

Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM

Framework de comunicação para
Webservices P2P

Aluno: Brayan Vilela Alves Neves
Matricula: 07.1.4159

Orientador: Ricardo Augusto Oliveira Rabelo

Ouro Preto
19 de dezembro de 2011

Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM

Framework de comunicação para Webservices P2P

Relatório de atividades desenvolvidas apresentado ao curso de Bacharelado em Ciência da Computação, Universidade Federal de Ouro Preto, como requisito parcial para a conclusão da disciplina Monografia I (BCC390).

Aluno: Brayan Vilela Alves Neves
Matricula: 07.1.4159

Orientador: Ricardo Augusto Oliveira Rabelo

Ouro Preto
19 de dezembro de 2011

Resumo

Recentemente os *softwares* móveis estão em foco na área de pesquisa e desenvolvimento. Há hoje uma gama de novos sistemas sendo propostos para conseguir o mesmo sucesso dos programas utilizando a nuvem como meio de comunicação, inerente a isso temos custos relacionados ao acesso ao servidor central, tanto financeiro quanto de energia necessária. Neste trabalho vamos explorar a descentralização da informação por meio da propagação vertical dos dados utilizando *webservices* sobre conexões ponto-a-ponto (P2P). O trabalho visa a criação de meta-aplicações móveis no qual é possível o compartilhamento de funcionalidades através de *webservices* sobre P2P. A aplicação será desenvolvida para o sistema operacional Android, um sistema operacional para dispositivos móveis, e fará comunicação via Bluetooth. Será medido também os níveis de bateria durante as transmissões e também como se comporta a fila de requisições a um aparelho quando existem muitas requisições pela informação que ele possui além do problema da sincronização de bancos de dados em vários locais.

Palavras-chave: BPM. Sincronização. *Webservice*. P2P. *Framework*. *Widgets*. Meta-aplicação.

Sumário

| | | |
|----------|---------------------------------|----------|
| 1 | Introdução | 1 |
| 2 | Justificativa | 2 |
| 3 | Objetivos | 3 |
| 3.1 | Objetivo geral | 3 |
| 3.2 | Objetivos específicos | 3 |
| 4 | Metodologia | 4 |
| 4.1 | BPMFramework | 4 |
| 4.2 | Sincronização | 5 |
| 5 | Desenvolvimento | 6 |
| 5.1 | Implemetação | 7 |
| 6 | Trabalhos Futuros | 8 |
| 7 | Cronograma de atividades | 8 |

Lista de Figuras

| | | |
|---|--------------------------------------|---|
| 1 | Exemplo de um processo BPM | 5 |
| 2 | Módulos do BPMFramework | 6 |
| 3 | Árvore gerada pelo BPMF | 7 |

Lista de Tabelas

| | | |
|---|-----------------------------------|---|
| 1 | Cronograma de Atividades. | 8 |
|---|-----------------------------------|---|

1 Introdução

Junto com o rápido desenvolvimento mundial da Tecnologia da Informação, sistemas de informações para todos os fins são necessários em todos os lugares, mas o que acontece com grande parcela dos sistemas é que eles não conseguem atender a expectativa ou então acabam entrando em colapso por diversas causas, como a não extração correta dos requisitos do usuário ou a não adaptação a feroz mudança do mercado de negócios [6].

Prestando atenção a questão de que o sistema poderá rodar em dispositivos com configurações diferentes, por exemplo o tamanho de tela e capacidade de processamento, hoje, um cliente que planeja construir um sistema de informação espera que o sistema tenha flexibilidade e estar pronto para personalizações além dos pontos clássicos: função, estabilidade, segurança e eficiência [5].

Além disso, temos uma nova perspectiva onde o cliente poderá usar seu sistema de qualquer lugar, mas nem sempre poderá ter conexão ao servidor central por conta de economia de recursos do dispositivo ou até indisponibilidade de rede.

Sendo questão da bateria hoje no mundo dos dispositivos móveis é um assunto crítico, uma vez que não se tem uma fonte de energia por perto sempre, o mau planejamento do uso dos recursos do dispositivo bem como a arquitetura e sua rede pode fazer com que a bateria se esgote rapidamente.

Então para que suas informações sejam consistentes com o servidor central, ele precisaria de usar outros meios de comunicação para atualizá-las, o que será o foco do estudo nesse projeto, a atualização das informações de maneira vertical, ou seja, dispositivos se comunicando ponto-a-ponto a fim de sempre se manterem atualizados.

Este trabalho será dividido em duas partes. A primeira parte consiste em criar um *Framework* para gerar meta-aplicações de Gerência de Processo de Negócios (*Business Process Manager* - BPM) combinado com uma Arquitetura Orientada a Serviços (*Service-oriented Architecture* - SOA) onde segundo [4], resolveria nosso primeiro problema, a flexibilidade.

E a segunda parte seria a criação do *testbed* de sincronização de informações, visando pesquisar qual a melhor arquitetura de sincronização poderemos utilizar no *Framework* proposto.

2 Justificativa

Tendo em vista as dificuldades já apresentadas que teremos em nossa rede, é preciso descobrir qual arquitetura a rede terá para manter a integridade do banco de dados em todos os dispositivos economizando o máximo possível de recursos.

3 Objetivos

3.1 Objetivo geral

- O objetivo geral do projeto é desenvolver um *framework* de comunicação via P2P onde os usuários possam compartilhar tanto dados para uma aplicação quanto uma meta-aplicação que poderá ser usada pelos vários usuários do sistema. Nossos estudos serão feitos através do funcionamento deste *framework*.

3.2 Objetivos específicos

- Criar uma aplicação Android que permita a criação de meta-aplicações dinâmicas e o compartilhamento das mesmas.
- Criar uma *testbed* para simular quais tipos de arquiteturas funcionariam melhor com nossa aplicação descrita.
- A partir da nova arquitetura, verificaremos se realmente conseguiremos atingir nosso objetivo de fazer a transmissão da informação de forma rápida e com menor custo de bateria do que a forma tradicional, via acesso à um servidor na nuvem.
- Determinar a velocidade com que a informação chegará aos usuários da aplicação com e sem a presença do servidor na nuvem.

4 Metodologia

O trabalho é dividido em duas partes, a primeira é a construção de um *framework* em Android que permite a criação e execução de meta-aplicações através de um arquivo XML. Chamaremos ele de *Business Process Manager Framework* (BPMFramework), e a segunda é a construção de um *testbed* para saber qual a melhor arquitetura usar para sincronização de informações no primeiro programa.

4.1 BPMFramework

Business Process Management (BPM) é um tipo de aplicativo na TI que fornece estratégias de integração entre pessoas, entre sistemas e entre pessoas e sistemas correspondentes as mudanças do ambiente de negócio. É composto de vários processos, como modelagem, monitoramento e gerenciamento.

O principal objetivo de um processo BPM é a automação e a otimização. Quando se automatiza processos, rapidamente é possível obter-se um controle mais rígido e adaptado às necessidades da empresa. Através da automação, um serviço melhor é oferecido ao cliente, dada a rapidez e organização que a empresa passará a apresentar. Além disso, terá seus custos reduzidos.

A modelagem de um processo é feita através de uma *Business Process Management Suite* - BPMS (que será trabalho futuro). Para a criação de um processo, o BPMS respeita as notações da BPM Notation [1], que nada mais é do que um conjunto de regras, ícones e padrões de processos que facilitam o entendimento do usuário.

A partir de um modelo criado pelo usuário é possível simular como um processo irá funcionar e corrigir possíveis falhas do mesmo. A partir de um modelo delegamos tarefas para outros usuários da aplicação e é possível fazer o monitoramento do andamento de cada tarefa que está ocorrendo. Tendo total visão de produtividade e desempenho da equipe.

A otimização do processo é uma parte crucial quando se trata de BPM. É essencial que sejam feitas melhorias no processo, pois com o processo otimizado pode-se chegar a resultados positivos com mais agilidade.

Neste projeto será feita a implementação de um programa capaz de interpretar arquivos XML que seguem a BPMN 1.2 adaptado, transformá-los em processos, e executá-los.

A estrutura de um processo em um BPM é dada por um conjunto de *Pools*, cada *Pool* agrupa um conjunto de *Lanes* que contenham *Tasks* associadas e cada *Task* é uma parte da execução do processo. Por exemplo, numa oficina mecânica, a própria oficina poderia ser uma *Pool*, e as *Lanes* seriam as equipes de trabalho onde cada uma delas pode realizar certos tipos de tarefas (*Tasks*).

É importante lembrar que cada grupo pode conter sub-grupos, logo as *Lanes*

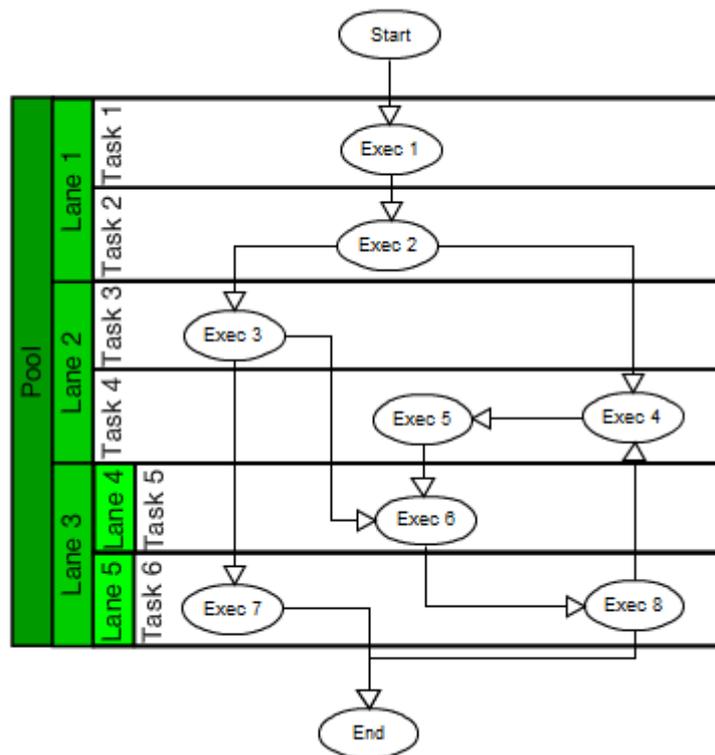


Figura 1: Exemplo de um processo BPM

podem conter *Sub-Lanes*, onde um usuário que pertença a *Lane* principal, pode realizar todas as tarefas das *Sub-Lanes* dela, mas se ele for de uma *Sub-Lane* ele não poderá executar as tarefas pertencentes a *Lane* principal ou suas outras *Sub-Lanes*.

Cada *Task* contém um *Gateway* associado, este *Gateway* é quem diz qual a próxima tarefa deverá ser executada ou se é o fim do processo. Um *Gateway* também pode ligar tarefas de *Pools* diferentes ou até de processos diferentes.

4.2 Sincronização

O *testbed* de sincronização será tratado em BCC 391 - Monografia II.

5 Desenvolvimento

O desenvolvimento do BPMF foi dividido em módulos.

- Módulo de conexão: responsável pela comunicação com o servidor principal.
- Módulo de gerenciamento de usuários: responsável por autenticar usuários e suas respectivas restrições.
- Módulo servidor: um gerenciador de webservices em *Rails* [3] que responde as requisições feitas pelo módulo de conexão.
- Módulo administrador: responsável pela criação, execução e monitoramento de tarefas no processo.
- Módulo cliente: responsável por disponibilização de tarefas para os usuários.
- Módulo execução: responsável por interpretar um tarefa designada a um cliente.
- Módulo de interfaces: responsável por auxiliar na construção das interfaces do sistema.
- Módulo de integração: responsável por integrar os demais módulos.

Cada dispositivo terá uma réplica do banco de dados central.

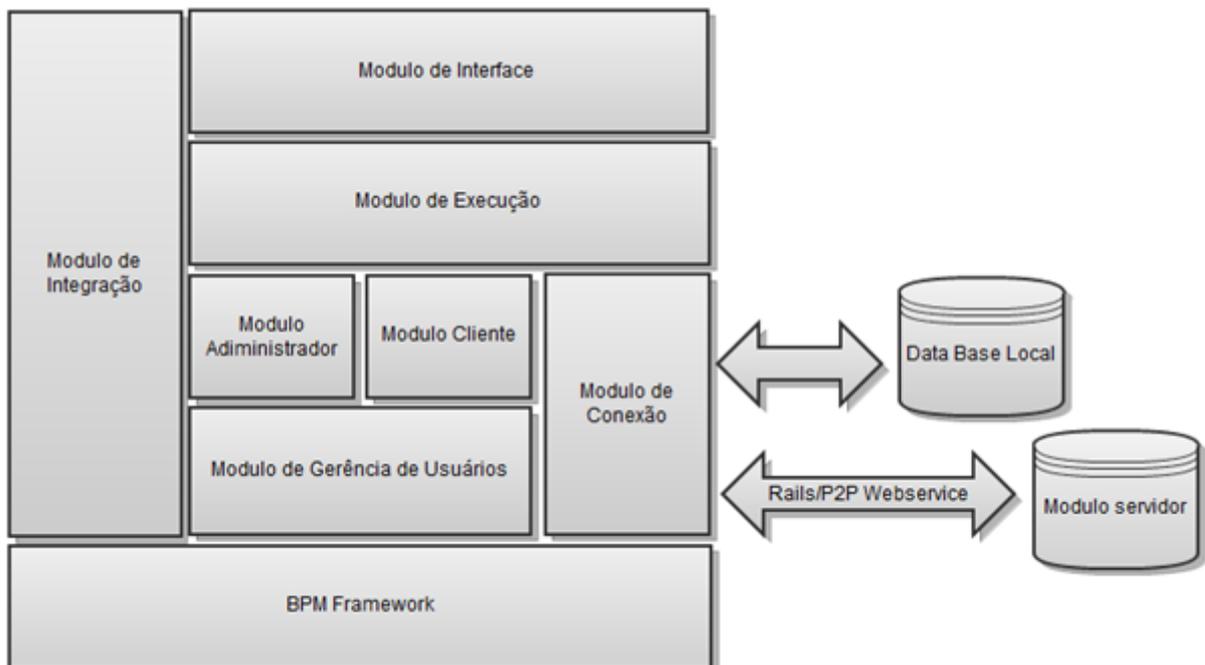


Figura 2: Módulos do BPMFramework

5.1 Implementação

O primeiro módulo criado foi o módulo de execução, em uma versão simplificada. Este módulo lê um XML e gera para o cliente uma tela com a tarefa em execução. Esta tarefa pode receber ou não parâmetros de entrada, mas sempre retorna alguma coisa.

O interpretador ao ler o XML, gera uma árvore de objetos filhos de *BPMGenericElement* que é um objeto que contém uma *LinkedHashMap<String,String>*, no mesmo padrão do XML, com a diferença que é possível inserir dados facilmente nela assim como lê-los. Ao final de cada tarefa, a resposta obtida é esta árvore com os dados estruturados.



Figura 3: Árvore gerada pelo BPMF

Após isso foi construído um servidor em *Rails* [3], este servidor é um repositório onde é guardado os XMLs de processos, e um banco de dados MySQL [2] que

armazenará as informações dos processos. Junto com o servidor foi construído o Módulo de conexão que irá fazer com que a aplicação possa acessar os serviços do servidor.

Podendo estabelecer conexões com o servidor foi feito o Módulo de gerência de usuários, que irá dizer qual usuário tem permissão para fazer cada tipo de tarefa de acordo com seu grupo e nível de permissão (administrador ou cliente).

Cada XML armazenado no servidor representa um processo, e ele deverá criar no banco uma tabela para cada um. Assim todo dado retornado de uma tarefa é armazenado em sua tabela correspondente.

O módulo de interfaces auxilia como as telas estáticas do módulo de integração devem ser construídas, dizendo os pré-requisitos mínimos que cada tela deve ter, por exemplo uma tela de *login* precisa ter um campo para digitar usuário, outro para senha, um botão para confirmar e uma atividade de saída.

6 Trabalhos Futuros

O primeiro trabalho futuro é o refinamento do interpretador de XMLs, uma vez que de início foi feita uma versão simplificada para testes de prosseguimento do projeto nos outros módulos. Após isto o desenvolvimento do módulo de administrador onde será possível criar um diagrama de fluxos que irá gerar os XMLs de processos, e gerenciá-los.

Após a conclusão das tarefas acima será dado início a segunda parte do projeto que consiste em criar uma *testbed* onde teremos como resultado os tipos de arquitetura de sincronização de informações mais viável na aplicação definida.

7 Cronograma de atividades

Na Tabela 1, cronograma para a conclusão da monografia na disciplina BCC391 - Monografia II.

| Atividades | Jan | Fev | Mar | Abr | Mai | Jun |
|---|-----|-----|-----|-----|-----|-----|
| Termino do BPMF | X | X | | | | |
| Construção do <i>testbed</i> de sincronização | | X | X | X | | |
| Simulações e testes | | | | X | | |
| Redigir a Monografia | | | X | X | X | X |
| Apresentação do Trabalho | | | | | | X |

Tabela 1: Cronograma de Atividades.

Referências

- [1] Business process manager notation. <http://www.bpmn.org/>, 2011.
- [2] Mysql. <http://www.mysql.com/>, 2011.
- [3] Ruby on rails. <http://rubyonrails.org/>, 2011.
- [4] Yang Liu et al. Architecture of information system combining soa and bpm. *IEEE Information Management, Inovation Management and Industrial Engeeniering*, pages 42 – 45, 2008.
- [5] Chancheng Hu. We're on the highway of "agility". *Programmer*, pages 49 – 51, 2007.
- [6] Xiyuan Wan and Bipeng Li. Studies on agile architecturing for mis. *Computer Science*, pages 111 – 113, 2006.