

Universidade Federal de Pernambuco

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CENTRO DE INFORMÁTICA

2011.1



Desenvolvimento de Aplicações Web
Sob o Modelo de Social Machines

Trabalho de Graduação

Aluno
Orientador

João Vitor Oliveira Batista
Sílvio Romero de Lemos Meira

{jvob@cin.ufpe.br}
{srlm@cin.ufpe.br}

13 de Dezembro de 2011

Universidade Federal de Pernambuco

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CENTRO DE INFORMÁTICA

2011.1

Desenvolvimento de Aplicações Web
Sob o Modelo de Social Machines

Trabalho de Graduação

Trabalho de graduação apresentado no Centro de Informática da Universidade Federal de Pernambuco por João Vitor Oliveira Batista, orientado por Sílvio Romero de Lemos Meira, como requisito para a obtenção do Grau de Bacharel em Ciência da Computação.

Orientador

Sílvio Romero de Lemos Meira

{srlm @cin.ufpe.br}

13 de Dezembro de 2011

Aos meus pais, Manuel e Lúcia Helena.

Agradecimentos

“Quod per sortem, sternit fortem. Mecum omnes plangite”.
Carl Orff

Em primeiro lugar, o agradecimento é à minha família, que em nenhum momento deixou de me apoiar neste empreendimento. Em especial agradeço a meus pais, principais responsáveis pela educação e pela formação que me trouxeram aqui. Deixo também um lugar especial para meus irmãos, aos quais me inspiro a cada passo que dou em minha vida.

Aos meus amigos e colegas e professores, deixo o meu fiel agradecimento. Sem eles, nada teria valido a pena.

Resumo

O desenvolvimento de aplicações voltadas para a internet está se tornando uma das principais áreas de atuação da engenharia de software no cenário atual. Para atender a essa crescente demanda no desenvolvimento de software para esta plataforma, são necessárias novas maneiras de projetar e modelar aplicações.

Este trabalho propõe um estudo descritivo do modelo de Social Machines para aplicações web. Será apresentado todo o modelo em detalhes, bem como suas características e propriedades. Uma avaliação completa dos pontos fortes e fracos do modelo foi elaborada. Além disso, um comparativo entre este modelo e outros modelos usados atualmente será apresentado.

Palavras-chave: Social Machines, Internet, Engenharia de *Software*, Aplicações Web, Projeto de Software.

Abstract

The development of software for internet is becoming one of the most prominent working areas of software engineering nowadays. To meet this growing demand in the development of software for this platform, new ways to project and to model applications are strictly necessary.

This paper proposes a descriptive study of the model of Social Machines for web applications. The whole model will be presented in detail, as well as its characteristics and properties. A complete review of the features and weaknesses of the model is done. Furthermore, a comparison between this model and other models used today is presented.

Keywords: Social Machines, Internet, Software Engineering, Web Applications, Project of software.

Sumário

1 - Introdução	9
2 - Máquinas Sociais	12
2.1 - Propriedades	13
2.1.1 – Conexões	13
2.1.2 – Wrapper Interface	14
2.1.3 – Requisições (Requests)	14
2.1.4 – Respostas (Responses).....	14
2.1.5 – Estados.....	15
2.1.6 – Restrições (Constraints).....	15
2.1.7 – Entrada (Input)	15
2.1.8 – Saída (Output)	15
2.1.9 – Unidade de Processamento	15
2.2 - Características Principais.....	16
2.2.1 – Sociabilidade.....	16
2.2.2 – Composicionabilidade.....	16
2.2.3 – Independência de Plataforma	17
2.2.4 – Independência de Implementação	17
2.3 – Classificação.....	17
2.3.1 – Isoladas	17
2.3.2 – Consumidoras	17
2.3.3 – Provedoras	17
2.3.4 – Híbridas	18
3 – Processo de Projeto.....	19
3.1 – Definição do Aplicativo	19
3.1.1 – Reuso de Componentes.....	20
3.2 – Definição das Conexões	20
3.3 – Definição da Interface e Propriedades.....	20

3.4 – Detalhamento	21
4 – Estudo de Caso.....	22
4.1 – Descrição Simplificada dos Aplicativos	22
4.1.1 – ERP	23
4.1.2 – CRM	24
4.1.3 – Loja Virtual	25
4.2 – Considerações do Estudo de Caso	26
5 – Conclusão.....	28
5.1 – Trabalhos Futuros	29

1 – Introdução

O cenário de desenvolvimento de software nunca esteve num cenário tão favorável para a programação voltada para a serviços web. Este cenário pode muito bem ser observado tendo em vista a franca expansão das redes sociais e das plataformas de computação nas nuvens. Levando-se em consideração a quantidade de horas semanais gastas em atividades, em breve a internet se tornará a principal atividade responsável pelo tempo dos jovens, tomando cerca de 50% do tempo diário [3].

Previsões em relação ao crescimento do volume de dados da rede, estimam [1] que o tráfego de dados na Internet quadruplique seu volume até o ano de 2015, alcançando a marca de 8 Zettabytes. Dentre os principais responsáveis por este aumento figuram os dispositivos móveis. A figura a seguir mostra o crescimento do volume de dados para cada tipo de dispositivo.

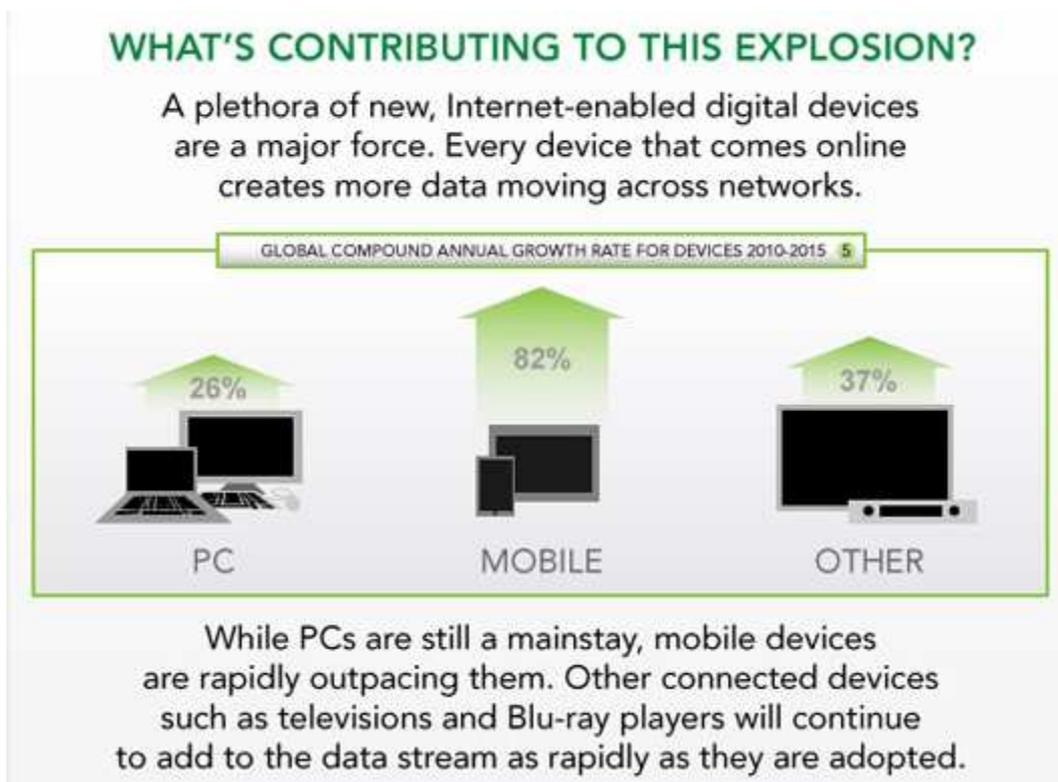


Figura 1: Crescimento dos dados e suas fontes de origem

Devido a esta mudança na forma de uso de software, com o mesmo passando a ser utilizado em mais tipos de dispositivos e se tornando cada vez mais pervasivo e ubíquo, é preciso mudar a abordagem da engenharia de software e sistemas dos dias atuais, de forma a atender à esta nova necessidade de softwares.

Com o aumento da disponibilidade de serviços disponíveis na rede [12], o desenvolvimento de aplicativos web tem se assemelhado cada vez mais ao conceito discutido por Wade Roush[13], de que computação significará conexão entre softwares.

Esta tendência é observada na evolução das tecnologias web. No início da internet, a web era praticamente conteúdo estático, apresentados sob uma forma de interação somente leitura, hoje, é comumente chamada de web 1.0. O desenvolvimento de novas tecnologias, como o AJAX, permitiu o aparecimento de sites com interação em mão dupla, incluindo a produção de conteúdo por parte do usuário final. O conjunto de conceitos e técnicas de programação que permitem a participação ativa do usuário na criação de conteúdo, é hoje chamada web 2.0.

Uma nova tendência de mudança está se consolidando atualmente: a estruturação de uma web programável e inteligente. este próximo estágio da web é comumente chamada de web 3.0. Neste novo cenário, a forma como o software é desenvolvido, distribuído e usado, muda drasticamente. É cada dia mais comum que o software seja planejado e desenvolvido para a rede, pela rede, na rede e fazendo uso da rede. Estas características trazem a necessidade de um novo modelo de entendimento de software. A proposta do conceito de máquinas sociais é assumir este papel, permitindo que software conectado, distribuído e programável seja desenvolvido de forma simples e eficaz. Vários exemplos de software que já são compatíveis com esta maneira de desenvolvimento, mantendo uma API de serviços disponível para interação com outros softwares pode ser encontrada em [12].

Esta nova onda de softwares programáveis está sendo permitida pelo desenvolvimento de arquiteturas como SOA e REST, além de novas formas de prover serviço, como Software como Serviço e Computação nas Nuvens. Cada

uma destas tecnologia é de fundamental importância para o desenvolvimento destas aplicações, entretanto, estes modelos tem tido certa dificuldade em representar o desenvolvimento de aplicações que possuam interação efetiva entre si, mostrando-se necessário o desenvolvimento de novos conceitos para possibilitar essa nova fase de inovação em software que está surgindo.

Este trabalho pretende apresentar o conceito de máquinas sociais, bem como um guia prático para seu uso no estado atual de desenvolvimento. É importante salientar que este conceito ainda não está consolidado em sua forma teórica, sendo, este trabalho, um estudo de seu mais recente estado de desenvolvimento, conforme descrito em [11].

Além deste capítulo de contextualização e motivação, o presente trabalho é organizado na seguinte estrutura:

No capítulo 2 é descrito o conceito de máquinas sociais, bem como suas propriedades e características principais. Neste capítulo é possível formar uma visão estruturada de como é proposto o modelo, quais suas implicações e quais os objetivos principais do uso do mesmo.

No capítulo 3 é proposta uma dinâmica de projeto para o conceito abordado anteriormente. são apresentadas as etapas de projeto para modelagem de uma máquina social, bem como indicação de como proceder em cada uma das etapas.

No capítulo 4 é apresentado um estudo de caso, com aplicações exemplo. Neste capítulo, pode-se observar a descrição de um software de acordo com o conceito de máquina social e com a metodologia proposta por este trabalho.

O capítulo 5 é uma síntese dos resultados obtidos no estudo, bem como uma avaliação crítica do método, com a apresentação de pontos fortes e fracos do conceito, e as oportunidades que podem ser abertas com o projeto de software através de máquinas sociais e as ameaças que podem prejudicar softwares projetados de acordo com este conceito.

Por fim, o capítulo 5 consiste de uma conclusão dos resultados deste trabalho assim como a enumeração dos próximos passos para estender e melhorar os resultados obtidos neste trabalho.

2 - Máquinas Sociais

O cenário de desenvolvimento de software atual está mudado. É crescente o número de aplicações web que fazem uso de Interfaces de outros aplicativos para prover seus serviços. Esta tendência de web programável vem mostrando a necessidade crescente de criar um novo entendimento do projeto de software, que promova e facilite o entendimento da interação entre os aplicativos.

De forma geral, o conceito de Máquina Social representa uma nova forma de representar software, fazendo-o de forma a enxergá-lo como um serviço conectável em uma rede. Neste modelo, estas entidades são representadas através de uma estrutura composta de oito componentes: Inputs, Outputs, Requests, Responses, Processing Unit, States, Constraints e Wrapper Interface.

Wrapper Interface é a representação da estrutura responsável por esperar a chegada de novas Requests e responder, se necessário, através de Responses a outras máquinas sociais. A Unidade de Processamento representa o corpo central do software e é responsável por processar as entradas e produzir saídas. A Unidade de processamento possui conexão com outras Máquinas sociais. Estas conexões podem ser permanentes ou intermitentes.

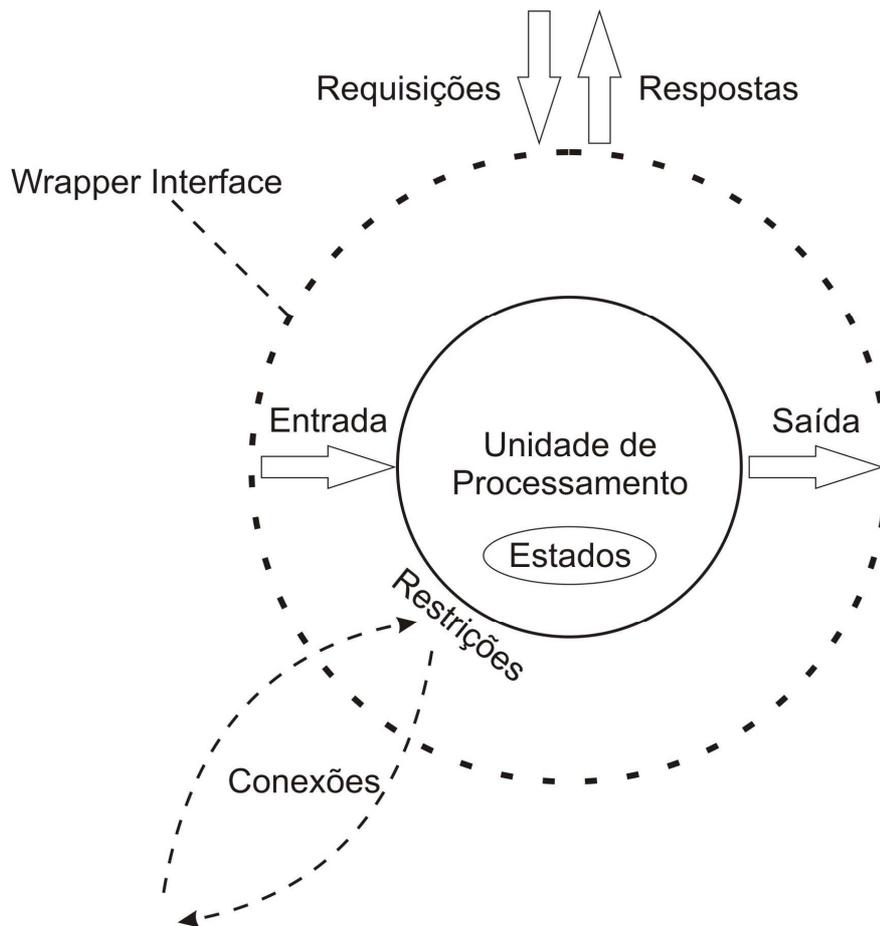


Figura 2: Representação gráfica de uma Máquina Social

2.1 - Propriedades

Neste tópico estão descritas detalhadamente cada uma das propriedades de uma máquina social. Cada uma de suas propriedades é utilizada como indicador no processo de desenho do software.

2.1.1 - Conexões

As conexões entre máquinas sociais são o estabelecimento de uma relação de confiança entre duas máquinas. Em tese, quaisquer duas Máquinas Sociais são capazes de se conectar através do uso de algum protocolo de comunicação bem definido, desde que sejam satisfeitas as devidas restrições impostas por cada Máquina Social.

2.1.2 - Wrapper interface

Esta é a camada de comunicação com que uma máquina social se torna apta a comunicar-se com outras máquinas ou com o usuário. Ela pode ser caracterizada através de uma API ou de uma interface com o usuário. Através dessa interface é possível acessar os serviços de uma máquina social ou um subconjunto específico deles, para o caso de conexões com finalidade restrita.

2.1.3 - Requisições (Requests)

Possuem o seu entendimento alinhado com o já consolidado em engenharia de software. Requisições são chamadas remotas (RPC) de algum dos procedimentos do aplicativo em questão. Requisições podem ou não conter parâmetros de entrada, dependendo do procedimento a ser invocado. No padrão de máquinas sociais, as requisições podem ser classificadas entre dois tipos:

Requisição de Funcionalidade: Responsáveis pela chamada de funções implementadas pela máquina social.

Requisição de Meta-Informações: Responsáveis pela solicitação de informações sobre a própria máquina social. Estas informações podem ser descritivas sobre a funcionalidade da máquina social, sobre seu estado atual ou sobre suas restrições de uso. Além disso, estas requisições são bastante úteis para negociação entre máquinas sociais, pois podem prover informação sobre custo e garantias do serviço ofertado.

2.1.4 - Respostas (Responses)

Possuem o entendimento alinhado com o já consolidado em engenharia de software. Respostas podem ser consideradas como resultado da interação entre duas máquinas sociais ou com o usuário através da interface de comunicação (Wrapper Interface). Responses podem ser classificadas em três tipos:

- Resposta de Funcionalidade: Uma resposta de funcionalidade é o resultado direto de uma requisição de funcionalidade.

- Resposta de Meta-Informações: Uma resposta de Meta-informação é o resultado direto de uma requisição de Meta-Informações.
- Resposta de Notificação: Uma mensagem originada sem uma requisição, geralmente para sinalizar uma mudança de estado de funcionamento para as outras máquinas sociais conectadas. Pode conter informações sobre erros de execução ou problemas com a comunicação com a interface.

2.1.5 - Estado

Esta propriedade é a descrição da situação de funcionamento atual da máquina social. Uma máquina social pode manter ou não o registro de estado, sendo possível o projeto de máquinas sem estado.

2.1.6 - Restrições (Constraints)

Nesta propriedade são descritas as limitações de uso de um aplicativo. Estas limitações vão desde a determinação de limites, como carga máxima, a características de funcionamento, como regras de segurança.

2.1.7 - Entrada (Input)

Informação que será utilizada como parâmetro de entrada para a unidade de processamento na execução do serviço requisitado.

2.1.8 - Saída (Output)

É o resultado do processamento da máquina social após o processamento de uma determinada entrada. É possível que, para alguns serviços, nenhuma saída seja gerada.

2.1.9 - Unidade de processamento

É a estrutura responsável por prover as funcionalidades centrais de uma máquina social. A unidade de processamento pode implementar essas funcionalidades através de vários serviços ou pela combinação de serviços de outras máquinas sociais. A implementação da máquina social tem seu teor de processamento de informação nesta estrutura.

2.2 - Características Principais

Com o entendimento das propriedades e da estrutura esquemática do conceito de máquinas sociais, é possível definir e descrever algumas de suas características mais destacadas.

2.2.1 – Sociabilidade

Pela própria definição de máquina social, o objetivo central é que uma das qualidades do software seja a existência de comunicação efetiva entre aplicações. O desenho de software através do modelo social machines não somente é uma fundamentação para descrever o compartilhamento e reuso de software na rede mas deveria ser o caminho padrão de desenvolvimento das máquinas sociais. Obter vantagem do ambiente de rede em que o aplicativo está inserido é a idéia que torna o conceito de máquinas sociais tão atraente para viabilizar e facilitar o reuso e a combinação de aplicações como forma de implementar novas máquinas com a menor quantidade de retrabalho possível. Softwares desenhados pelo modelo de máquinas sociais também podem ser independentes de outras máquinas, porém, isto não segue a ideologia apresentada no conceito, de compartilhamento e reuso.

2.2.2 – Composicionabilidade

Esta característica permite que softwares mais complexos sejam divididos em componentes mais simples, e sua funcionalidade plena será representada pela composição de serviços de suas partes. Isto permite que técnicas como “dividir para conquistar” sejam aplicadas no desenvolvimento de software. Além disso, esta característica de modularidade por serviços facilita a possibilidade de reuso e compartilhamento de componentes com outros softwares, diminuindo a quantidade de retrabalho necessário para o desenvolvimento do aplicativo.

2.2.3 - Independência de Plataforma

Como o modelo máquina social pretende ser descritivo em alto nível, ele limita-se na descrição de implementação do software. Portanto, por definição, ele será independente de plataforma, tecnologia e implementação. O modelo pode ser visto como uma nova forma de descrever a arquitetura e comportamento de sistemas de informação.

2.2.4 - Independência de Implementação

O reuso de código entre diferentes máquinas sociais, como definido anteriormente, será baseado no uso dos serviços, independente de como eles tenham sido implementados.

2.3 - Classificação

Uma forma, entre várias, de classificar máquinas sociais é através das interações que a mesma demonstra. Dessa forma, levando em consideração esta característica, podemos definir quatro classes de máquinas sociais:

2.3.1 – Isoladas

Máquinas que não possuem interação com outros softwares. A interface de comunicação (Wrapper Interface) é a estrutura através da qual uma máquina social realiza entradas e saídas de dados e o fornecimento de serviços entre máquinas sociais. Caso um aplicativo não possua nenhum tipo de interação com outros softwares, então este pode ser considerado uma máquina social isolada.

2.3.2 – Consumidoras

Máquinas sociais que fazem uso de serviços de outras máquinas sociais.

2.3.3 – Provedoras

Aplicativos que tem por objetivo fornecer serviços à outras máquinas sociais.

2.3.4 – Híbridas

Máquinas sociais que consomem e fornecem serviços. Em suma, softwares que, para prover o serviço a que se propõem, fazem uso dos serviços de outras máquinas sociais.

Máquinas Sociais que possuem interação com outras máquinas são chamadas Sociáveis. Entre as máquinas sociais sociáveis, as que fornecem seus serviços à outras máquinas sociais são classificadas como provedoras. Já as que fazem uso dos serviços oferecidos por outras máquinas sociais, são ditas consumidoras. Caso uma máquina social seja caracterizada em ambos os casos, esta é classificada como híbrida, devido ao seu comportamento de consumir e prover conteúdo para a rede. Estes casos geralmente são softwares gerados pela combinação de serviços e pelo processamento de informação para prover seus próprios serviços.

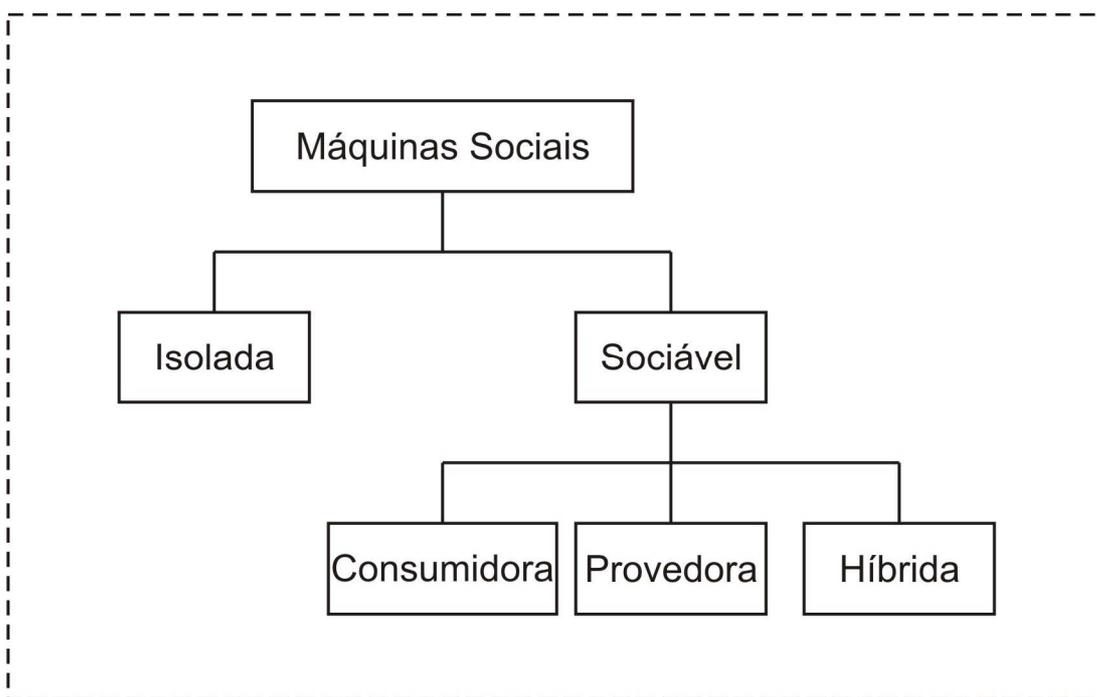


Figura 3: Taxonomia das Máquinas Sociais

3 - Etapas de projeto

O objetivo central deste trabalho é o desenvolvimento de um guia de desenvolvimento para aplicações web seguindo o modelo de máquinas sociais. Para tal, é proposto um modelo de desenvolvimento incremental do projeto do software. Neste, a fase inicial consiste em definir qual a funcionalidade desejada no software, através de uma abordagem de alto nível, não se limitando à detalhes de funcionamento do mesmo. Dessa forma, é possível observar a viabilidade de subdivisão do software em mais de uma máquina social, visando o reuso de componentes disponíveis na rede. Dessa forma, é possível, em tempo de projeto, prever os padrões necessários a ser seguidos durante a fase de desenvolvimento.

Depois de definidos, em alto nível, as funcionalidades da máquina social e os componentes que serão reusados, é possível, portanto definir quais as conexões que serão estabelecidas e suas peculiaridades. Após este passo, fica simples de definir a interface, de forma simplificada, necessária para atender aos requisitos descritos nas etapas anteriores. Com esta definida, basta apenas descrever o restante das propriedades do aplicativo, de acordo com estas características. Com este modelo simplificado construído, um trabalho de detalhamento, ao nível de funcionamento é o suficiente para se ter um modelo do software a ser construído. Este modelo de alto nível estará de acordo com o conceito de máquina social e possuirá suas características e vantagens.

3.1 - Definição do aplicativo

Nesta etapa, são definidas as funcionalidades em alto nível do software a ser desenhado. Com base nesta definição avalia-se a necessidade de fragmentar o software em mais de uma máquina social, numa estratégia de dividir para conquistar. Esta estratégia visa simplificar o desenvolvimento do mesmo, compartilhar componentes e viabilizar o reuso de componentes pré-existentes. A descrição das funcionalidades do software já permite determinar, de forma bastante simplificada, a unidade de processamento, os estados da máquina

social, quais as entradas de dados que são esperadas no sistema e quais as saídas desejadas para atender aos requisitos discriminados.

3.1.1 - Reuso de componentes

Para o desenvolvimento de software nestes moldes, é bastante interessante colocar como prioridade o reuso de componentes, como forma de padronização e diminuição da quantidade de retrabalho necessário para o desenvolvimento do software. Portanto, é fundamental que os componentes de software sejam organizados de forma a facilitar a descoberta e uso efetivo dos mesmos. Técnicas na área de reuso de software e organização de repositórios podem ser encontradas na literatura especializada[10].

3.2 - Definição das conexões

Após a etapa de definição do aplicativo, é possível determinar as dependências entre as diversas máquinas sociais definidas, definindo, portanto, as conexões existentes entre as máquinas sociais. Nesta etapa, estas conexões são apenas uma simples definição de dependência. As propriedades e restrições de cada conexão serão definidas na etapa de detalhamento do projeto. É importante salientar, contudo, que, levando em consideração os componentes escolhidos para reuso, já nesta etapa podem ser descobertas algumas restrições e características da conexão e de algumas máquinas sociais, tendo em vista que os componentes que serão utilizados já possuem restrições de uso.

3.3 - Definição da interface e propriedades

Com os componentes e conexões definidos, bem como as funcionalidades principais do software descritas, torna-se possível desenhar o esboço da interface necessária para atender às características já determinadas. Nesta etapa, uma versão simplificada da interface é gerada, contendo os serviços necessários para o fornecimento das funcionalidades do software. Além disto, esta interface já considera os serviços necessários para devida comunicação com as outras máquinas sociais conectadas.

Além da definição da interface, nesta etapa é possível refinar a definição do conjunto básico de entradas e saídas que o software irá gerar, bem como determinar quais serão as requisições a ser tratadas e respostas a serem geradas pelo software durante o fornecimento do serviço.

3.4 - Detalhamento

A etapa de detalhamento engloba todo o processo incremental de refino do modelo. Este refino é baseado na melhor descrição de cada propriedade visando aproximar-se mais do comportamento desejado da máquina social. Através do documento de casos de uso, e do documento de requisitos, é possível determinar quais ajustes do modelo devem ser feitos, de forma a adaptar a máquina social ao uso pretendido. Além disto, o próprio desenvolvimento do software será responsável por tênues alterações no modelo descrito originalmente.

4 - Estudo de caso

Nesta seção será descrito um conjunto de aplicações para ajudar a definir as características do padrão de máquinas sociais. As aplicações, propositalmente serão desenhadas de forma a explicitar algumas características como reuso de componentes, comunicação interna e externa, além do caráter vinculado à agilidade de desenvolvimento e facilidade na visualização das interações entre os aplicativos.

A escolha dos aplicativos visou simplificar a demonstração dos conceitos e são de funcionalidade simples, para não atrapalhar o entendimento dos conceitos de projeto de máquinas sociais. As aplicações escolhidas foram versões simplificadas de um ERP (gestor financeiro), de um CRM e de uma Loja Virtual, que são conectados entre si.

Como é notório, observando suas aplicações no mercado, estas aplicações são geralmente utilizadas em conjunto, sendo um diferencial que as mesmas compartilhem informações entre si. Isto é bastante facilitado pela característica da composição de máquinas sociais, permitindo que ambas as aplicações comuniquem-se com máquinas compartilhadas.

Os serviços compartilhados pelas aplicações vão desde um simples controle de acesso e de uma manipulação de uma base de dados até serviços mais complexos, como algoritmos para determinar padrões de comportamento dos clientes. Neste ponto, fica claro que existe uma redução significativa da quantidade de retrabalho necessária para a construção das aplicações.

4.1 - Descrição simplificada dos aplicativos

As aplicações escolhidas optam por trazer uma abordagem conectada de sistemas de negócio, levando em consideração o projeto como uma máquina social. Podemos definir, como exemplo para este trabalho, uma rede de máquinas sociais conforme demonstrada a seguir.

4.1.1 - ERP

Conexões: Conexões com outras máquinas sociais trazem agilidade no desenvolvimento e permitem o reuso de código, facilitando a estruturação das aplicações. No caso desta aplicação exemplo, é posto como exemplo simples de sociabilidade a conexão com uma máquina social responsável por realizar o gerenciamento das operações do banco de dados. APIs simples podem ser usadas, como é o caso do serviço SimpleDB da Amazon, que disponibiliza o uso de sua API de forma gratuita para desenvolvimento e pequenos usos. Além disso, como exemplo da possibilidade de diminuição de retrabalho no desenvolvimento de um aplicativo, podemos citar o serviço Xeround, que pode ser conectado sem necessidade de alteração do código do aplicativo.

Além da conexão com o banco de dados, é fundamental que o sistema gerencial se comunique diretamente com a loja virtual, fornecendo o serviço de controle de pedidos e estoque de forma efetiva. Esta conexão diminui a complexidade de construção da loja ao mesmo tempo que melhora a qualidade do serviço de controle ofertado, pois transporta cada serviço específico para a aplicação dedicada.

Entradas e Saídas: Entradas e saídas são modeladas em um padrão bastante estabelecido na indústria: JSON. Desta forma, é possível utilizar bibliotecas prontas de encapsulamento de objetos para JSON, diminuindo o risco de falhas neste ponto e reduzindo a quantidade de trabalho de programação envolvido. Além disso, devido ao reuso desta estrutura, o foco do desenvolvedor permanece concentrado nas características de negócio da aplicação.

Requisições e Respostas: Requisições e respostas são tratadas na interface do software. Cada interface possui um conjunto definido de requisições e respostas. Esta camada pode ser efetivamente utilizada para a programação de *adapters*, para entrar em consonância com o formato utilizado pela máquina social destino. Neste exemplo de desenvolvimento, as requisições e respostas serão seguindo o protocolo http, para as requisições feitas pela front-end web e pela máquina social da Loja Virtual.

Unidade de Processamento: Estrutura responsável por fornecer as regras de negócio desta máquina social. Nesta estrutura encontram-se os algoritmos responsáveis por realizar o controle financeiro e de estoque necessários no fornecimento do aplicativo ERP. Funciona em composição com o serviço de banco de dados, portanto, seu funcionamento é dependente desta máquina social.

Estados: Os estados deste software apenas indicam sua condição de funcionamento: Operacional, Indisponível.

Restrições: As restrições desta máquina social seguem as restrições de uso de sua dependência, o serviço de banco. Estas são limitações no volume, tamanho e frequência de requisições. Além destas, restrições de segurança são necessárias para este serviço, como o acesso registrado e criptografado e limite de requisições.

Interface: Esta máquina social possui duas interfaces de funcionamento. Uma para o usuário final web, com serviços básicos de exibição e modificação de dados, além de visualização de relatórios. A outra interface é a API utilizada pela loja virtual, para cadastrar e requisitar clientes, atualizar estoque e cadastrar pedidos de venda. Na lista de trabalhos futuros está o desenvolvimento de uma conexão com o facebook, como forma de *sign in* alternativo.

4.1.2 - CRM

Conexões: As conexões presentes neste aplicativo são com o software responsável pelo gerenciamento do banco de dados e com o software ERP. O primeiro caso, para coletar informações sobre os clientes que se pretende comunicar, bem como para armazenar o perfil traçado através das informações obtidas no ERP.

Entradas e Saídas: A entrada deste sistema são os dados referentes à um cliente ou grupo de clientes. A Saída do software será uma lista de clientes pertencentes à um mesmo grupo de afinidade, a serem contactados com uma abordagem específica.

Requisições e Respostas: Requisições e respostas serão realizadas na interface web do software e através da conexão estabelecida entre as máquinas sociais.

Unidade de Processamento: Estrutura responsável por organizar o perfil dos clientes e manter o registro dos contatos realizados com o cliente e seus resultados.

Estados: Os estados desta máquina social são apenas relativos ao seu funcionamento: Indisponível, Operacional.

Restrições: Assim como o ERP, as restrições de uso deste software seguem as restrições do banco de dados.

Interface: Inicialmente, apenas interface web, seguindo o documento de requisitos e casos de uso.

4.1.3 - Loja Virtual

Conexões: Assim como o CRM, as dependências desse software são com o banco de dados e com o ERP. O primeiro caso, para armazenar informações específicas da loja, e a conexão com o ERP para gerenciar os dados de clientes, vendas e produtos. Conexão com sistema de pagamento digital.

Entradas e Saídas: Entradas fornecidas pelo usuário ou administrador através da interface web. As saídas deste software serão as modificações geradas no ERP, através de clientes, vendas e produtos cadastrados. Além disto, em uma próxima interação, podem ser definidos relatórios como saída do sistema.

Requisições e Respostas: Apenas requisições de uso através da interface web e de comunicação através da conexão entre as máquinas sociais.

Unidade de Processamento: Estrutura responsável pelo processamento de cada pedido e devido cálculo dos custos. Esta unidade de processamento também é responsável por coordenar o processo de registro dos dados na base de dados.

Estados: Apenas estados indicativos de funcionamento: Operacional, Indisponível.

Restrições: As restrições de uso deste software seguem as restrições do banco de dados e as limitações de uso impostas pelo ERP.

Interface: Apenas interface web simplificada, seguindo o documento de requisitos e casos de uso.

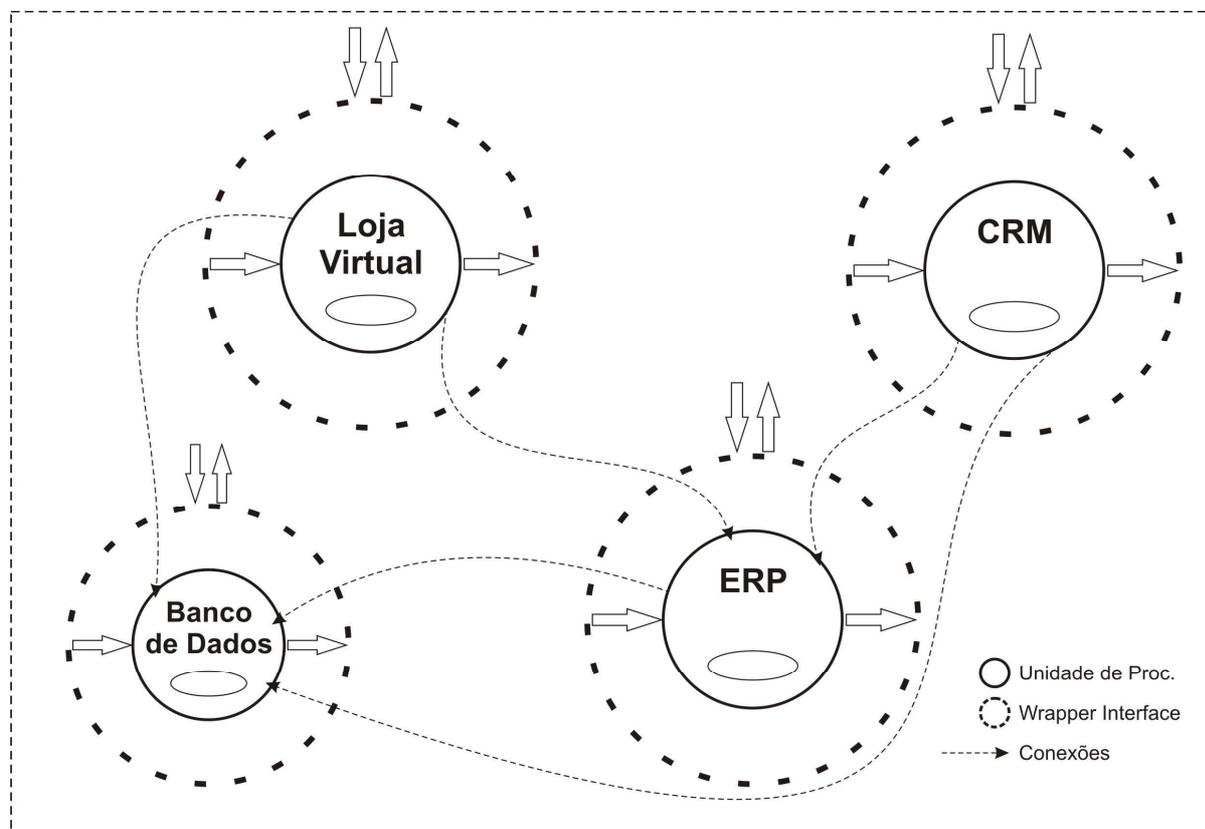


Figura 4: Esquema de relacionamentos simplificado

3.2 - Considerações do estudo de caso

Conforme pode ser observado neste resultado do estudo de caso, boa parte das funcionalidades do software pode ser efetivamente reutilizada em outros aplicativos, através da simples comunicação por uma API. Desta forma, novos serviços podem ser facilmente programados com o auxílio dos softwares preexistentes.

A discriminação de cada ponto descrito no estudo de caso é eficaz em descrever um software e as características fundamentais de seu desenvolvimento. Na descrição deste trabalho, consta apenas o detalhamento inicial dos aplicativos. Conforme descrito no capítulo 2, o detalhamento mais

profundo de cada uma das propriedades apresentadas produz o projeto a ser seguido durante a implementação do software.

Alguns pontos do estudo de caso poderiam ser alterados, para seguir ainda mais ao conceito de web programável e desenvolvimento através da conexão de softwares, porém, a perda na explicação do conceito seria notória e forçou a opção por não fazer uso de aplicações prontas. Todos os aplicativos citados no estudo de caso poderiam ter sido substituídos por versões consolidadas disponíveis na rede como software como serviço, caso do ERP SAP[7] e do CRM Salesforce[6], ou como softwares open-source que necessitariam apenas de pequenas adaptações, como o Magento webstore[8].

4. Conclusão

Durante o desenvolvimento deste trabalho, várias características do modelo de máquinas sociais ficaram mais evidentes. Entre estas características estão pontos fortes e fracos. Entre os pontos bons do conceito está a facilidade que o desenvolvedor passa a possuir do entendimento do software, pois é gerada uma representação clara e completa do comportamento do aplicativo. Além disso, com o detalhamento do modelo, é trivial descrever o funcionamento dos serviços necessários ao funcionamento da máquina social.

O próprio conceito de máquina social inclui um forte embasamento em modularidade e orientação a serviços, acaba possuindo características que favorecem o reuso de software, através da composição de componentes. Esta vantagem é bastante interessante, pois garante que o foco do desenvolvedor seja mantido nas regras de negócio do software que está sendo construído.

Entretanto, esta mesma característica é responsável por alguns problemas tipicamente apresentados por softwares projetados desta maneira. Devido ao caráter distribuído das máquinas sociais, é mais fácil que ocorram falhas devido à problemas de comunicação entre as máquinas ou indisponibilidade das máquinas remotas. Além disso, a perda de eficiência pode ser um sério entrave, para algumas aplicações de maior porte ou que necessitem de alto desempenho. Outro ponto fraco verificável neste modelo é a grande dependência dos serviços de terceiros. É fácil observar que a estabilidade do sistema principal depende totalmente da disponibilidade dos sistemas que o compõem.

O modelo de máquinas sociais é um bom conceito para uma nova compreensão de engenharia de software, levando em consideração o reuso de software a composição de componentes como ponto prioritário no desenvolvimento de novas aplicações. Esta mudança de paradigma trás uma facilidade e uma agilidade maior de desenvolvimento de novas aplicações, pois é capaz de oferecer a possibilidade de que novos serviços sejam desenvolvidos à partir de serviços bem estabelecidos. A oportunidade que este modelo tem é o aumento de sua comunidade de usuários e definição real de uma padronização

do desenvolvimento web, consolidando um cenário favorável ao desenvolvimento através de máquinas sociais.

Entretanto, é crescente a quantidade de ameaças presentes na rede, estas vão desde falhas de segurança na infraestrutura da rede até APIs maliciosas fornecidas por terceiros. Problemas como estes podem comprometer a confiabilidade do modelo, fazendo com que stakeholders prefiram manter-se no modelo tradicional de desenvolvimento, devido à consolidação e garantias mais claras de segurança.

Além disso, por definição, uma máquina social corre o risco de ser superposta por terceiros que fazem uso de sua interface. Um exemplo a ser citado é o twitter cuja principal forma de acesso, hoje em dia, é através de softwares de terceiros, como o hootsuite e o tweetdeck. Esta perda de usuários em sua interface direta acaba inviabilizando o estabelecimento de modelos de negócio eficazes, prejudicando a rentabilidade da operação.

4.1 Trabalhos Futuros

Trabalhos futuros nesta área envolvem diretamente a especificação formal de soluções para os entraves específicos enfrentados pelas máquinas sociais ou o desenvolvimento de formas de aperfeiçoar a efetividade deste modelo.

O desenvolvimento de uma linguagem específica para descrição das máquinas sociais. Esta linguagem, SMADL, é proposta no artigo original do conceito [11]. O desenvolvimento de uma ferramenta deste tipo seria fundamental para a padronização do projeto.

Outra proposta de trabalho futuro é o desenvolvimento de um mecanismo de conversão de um modelo de máquina social para sua possível implementação prática da mesma.

Definição de um modelo de padronização de interfaces. O padrão REST vem se desenvolvendo nos últimos anos com este objetivo, porém, o trabalho relacionado à esta linha de pesquisa é bastante interessante para o desenvolvimento da engenharia de software voltada para web.

Definição de uma linguagem, variante da SMADL para descrever relacionamentos entre as máquinas sociais, bem como as restrições e características destas. Desenvolvimento de técnicas que auxiliem no desenvolvimento das interfaces com base no descritivo destas conexões.

Referências

- [1] BARNETT, T. The Dawn of the Zettabyte Era. Disponível em: <http://bit.ly/sBk4MM>". Acessado em: Novembro, 2011.
- [2] AMMIRATI, S. Data Deluge. Disponível em: <http://rww.to/uxvGGt>". Acessado em: Novembro, 2011.
- [3] Ofcom: Communications Market Report. Disponível em: <http://bit.ly/tfUhui>. Acessado em: Outubro, 2011.
- [4] Xeround: Cloud Database. Disponível em: <http://xeround.com>. Acessado em: Novembro, 2011.
- [5] FINLEY, C. Cloud-Based Services. Disponível em: <http://rww.to/rP7JPG>. Acessado em: Outubro, 2011.
- [6] Salesforce: cloud CRM. Disponível em: <http://www.salesforce.com>. Acessado em: Março, 2011.
- [7] SAP: business management software solutions. Disponível em: <http://www.sap.com>. Acessado em: Novembro, 2011.
- [8] Magento: ecommerce software for growth. Disponível em: magentocommerce.com. Acessado em: Novembro, 2011.
- [9] WILLIAMS, A. How the clouds redefines e-commerce. Disponível em: <http://rww.to/tsTzoL>. Acessado em: Outubro, 2011.
- [10] MEIRA, S. R. L.; et al. Component Reuse in Software Engineering. Recife: C.E.S.A.R E-books, 2007. 210p.
- [11] MEIRA, S. R. L.; et al. The Emerging Web of Social Machines. Recife: CIn-UFPE, 2007. 19p.
- [12] Programmable Web: the web as platform. Disponível em: www.programmableweb.com/. Acessado em: Novembro, 2011.
- [13] ROUSH, W. Social Machines: Computing Means Connecting. MIT Technology Review, Agosto, 2005.
- [14] CORMODE, G. KRISHNAMUTHY, B. Key Differences Between web1.0 and web2.0. AT&T Labs, Fevereiro, 2008.
- [15] Wikipedia. Service Oriented Architecture. Disponível em: http://en.wikipedia.org/wiki/Service-oriented_architecture. Acessado em: Junho 2011.

- [16] Wikipedia. Representational State Transfer. Disponível em: http://en.wikipedia.org/wiki/Representational_state_transfer. Acessado em: Junho, 2011.
- [17] KRAUS, A. Model Driven Engineering for Web Applications. 408 p.
- [18] OLSEN, D. R. Putting the Web Services Specifications to REST. 35 p.