; estudo e testes da arquitetura e da implementação do SysSU em plataformas móveis (Android); estudo, implementação e testes de mecanismos de comunicação e descoberta de serviços em redes *ad hoc*; e estudo de técnicas de distribuição de informações de contexto e gerenciamento de contexto global.

²Rossana Andrade, Windson Viana, Marcio Maia, Lincoln Rocha, Benedito Neto, André Fonteles e Rômulo Gadelha.

Para avaliar essa proposta será desenvolvida uma aplicação que exercite as funcionalidades oferecidas pelo SysSU em uma rede *ad hoc*. Será verificado o comportamento do sistema em um cenário de mobilidade, desconexão, compartilhamento de recursos remotos e adaptação a mudanças no contexto. Será realizada uma avaliação de desempenho a partir do tempo gasto para acessar informações contextuais distribuídas traçando um comparativo com a abordagem centralizada.

6. ESTADO ATUAL DO TRABALHO

Atualmente o SysSU foi migrado com sucesso para a plataforma Android, tarefa realizada em conjunto com pesquisadores do projeto LoCCAM. Já é possível realizar testes de acesso direto a um espaço de tupla embarcado em um *smartphone*. A partir daí foi construído um *UbiBroker* interno para acesso local ao espaço de tupla. O próximo passo é a construção de uma camada de comunicação adaptável que permitirá conexão via *Bluetooth* e via *WiFi* de forma *ad hoc*, de acordo com o contexto do sistema. Serão definidas ainda, as políticas de sincronização e propagação nos espaços de tuplas distribuídos.

7. COMPARAÇÃO COM TRABALHOS RELACIONADOS

Os trabalhos listados nessa seção apresentam modelos de coordenação adaptados do modelo *Linda* e baseados em espaço de tuplas distribuído. Eles Introdzem a ideia de reação, baseada em mecanismos de evento, reagindo a mudanças no espaço de tuplas. Foram projetados para sistemas móveis e sensíveis ao contexto, e suportam operações em redes *ad hoc*. Adequando-se, assim, aos principais requisitos da computação ubíqua.

Lime (*Linda in a Mobile Environment*) [12], middleware que aprimora o modelo *Linda* para ser utilizado em ambientes móveis. Adota uma abordagem de espaço de tuplas descentralizado compartilhando as tuplas de acordo com a conectividade de seus componentes móveis. Dessa forma o espaço de tuplas não fica associado a um dispositivo específico. Cada processo possui seu próprio espaço de tuplas, e este é compartilhado com os demais processos locais ou acessíveis através de uma rede de comunicação. O acesso aos espaços de tuplas remotos é feito de forma transparente, sendo que a operação de leitura de uma tupla é distribuída, enquanto a escrita é local.

TOTA (*Tuples On The Air*) [13], middleware projetado com foco na comunicação desacoplada entre componentes de aplicações distribuídas. Uma versão do TOTA é embarcada em cada dispositivo e as tuplas são propagadas pela rede de acordo com regras de propagação presentes na própria tupla. Ao contrário do Lime, a operação de leitura no espaço de tupla é feita de forma local e a escrita é distribuída. O TOTA monitora constantemente a composição da rede, disponibilizando a informação de quem entra e quem sai da rede.

Os trabalhos citados se assemelham a esta proposta, porém não oferecem simultaneamente a interoperabilidade, o desacoplamento e a expressividade semântica existentes no SysSU. Por outro lado, o SysSU não oferece suporte a operações em redes *ad hoc*, uma importante funcionalidade para sistemas ubíquos [12] [13] [14]. Assim, pode-se perceber a relevância de habilitar o funcionamento do SysSU em redes *ad hoc*, agregando estratégias para coordenação em espaço de tuplas distribuído.

8. AGRADECIMENTOS

Este trabalho é um resultado parcial do projeto UbiStructure, financiado pelo CNPq (MCT/CNPq 14/2011 - Universal - 481417/2011-7) e do projeto Learning While Moving (LWM) financiado pelo Ministério de Educação Superior Francês e Ministério da Educação Brasileiro sob o programa de cooperação científica chamado STIC-AmSud. Agradecimentos especiais para Rossana Andrade, Marcio Maia, Windson Viana, Lincoln Rocha, André Fonteles e Rômulo Gadelha.

9. REFERÊNCIAS

- [1] Corradi, A., Lodolo, E., & Monti, S. (2009). Dynamic reconfiguration of middleware for ubiquitous computing. *Dependable mobile ubiquitous*, 7-12.
- [2] Weiser, M. (1991). The computer for the 21st century. *Scientific American*, 265(3), 94–104. New York.
- [3] Hansmann, U., et al. (2003). *Pervasive Computing*. Springer-Verlag, ISBN 3-540-00218-9, Germany
- [4] Lima, F. F. P. (2011). SysSU - Um Sistema de Suporte para Computação Ubíqua. Dissertação de mestrado, UFC.
- [5] Sykes, D., et al. (2008). From goals to components: a combined approach to self-management. *Proceedings of the international workshop on Software engineering for adaptive and self-managing systems* (pp. 1–8). ACM.
- [6] Rocha, R. C. A., Endler, M. (2009). Context Management for Distributed and Dynamic Context-Aware Computing. Program. PhD Thesis, PUC-Rio.
- [7] Maia, M., Rocha, L., & Andrade, R. M. C. (2009). Requirements and challenges for building service-oriented pervasive middleware. *Conference on Pervasive services*.
- [8] Rocha, R. C. A., Endler, M., & Siqueira, T. S. (2008). Middleware for ubiquitous context-awareness. *Proceedings of the 6th international workshop on Middleware for pervasive and ad-hoc computing* (pp. 43–48). ACM.
- [9] Lima, F. F. P., Rocha, L. S., Maia, P. H. M., & Andrade, R. M. C. (2011). Uma Arquitetura Desacoplada e Interoperável para Coordenação em Sistemas Ubíquos. SBCARS
- [10] Carriero, N. and Gelernter, D. (1989). *Linda in context*. *Communications of ACM*, vol. 32, no. 4, pp. 444–458.
- [11] P. T. Eugster, P. a. Felber, R. Guerraoui, and A.-M. Kermarrec. (2003). The many faces of publish/subscribe. *ACM Computing Surveys*, vol. 35, no. 2, pp. 114–131.
- [12] Murphy, A., & Picco, G. (2006). LIME: A coordination model and middleware supporting mobility of hosts and agents. *ACM Transactions on Software*, 15(3), 279-328.
- [13] Mamei, M. (2009). Programming pervasive and mobile computing applications: The TOTA approach. *ACM Transactions on Software Engineering*, 1-53.
- [14] Souza, R. S. D., et al. (2012). Um Modelo de Coordenação Escalável e Proativo para Aplicações Ubíquas. *IV SBCUP*.
- [15] Freitas, G. B. D., & Teixeira, C. A. C. (2010). Uma Arquitetura de Serviços para Aplicações Ubíquas em Redes Domésticas Centrada em TV Digital. *WebMedia'10*.
- [16] da Costa, C. A., Yamin, A. C., & Geyer, C. F. R. (2008). Toward a general software infrastructure for ubiquitous computing. *Pervasive Computing, IEEE*, 7(1), 64–73.