

Services Mobile: Incorporando serviços da Web 2.0 aos dispositivos móveis mediante o uso de informações de contexto

Robson Vinícius V. Sanchez, Ricardo R. de Oliveira, David Fernandes Neto, Renata Pontin M. Fortes, Maria da Graça C. Pimentel
Instituto de Ciências Matemáticas e de Computação - USP
Avenida Trabalhador São-Carlense, 400 - Centro
São Carlos - São Paulo
{rsanchez,ricardoramos,david,renata,mgp}@icmc.usp.br

ABSTRACT

The advent of *Web 2.0* has brought the idea of “Web as platform”, i.e., the software would work through the Internet, no longer only when installed on the machine or user’s mobile device. Thus, the objective of this work is to propose a tool capable of running *Web 2.0* services with context information retrieved from the user’s mobile device, enabling the expansion and customization of the features present in this device, to assist the user in their daily tasks, without the need to change the way they carry out their activities.

RESUMO

O advento da *Web 2.0* propiciou a idéia da “Web como plataforma”, ou seja, os softwares funcionariam através da Internet, não mais apenas quando instalados na máquina ou dispositivo móvel do usuário. Assim, o objetivo deste trabalho é apresentar a ferramenta *Services Mobile* com a proposta de executar serviços da *Web 2.0* a partir de informações de contexto recuperadas do dispositivo móvel do usuário, possibilitando a ampliação e personalização das funcionalidades presentes neste dispositivo, visando auxiliar o usuário em suas tarefas cotidianas, sem que seja necessário mudar a forma como realiza suas atividades.

Categories and Subject Descriptors

C.2.4 [Computer-Communication networks]: Distributed Systems—*Distributed applications*; H.3.5 [Information Storage and Retrieval]: Online Information Services—*Web-based services*

General Terms

Design, Human Factors

Keywords

Ubicomp, Context-Aware, Web 2.0, Mobile, REST

1. INTRODUÇÃO

Segundo Mark Weiser [14] as aplicações são consideradas ubíquas quando são utilizadas pelos usuários de forma transparente por estarem embutidas no ambiente em que eles se encontram. Devido a esta característica, é possível auxiliar as tarefas cotidianas dos usuários de forma espontânea sem interferir na forma como eles realizam suas atividades. Uma forma de atingir esse objetivo é através de aplicações sensíveis ao contexto que são capazes de se adaptar de acordo com as informações de contexto do usuário [13].

De acordo com Dey [6], contexto é qualquer informação que caracterize a situação de uma entidade. Essa entidade pode ser uma pessoa, lugar ou objeto que seja relevante para a interação entre o usuário e a aplicação. Assim, uma aplicação é sensível ao contexto se utiliza essa informação de contexto para prover serviços e/ou informações relevantes para o usuário de acordo com a tarefa que exerce.

As aplicações sensíveis ao contexto podem ser divididas em três categorias de acordo com suas características [6]:

1. Apresentação de informações e serviços para um usuário;
2. Execução automática de um serviço;
3. Marcação da informação de contexto para posterior recuperação.

Neste trabalho, é proposta uma ferramenta sensível ao contexto de acordo com a categoria “execução automática de um serviço” que se baseia nas taxonomias descritas por Schilit et. al [13] e Pascoe [11]. Schilit chamou de *context-triggered actions* e Pascoe chamou de *contextual adaptation*, a capacidade que as aplicações sensíveis ao contexto têm de executar serviços automaticamente de acordo com o contexto corrente, baseado em regras do tipo “se-então”. Assim, o protótipo da ferramenta proposta possibilita executar serviços da *Web 2.0* a partir das informações de contexto recuperadas do dispositivo móvel do usuário, permitindo a ampliação e personalização das funcionalidades presentes neste dispositivo, além de permitir que o conteúdo esteja disponível para o usuário em qualquer lugar e acessível em diversos dispositivos.

2. FUNDAMENTOS TEÓRICOS

Nesta seção são descritos os principais conceitos estudados, que fundamentam a proposta da ferramenta apresentada neste artigo.

2.1 Computação Ubíqua

O termo Computação Ubíqua foi originado por Mark Weiser para referenciar as aplicações às quais o usuário interage de modo transparente por estarem difundidas no ambiente que ele ocupa [3]. Dessa forma, cria-se uma visão de que os ambientes são melhorados com os recursos computacionais, que são capazes de prover informações e serviços aos usuários quando e onde quiserem [1]. De fato, um fator importante descrito por Weiser é a transparência da interação [14].

Há aplicações ubíquas em diversos domínios como educação [2], saúde [7], transporte [4], entre outros. Neste trabalho, nosso foco está voltado para o desenvolvimento de uma aplicação sensível ao contexto que seja capaz de melhorar a experiência do usuário quanto ao uso de serviços nos dispositivos móveis.

2.2 Aplicações Sensíveis ao Contexto

Uma aplicação sensível ao contexto é um sistema computacional cujo comportamento é governado pelo contexto atual do usuário. Ao contrário das aplicações tradicionais, as quais têm seus comportamentos dirigidos apenas pelas entradas dos usuários, um aplicativo sensível ao contexto pode perceber o ambiente atual que o usuário se encontra e/ou avaliar a situação do hardware ou software por meio de sensores, e de acordo com esse “contexto” decidir qual será o seu comportamento de forma automática, seguindo algumas regras pré-definidas [8]. O ponto-chave dessas aplicações é que tornam os dispositivos de computação mais inteligentes para os usuários.

Com a informática tornando-se cada vez mais móvel e pervasiva, é importante que as aplicações e serviços se tornem “conscientes” e sejam capazes de se adaptar a ambientes altamente dinâmicos para melhorar a experiência do usuário final [5].

Neste trabalho, foi projetado e desenvolvido um protótipo da ferramenta capaz de receber informações de contexto de diversos dispositivos móveis, interpretar e executar serviços pré-definidos pelo usuário de forma transparente.

2.3 Web 2.0

O termo Web 2.0 surgiu no ano de 2004, como nome de uma série de conferências realizadas pelas empresas O'Reilly Media e a MediaLive International sobre uma nova geração de comunidades e serviços na Web [10]. O principal conceito abordado nessas conferências era a idéia da “Web como plataforma”, ou seja, os softwares funcionariam através da Internet, não apenas quando instalados na máquina ou dispositivo móvel do usuário. Para alcançar este objetivo são utilizados os Serviços Web.

A ferramenta *Services Mobile*, proposta neste trabalho, fará uso de Serviços Web RESTful disponíveis na Web (por exemplo Google¹, Yahoo!², entre outros) para expandir os serviços disponíveis nos dispositivos móveis e possibilitar a publicação e/ou recuperação de conteúdo que seja pertinente à tarefa realizada pelo usuário.

¹<http://www.google.com>

²<http://www.yahoo.com>

3. TRABALHOS RELACIONADOS

No desenvolvimento de aplicações sensíveis ao contexto existem vários problemas que precisam ser solucionados, entre os quais estão: a recuperação e representação do contexto, interpretação destas representações e modelos de programação capazes de suportar a adaptação dinâmica que as aplicações devem sofrer.

Para a recuperação de contexto normalmente são utilizados *middlewares* capazes de monitorar e inferir o contexto de execução de um determinado dispositivo computacional. Entre esses *middlewares* podemos citar o MoCA [12].

O MoCA (*Mobile Collaboration Architecture*) possui uma infraestrutura que consiste de APIs para clientes e servidores. Os clientes, também chamados de monitores, rodam nos dispositivos móveis, capturando e enviando as informações de contexto recuperadas destes dispositivos aos servidores que estão implantados em nós estáticos da rede, os quais são responsáveis pelo processamento dessas informações. Entre os dados coletados estão a qualidade do sinal, energia disponível, uso de CPU e memória e o *access point* utilizado.

O trabalho de Koskela et al. [9] foi uma das principais inspirações para o desenvolvimento da ferramenta *Services Mobile*. Neste trabalho, os pesquisadores propõem uma arquitetura para o uso de serviços da *Web 2.0* utilizando informações de contexto do usuário. Nessa arquitetura, é proposto um componente para o cadastro e busca dos serviços e que também é responsável por interpretar o contexto fornecido pelo usuário. Há ainda um *middleware* instalado no dispositivo do usuário, que faz a captura e distribuição das informações de contexto no momento da invocação do serviço.

Observando um dos princípios da computação ubíqua que prevê a transparência na interação com o usuário, foi identificada uma limitação na arquitetura proposta por Koskela et al. e que foi adaptada na arquitetura da *Services Mobile*. De acordo com os pesquisadores, a execução dos serviços deve ser realizada explicitamente pelo usuário através do browser. Já na ferramenta *Services Mobile*, a busca e execução dos serviços é realizada por um componente da arquitetura de acordo com o evento realizado pelo usuário em seu dispositivo e em conformidade com regras pré-definidas por ele. Assim, é possível alcançar uma maior transparência na interação com o usuário.

4. ARQUITETURA DA FERRAMENTA

A ferramenta *Services Mobile* foi dividida em 3 componentes:

1. *Mobile Monitor*: componente cliente implantado no dispositivo móvel do usuário que tem como responsabilidade a coleta de informações de contexto e entradas do usuário;
2. *Context Server*: componente servidor responsável por interpretar as entradas e informações de contexto recebidas do *Mobile Monitor* e executar os Serviços Web RESTful pré-definidos;
3. Portal administrativo: portal Web no qual o usuário cadastra as regras de execução dos Serviços Web RESTful de acordo com as suas preferências.

O *Mobile Monitor* é o componente cliente da arquitetura que será instalado no dispositivo do usuário, e portanto será específico para cada plataforma. A sua função principal é recuperar as informações de contexto existentes no momento que o usuário interage com o dispositivo. Essas informações devem ser processadas por este componente e transformadas em um documento XML de acordo com uma semântica definida pela ferramenta *Services Mobile* para representar de maneira uniforme os dados de contexto e entradas produzidas pelo usuário.

O documento XML criado pelo *Mobile Monitor* pode ser visualizado na Listagem 1. Neste documento está representado o usuário que realiza a interação, bem como o evento executado por ele juntamente com as informações de contexto recuperadas e os parâmetros de entrada para a execução dos serviços. Uma vez construído o documento XML, este deve ser enviado em uma requisição HTTP para um Serviço Web disponibilizado pelo *Context Server*.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ServicesRequest>
  <user>usuário</user>
  <action id="addPhoto">
    <contexts>
      <context name="data">
        2011-12-01 10:05:32
      </context>
    </contexts>
    <params>
      <param name="foto">
        Foto codificada algoritmo Base64
      </param>
    </params>
  </action>
</ServicesRequest>
```

Listagem 1: Documento XML

No lado do servidor, o *Context Server* recebe o documento XML de requisição enviado pelo *Mobile Monitor* e extrai todas as informações de contexto e entradas do usuário. Em seguida, de posse desses dados, é realizada uma consulta nas preferências do usuário previamente cadastradas e armazenadas em um banco de dados e, caso alguma delas coincida com o evento reconhecido pelo *Mobile Monitor*, então os serviços Web RESTful cadastrados são executados em ordem sequencial. Atualmente, o *Context Server* não salva as informações contextuais recebidas em nenhum meio persistente, portanto não há como manter um histórico de eventos recebidos de cada usuário.

Para que esse mecanismo cliente-servidor possa funcionar de maneira eficiente, é necessário que as preferências de cada usuário sejam cadastradas em um portal Web. Nesse portal, o próprio usuário pode gerenciar as suas preferências quanto à execução de serviços Web RESTful em resposta a determinados eventos ocorridos em seus dispositivos. Dessa forma, o usuário da ferramenta fica livre para escolher quais os seus serviços preferidos e em quais momentos eles devem ser executados. Todas essas preferências são mantidas em um banco de dados que será consultado pelo *Context Server*.

A ferramenta *Services Mobile* possui ainda uma restrição no que se refere à comunicação entre cliente e servidor, que deve ser realizada sem estado, de tal forma que a requisição do cliente deve possuir toda a informação necessária para que o servidor possa atender o pedido. Assim, é possível melhorar a visibilidade, confiabilidade e escalabilidade da ferramenta. O funcionamento completo da arquitetura

Services Mobile é esquematizado na Figura 1.

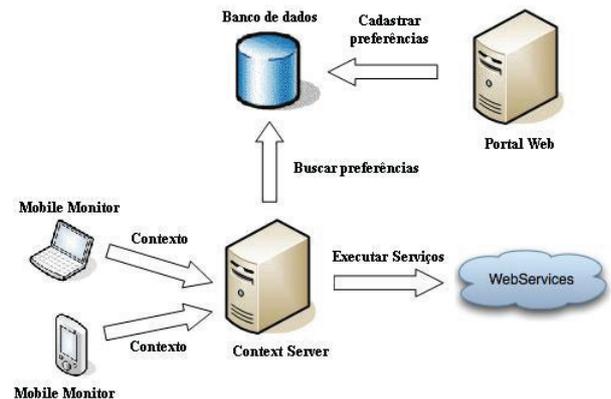


Figure 1: Arquitetura *Services Mobile*

5. RESULTADOS

Uma vez implementada a ferramenta *Services Mobile* conforme a arquitetura proposta na Seção 4, foi conduzido um estudo de caso para a integração dos serviços do *Google Calendar*³, *Google Contacts*⁴ e *PicasaWeb*⁵.

Para esse experimento, foi desenvolvido um *Mobile Monitor* para dispositivos com sistema operacional Windows ou Linux e máquina virtual Java instalada. Esse cliente foi instalado no aparelho do usuário e monitorava as suas ações, detectando a gravação de novas fotos neste dispositivo. Ao ocorrer este evento, uma requisição HTTP era enviada ao *Context Server* com um documento XML que continha como parâmetro a foto serializada pelo método de codificação Base64. Além dessa entrada, a data e a hora em que a foto foi tirada eram incluídas no documento como informações de contexto.

O *Context Server* ao receber a requisição extrai as informações do documento XML, interpreta quais as preferências do usuário para a ação realizada e a partir disso executa os Serviços Web pré-definidos. Nesse experimento, os serviços são executados sequencialmente na seguinte ordem:

1. Busca de evento cadastrado no *Google Calendar* do momento em que a foto foi tirada;
2. Criação de um álbum no *Picasa Web* de acordo com o evento recuperado anteriormente. Se não existir nenhum evento cadastrado, um novo álbum é criado com a data que a foto foi tirada, caso ainda não exista;
3. A foto é inserida com a descrição do evento no álbum criado pelo serviço executado anteriormente.

Como demonstração foi cadastrado um evento no *Google Calendar* e durante o período de ocorrência deste evento uma nova foto foi gravada no dispositivo do cliente, resultando na criação de um álbum no *PicasaWeb* com o nome do evento e a inclusão desta foto no álbum conforme se vê na Figura 2.

³<http://www.google.com/calendar>

⁴<http://www.google.com/contacts>

⁵<http://picasaweb.google.com/home>



Figure 2: Álbum criado conforme evento cadastrado

Este processo foi realizado de forma automática e transparente ao usuário, conforme os princípios de Weiser [14], porém seguindo as preferências definidas por este usuário, auxiliando-o na publicação de conteúdo na Web e permitindo que estes dados estejam acessíveis para outros dispositivos.

O experimento foi executado durante 10 dias com a participação de 15 usuários com conhecimentos avançados em computação, servindo apenas como prova de conceito do trabalho e para validar o funcionamento da ferramenta. Durante esse período, os usuários instalaram o *Mobile Monitor* em seus dispositivos móveis e foram estimulados a utilizar o *Google Calendar* para cadastrar os seus compromissos e também a tirar fotos durante esses eventos.

Analisando o resultado desse experimento, foi possível observar que a ferramenta *Services Mobile* atendeu às expectativas quanto a execução dos serviços propostos, com um percentual mínimo de falhas (inferior a 1%) e também mostrou-se eficaz para a publicação de conteúdo na Web.

6. CONCLUSÕES

Neste trabalho foi apresentada a *Services Mobile*, uma ferramenta que visa ampliar as funcionalidades existentes nos dispositivos móveis através da utilização de serviços da *Web 2.0*. Como a execução desses serviços geralmente é feita com uma entrada limitada de dados e deve ser realizada pelo usuário de forma explícita, constata-se que a forma com que o usuário interage para realizar suas atividades é modificada.

Assim, a ferramenta *Services Mobile* foi viabilizada para permitir que os serviços da *Web 2.0* sejam executados de maneira transparente ao usuário e com informações sobre o “contexto” no qual este usuário se encontra.

Outro diferencial encontrado na ferramenta é a adoção da arquitetura cliente-servidor, que melhora a escalabilidade e evolução dos componentes, pois divide as responsabilidades de aquisição e interpretação de contexto. Assim, é possível que os componentes clientes sejam aperfeiçoados para atender um número crescente de dispositivos móveis, e da mesma forma, o componente servidor pode evoluir permitindo a execução de uma quantidade maior de serviços da *Web 2.0*.

Os experimentos iniciais apresentados na Seção 5 deram indícios do potencial da ferramenta, mas novos experimentos devem ser realizados e avaliados para verificar a viabilidade de utilização da *Services Mobile*.

7. REFERENCES

[1] G. Abowd, E. Mynatt, and T. Rodden. The human experience. *Pervasive Computing, IEEE*, 1(1):48 – 57, jan-mar 2002.

[2] F. L. Arantes, C. R. Moraes, S. H. P. Silva, R. P. M. Fortes, and M. d. G. C. Pimentel. Where and with whom do you wanna meet? session-based collaborative work. In *Proceedings of the 16th Brazilian Symposium on Multimedia and the Web, WebMedia '10*, pages 123–130. ACM, 2010.

[3] V. Bahl. A reflection on mark weiser. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3:1, July 1999.

[4] S. Cheng and M. Trivedi. Turn-intent analysis using body pose for intelligent driver assistance. *Pervasive Computing, IEEE*, 5(4):28–37, oct.–dec. 2006.

[5] A. Devaraju, S. Hoh, and M. Hartley. A context gathering framework for context-aware mobile solutions. In *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology, Mobility '07*, pages 39–46, New York, NY, USA, 2007. ACM.

[6] A. K. Dey. Understanding and using context. *Personal Ubiquitous Comput.*, 5:4–7, January 2001.

[7] J. R. B. Diniz, C. A. G. Ferraz, F. A. M. Trinta, H. N. Melo, and L. M. Santos. Avaliação de um serviço de gerenciamento de sessão para ambientes de medicina ubíqua. In *Proceedings of the 14th Brazilian Symposium on Multimedia and the Web, WebMedia '08*, pages 4–11, New York, NY, USA, 2008. ACM.

[8] W. Du and L. Wang. Context-aware application programming for mobile devices. In *Proceedings of the 2008 C3S2E conference, C3S2E '08*, pages 215–227, New York, NY, USA, 2008. ACM.

[9] T. Koskela, N. Kostamo, O. Kassinen, J. Ohtonen, and M. Ylianttila. Towards context-aware mobile web 2.0 service architecture. In *Proceedings of the International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pages 41–48, Washington, DC, USA, 2007. IEEE Computer Society.

[10] T. O'Reilly. What is web 2.0, 2005.

[11] M. J. Pascoe. Adding generic contextual capabilities to wearable computers. In *Proceedings of the 2nd IEEE International Symposium on Wearable Computers, ISWC '98*, pages 92–99, Washington, DC, USA, 1998. IEEE Computer Society.

[12] V. Sacramento, M. Endler, H. Rubinsztein, L. Lima, K. Goncalves, F. Nascimento, and G. Bueno. Moca: A middleware for developing collaborative applications for mobile users. *Distributed Systems Online, IEEE*, 5(10):2, oct. 2004.

[13] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*, pages 85–90, Washington, DC, USA, 1994. IEEE Computer Society.

[14] M. Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3:3–11, July 1999.