

Um Processo Guiado para Levantamento e Modelagem de Requisitos de Aplicações Web Baseado em Objetivos e Casos de Uso

Dênis Zaniro

Sandra Fabbri

UFSCar – Universidade Federal de São Carlos

{denis_zaniro, sfabbri}@dc.ufscar.br

Abstract

Background: *Web application development differs from the development of traditional software in several significant ways. Therefore, Requirements Engineering for Web (WebRE) needs to be fitted in a proper way to deal with some distinctive aspects that must be addressed in the development. Aiming to be a good communication channel between stakeholders and developers, WebRE methods should be lightweight, but to be useful to developers and to allow metric application, they should have expressive modeling constructions. Aim:* *Based on this context, this paper presents a part of the method Web-SEMP – Web System Elicitation, Modeling and Planning, proposed to help in these three activities of a Web application development process. Method:* *The most quoted WebRE methods in the literature were analyzed aiming to reuse and to extend them to compose the proposal here presented. Results:* *The method Web-SEMP was applied in the development of a real web application and the results showed that the method gives support for a more systematic WebRE process. Conclusion:* *The results provide insights that the method can help to improve the development process quality of Web Applications.*

1. Introdução

Uma das tarefas mais difíceis enfrentadas por um engenheiro de software é compreender as reais necessidades de clientes e usuários. Nesse contexto, o objetivo da Engenharia de Requisitos (ER) é auxiliar os engenheiros de software a compreenderem o problema a ser revolido, fornecendo uma base sólida para o projeto e a construção.

A ER, como todas as outras atividades de engenharia de software, precisa ser adaptada às necessidades do processo, do projeto, do produto e da equipe que está desempenhando o trabalho [1]. No contexto de aplicações baseadas na Web, a ER se

depara com novas necessidades que acomodem as características distintas desse tipo de aplicação.

Um dos principais desafios enfrentados pela ER para a Web está relacionado à existência de especificações iniciais insuficientes acerca dos requisitos, uma vez que, na grande maioria dos casos, o desenvolvimento de uma aplicação Web é iniciado a partir de requisitos vagos ou mal-estruturados. Outro grande desafio é a necessidade de gerenciamento da complexidade de requisitos altamente voláteis, refletindo as exigências dinâmicas do mercado [2]. Mudanças constantes nos requisitos e diversas restrições são características comuns de projetos Web. Assim, é fundamental que haja condições de gerenciar as interdependências entre requisitos, e entre requisitos e outros artefatos, a fim de avaliar efetivamente o impacto de possíveis mudanças durante o desenvolvimento [4].

O terceiro desafio diz respeito à necessidade de liberação de versões funcionais da aplicação Web logo no início do desenvolvimento [5]. Dessa forma, os engenheiros Web precisam usar métodos de planejamento, análise, projeto, implementação e teste que tenham sido adaptados a prazos curtos, que podem variar de dias a algumas semanas. Outro problema está ligado ao gerenciamento de uma grande variedade de grupos de usuários cujos objetivos precisam ser satisfeitos pela aplicação a ser desenvolvida. Muitos *stakeholders* ainda são desconhecidos durante as atividades de ER e, além disso, é necessário, muitas vezes, levar em consideração requisitos de internautas e visitantes ocasionais. Diante dessas circunstâncias, é fundamental encontrar representantes adequados que possam fornecer requisitos realísticos.

Embora a maioria desses problemas correspondam ao processo de ER de forma geral, no contexto de aplicações Web, tais dificuldades são encontradas com maior frequência, devido às restrições impostas pelas características específicas de tais aplicações.

Outros fatores que devem ser tratados pela ER para a Web incluem aspectos de qualidade tais como

segurança, alta disponibilidade, usabilidade, padrões de navegação consistentes, estética, previsibilidade de informação, escalabilidade, desempenho e qualidade do conteúdo, no que diz respeito ao seu nível de atualização, consistência e confiabilidade.

Todas as características mencionadas acima permitem distinguir aplicações Web de muitas outras categorias de software. Embora haja diversas taxonomias propostas na literatura para classificar as aplicações Web em si, podem ser identificadas duas amplas categorias, de acordo com o seu estilo de interação [4] [6]:

- *Aplicações de hipermídia*: são as aplicações de hipertexto e multimídia formadas, basicamente, por um conjunto de páginas e *links*. Cada página é constituída por elementos de conteúdo, os quais são apresentados aos usuários, em resposta a alguma solicitação. Nessa categoria, os usuários navegam pelas páginas, exploram diversas categorias de conteúdo por meio de botões ou menus e selecionam *links* que dão acesso a outras páginas ou aplicações.
- *Aplicações transacionais*: são as aplicações que têm por objetivo prover maior interatividade, permitindo ao usuário ativar operações, enviar e atualizar dados, receber notificações e mensagens de sistema e efetuar transações, da mesma forma que nos sistemas de informação convencionais.

Nos dias de hoje, as aplicações Web complexas, no entanto, combinam esses dois paradigmas de interação [6]. Mesmo sites Web pequenos, constituídos por algumas operações simples e um conjunto de arquivos de hipertexto ligados que apresentam informação, precisam ser gerenciados ou, ainda, integrados com bancos de dados corporativos e aplicações de negócio. Sites Web de *e-commerce*, *e-business* e *e-banking* são alguns exemplos típicos dessa combinação de estilos de interação.

A necessidade premente de atender os atributos específicos encontrados na vasta maioria das aplicações Web tem levado à criação de estratégias, táticas e métodos que sejam conduzidos durante as atividades da ER, habilitando, assim, um engenheiro de software a entender, documentar e validar requisitos de aplicações e sistemas baseados na Web. Os métodos propostos na literatura para dar suporte às atividades de ER para a Web são aplicados de maneira isolada e, além disso, a maioria deles não integra algumas das práticas e princípios fundamentais da engenharia de software.

Assim, algumas dessas propostas falham em satisfazer todos os requisitos do usuário, quando um dos tipos de aplicação Web – hipermídia, transacional ou a combinação de ambos – não é contemplado pelo método em questão.

Outro problema focado durante a avaliação de métodos de ER para a Web diz respeito à ausência de diretrizes que possibilitem um mecanismo de transição efetivo entre requisitos e artefatos de análise e projeto. Diante do imediatismo que as aplicações Web, em geral, exigem para serem colocadas no mercado, é fundamental que a distância entre especificações de requisitos e modelos subsequentes seja reduzida.

Métodos de ER para a Web devem levar em consideração que muitos analistas Web e *stakeholders* do projeto têm pouco ou nenhum conhecimento técnico, portanto tais métodos não podem exigir muito esforço de treinamento para que sejam adotados e efetivamente integrados em práticas atuais [5]. Os modelos e notações propostas pelo método devem ser leves e intuitivas para que sirvam como um canal de comunicação entre todos os *stakeholders* envolvidos, mas, ao mesmo tempo, suas representações devem ser completas de forma a permitir a aplicação de métricas de software durante a atividade subsequente de planejamento.

Diante desse contexto, apresenta-se, neste artigo, o método Web-SEMP – Web System Elicitation, Modeling and Planning. O método é composto por três fases: I) Elicitação de Requisitos e Planejamento Inicial, II) Aplicação da Estratégia de Mapeamento e III) Aplicação do Modelo de Métrica. Neste artigo, serão tratadas apenas as duas primeiras fases, que dão suporte ao processo de ER para a Web.

Para as atividades de elicitação e especificação de requisitos (Fase I), o método Web-SEMP adota uma abordagem orientada a objetivos, denominada AWARE – Analysis of Web Application Requirements – [6]. A partir da análise de requisitos, são elaborados casos de uso funcionais e navegacionais, e o método ainda define um conjunto de diretrizes para a construção efetiva de modelos de estrutura de hipertexto de uma aplicação Web (Fase II). Para a construção de modelos de estrutura de hipertexto, o método Web-SEMP adota e estende as notações conceituais da WebML – Web Modeling Language – [28].

O artigo está organizado da seguinte forma: na Seção 2, são discutidos alguns trabalhos relacionados ao método aqui proposto. Na Seção 3 apresenta-se, em detalhes, cada fase do método Web-SEMP. Na Seção 4 é descrito um exemplo real de aplicação do método proposto, e, finalmente, na Seção 5 são discutidas algumas conclusões e trabalhos futuros.

2. Trabalhos Relacionados

O aumento da complexidade das aplicações Web e a necessidade cada vez maior de se obterem produtos

finais de alta qualidade são os principais fatores que justificam os esforços de muitos pesquisadores buscando estabelecer estratégias que dêem suporte efetivo à fase de ER para a Web. Foi feita uma Revisão Sistemática (RS) com o intuito de identificar e analisar alguns dos principais métodos usados na ER em geral e, dentre os encontrados, destacam-se os métodos comentados a seguir. Tais métodos têm sido também aplicados no contexto Web e apresentam alguma similaridade com o modelo AWARE, adotado pelo método Web-SEMP aqui proposto, o qual é baseado na identificação de objetivos.

Considerada como a primeira abordagem orientada a objetivos proposta na literatura, KAOS [11] é um método formal, adequado para a definição e análise de requisitos de sistemas de segurança crítica. O método é baseado na premissa de que para construir um software em um ambiente organizacional, é necessário analisar e modelar explicitamente esse ambiente, em termos de atores, seus objetivos e dependências. Uma avaliação sistemática da efetividade do método KAOS assim como de uma ferramenta associada pode ser encontrada em [12].

Com base no método KAOS, o *framework i** [13] introduz o conceito de agente organizacional na ER e propõe um conjunto de operadores para modelar dependências entre agentes com relação aos seus objetivos, tarefas e recursos a serem fornecidos.

Já o método Tropos [14] compreende todas as atividades de análise e projeto durante o processo de desenvolvimento de software. O objetivo é construir um modelo tanto do sistema a ser desenvolvido quanto do seu ambiente, que deve ser refinado e estendido, de forma incremental, por meio de 4 fases: requisitos iniciais, requisitos bem elaborados, projeto arquitetural e projeto detalhado. O método adota os mesmos conceitos oferecidos pelo *framework i** como suporte a todas as fases mencionadas.

GBRAM [15] é um método que combina a abordagem orientada a objetivos com técnicas baseadas em cenário. Além disso, o método estabelece um conjunto de heurísticas para identificar, analisar e refinar objetivos em requisitos. Assim como Tropos, ele é voltado para sistemas de informação convencionais e, portanto, não contempla os aspectos de navegação e de composição, inerentes aos artefatos de hipermídia como são as aplicações Web.

O método Vord foi estendido em [16] para atender às necessidades específicas de definição de requisitos para aplicações Web de *e-business*. Esse método tem como objetivo elicitar e formular requisitos relacionados a estratégias e processos de negócio de acordo com diferentes pontos de vista, com ênfase nas operações e serviços que serão automatizados. Outros métodos de ER que incorporam estratégias e operações

de negócios organizacionais para aplicações Web são encontrados em [17], [18] e [19].

MODFM [2] [3] é um método de prototipação rápida orientado a modelo, proposto para auxiliar nas atividades de elicitação e validação de requisitos para aplicações Web. O método aplica conceitos da ER tais como a técnica de casos de uso, análise estruturada, técnica de prototipação rápida e engenharia de requisitos baseada em arquitetura. Com base nesses conceitos e em uma proposta de arquitetura de software formalizada, o método estabelece um algoritmo que deve receber como entrada uma especificação funcional de alto-nível do sistema e gerar como saída um protótipo para a aprovação do cliente. A principal desvantagem do método MODFM é que sua efetividade depende fortemente da tecnologia utilizada no desenvolvimento. Além disso, o método fornece suporte apenas para aplicações Web estritamente transacionais.

DSL [20] é uma linguagem formal de especificação de requisitos para a Web, baseada no cálculo de predicado de primeira ordem. A especificação DSL é constituída de predicados e definições, divididos em 3 tipos principais para expressar a noção de tempo: estado, recurso e transição. O primeiro tipo é usado para representar eventos que podem ocorrer na aplicação (por exemplo, o clique de um botão). Já o segundo tipo normalmente está relacionado a alguma entidade física na página (por exemplo, um menu), e o terceiro tipo refere-se a tarefas realizadas pela própria aplicação (por exemplo, envio de alguma notificação ao usuário). Como em outros métodos formais, a grande limitação da DSL é a dificuldade de entendimento do formalismo matemático utilizado, conforme as especificações se tornam maiores e mais ricas. Um estudo de caso envolvendo esse método pode ser encontrado em [21].

A seguir, serão descritas as técnicas adotadas pelo método Web-SEMP, assim como a motivação para a adoção de cada uma dessas técnicas.

2.1. O Modelo AWARE

O modelo AWARE, conforme já citado, adota uma abordagem orientada a objetivos para identificar e especificar requisitos de hipermídia de aplicações Web. A abordagem orientada a objetivos é especialmente adequada para o ambiente Web, uma vez que permite tratar objetivos vagos, imprecisos ou mal-definidos, refletindo necessidades de todos os *stakeholders* envolvidos no projeto [22]. Essa é a principal razão para a adoção desse modelo no método Web-SEMP.

AWARE é considerado como uma extensão do *framework i**, na medida que permite organizar

requisitos de acordo com uma taxonomia proposta para auxiliar na atividade de projeto. A taxonomia de requisitos proposta compreende as seguintes dimensões [6]:

- *Conteúdo*: são requisitos que se referem ao conjunto de idéias e mensagens que o site Web comunica aos seus usuários.
- *Estrutura de Conteúdo*: são requisitos que têm como objetivo indicar como as partes de um objeto de conteúdo identificado podem ser estruturadas.
- *Apresentação*: são requisitos que estão relacionados às estratégias de comunicação visual para apresentar conteúdo, opções de navegação e operações ao usuário.
- *Interação*: são requisitos que descrevem os estilos de interação disponíveis ao usuário e estão relacionados a recursos capazes de aumentar o interesse e envolvimento do usuário por algum assunto ou conteúdo oferecido.
- *Navegação*: são requisitos que indicam conexões entre partes de informação, permitindo que o usuário navegue de um trecho de conteúdo a outro semanticamente relacionado.
- *Caminhos de Acesso*: são requisitos que indicam todas as formas disponíveis ao usuário para chegar ao conteúdo necessário.
- *Operação de Usuário*: são requisitos que se referem a operações visíveis aos usuários, os quais podem ativá-las para completar algumas tarefas, interagindo com a aplicação.
- *Operação de Sistema*: são requisitos que indicam operações que não são diretamente visíveis aos usuários, mas são obrigatórias para construir as operações de usuário.

As notações oferecidas pelo modelo AWARE incluem *stakeholders*, objetivos, tarefas e requisitos de hipermídia. Cada *stakeholder* possui um conjunto de objetivos que podem ser decompostos em sub-objetivos e tarefas, e transformados em requisitos. Todos esses conceitos são representados em um diagrama hierárquico, também referenciado como diagrama GTR (*Goal-Task-Requirement*) [32]. Além dessas notações, ainda é possível representar valores de prioridade (quantitativos ou qualitativos) associados tanto a *stakeholders* quanto a seus objetivos.

A taxonomia de requisitos proposta também pode ajudar a definir e organizar requisitos não-funcionais de uma maneira coerente. Assim, objetivos cruciais relacionados a aspectos de qualidade de aplicações Web são identificados e, conseqüentemente, refinados em requisitos não-funcionais.

Trechos de exemplos de aplicação do modelo AWARE na indústria podem ser encontrados em [6] e [7], e experiências e lições aprendidas com a aplicação

do modelo em outro estudo de caso real podem ser observadas em [8] e [9].

2.2. Modelo de Casos de Uso

O conceito de caso de uso foi proposto por Jacobson et al. [23] em um modelo de processo denominado *Objectory*. O modelo de casos de uso pode ser aplicado durante a fase de ER como uma técnica poderosa de modelagem de requisitos.

Ao contrário da abordagem orientada a objetivos, o modelo de casos de uso é considerado uma abordagem orientada a tarefas [22] e, como tal, pode ser utilizado de forma complementar à abordagem orientada a objetivos para capturar apenas interações entre os usuários finais e o sistema.

Como o modelo de casos de uso é bastante conhecido e bastante utilizado na prática, no contexto deste trabalho, o importante de ser ressaltado é que para tratar as peculiaridades de aplicações Web, foi proposta, na literatura, a divisão do modelo de casos de uso em duas categorias: casos de uso funcionais e casos de uso navegacionais [25] [26]. Em [25], a proposta é criar dois modelos distintos, e em [26] a proposta é criar um único modelo de casos de uso, que usa o estereótipo da UML <<navigation>> para denotar a diferença entre os casos de uso funcionais e aqueles casos de uso específicos de hipertexto. Esta última alternativa é a adotada neste artigo.

Nessa abordagem, os casos de uso funcionais representam funcionalidades e serviços providos pela aplicação, da mesma forma que em sistemas de informação convencionais. Já os casos de uso navegacionais descrevem todos os caminhos que podem ser percorridos pelo usuário para alcançar conteúdos específicos na aplicação Web [26]. Esses casos de uso são apoiados pelo modelo de hipertexto da aplicação.

2.3. A linguagem de modelagem WebML e a extensão proposta

Dentre os principais métodos de modelagem conceitual propostos na literatura nos últimos anos – OOHDM [27], W2000 [25], WebML [28], UWE [26] e WAE [30] – WebML, de acordo com [31], é o mais promissor, uma vez que permite especificar aplicações Web fornecendo um conjunto de notações visuais para modelar a estrutura de conteúdo e aspectos navegacionais de uma aplicação. Essa é a principal razão para o sucesso de WebML, segundo [31], e foi o que motivou a sua adoção no método Web-SEMP.

A especificação de uma aplicação Web em WebML consiste de duas dimensões ortogonais: modelo

estrutural (modelo de dados) e modelo de hipertexto constituído pelos modelos de navegação e de composição [28]. O modelo estrutural expressa entidades e relacionamentos, e o modelo de hipertexto especifica visões, áreas, páginas e unidades de conteúdo. Uma visão é um hipertexto definido para satisfazer um conjunto específico de requisitos. Assim, várias visões podem ser identificadas a partir do mesmo modelo de dados.

Uma visão é composta de áreas, que são as principais seções do hipertexto e compreendem recursivamente outras sub-áreas ou páginas. Páginas são recipientes de informações entregues ao usuário e são constituídas por unidades de conteúdo. Essas, por sua vez, são elementos atômicos de publicação de informação extraídos do modelo de dados. A Tabela 1 mostra as unidades de conteúdo básicas da WebML.

Tabela 1. Unidades de conteúdo da WebML.

Unidade	Notação Visual	Descrição
Data Unit	 Entity [Selector]	Essa unidade publica informações acerca de uma única instância de uma entidade.
MultiData Unit	 Entity [Selector]	Essa unidade publica informações sobre um conjunto de instâncias de uma entidade.
Index Unit	 Entity [Selector]	Essa unidade publica uma lista de atributos de um dado conjunto de instâncias de uma entidade.
Scroller Unit	 Entity [Selector]	Essa unidade fornece comandos para navegar por objetos em uma lista (por exemplo, a seqüência de todas as instâncias de uma dada entidade).
Entry Unit		Essa unidade publica um formulário com vários campos para coletar dados de entrada do usuário.

Unidades de conteúdo e páginas são interconectadas por *links* para constituir visões. *Links* podem ser não-contextuais, quando conectam páginas semanticamente independentes ou contextuais, quando conectam duas unidades de conteúdo, carregando alguma informação que será usada pela unidade de destino.

Neste trabalho foi feita uma extensão da WebML, fornecendo um conjunto de notações visuais ajustadas

para modelar elementos de conteúdo estático e de apresentação. Essa extensão foi necessária por dois motivos: primeiro, para permitir que itens importantes para uma aplicação Web, que fizessem parte dos requisitos dos usuários, pudessem ficar explicitamente declarados no modelo; segundo, que esses itens podem ter um grande impacto no tempo de desenvolvimento e esse fato é essencial para a Fase III do método Web-SEMP, a qual não é apresentada neste artigo.

Os elementos de conteúdo propostos incluem a unidade textual – *text unit* – e unidades não-textuais – *image unit*, *multiimage unit* e *animation unit* –, conforme descrito na Tabela 2. A *text unit* é usada apenas estaticamente, enquanto que as outras unidades podem ser usadas tanto estaticamente quanto em conjunto com outras unidades de conteúdo.

Tabela 2. Unidades de conteúdo estendidas.

Unidade	Notação Visual	Descrição
Text Unit		Essa unidade publica dados estáticos sobre um tópico, idéia ou mensagem acerca de um tema.
Image Unit		Essa unidade publica uma imagem – gráficos, ícones, figuras, fotos, logomarcas, etc. sobre um tema.
MultiImage Unit		Essa unidade publica um conjunto de imagens sobre um tema.
Animation Unit		Essa unidade publica uma animação Gif ou Flash para transmitir alguma mensagem ou destacar uma idéia.

3. O Método Web-SEMP

O método Web-SEMP, aqui proposto, dá apoio às atividades de elicitação e modelagem de requisitos no contexto da ER para Web. Considerando-se o processo de desenvolvimento de uma aplicação Web discutido em [1], [5] e [10], o método cobre as fases de Análise de Requisitos e Projeto de Requisitos, conforme mostra a Figura 1. Considera-se, como em outros processos de desenvolvimento, que essas fases sejam executadas de maneira iterativa e incremental.

A Fase I do método Web-SEMP corresponde às atividades de concepção, levantamento e especificação de requisitos. Para essas atividades, o método proposto adota o modelo AWARE (1), comentado na Seção 2.1, do qual é utilizado o modelo GTR que descreve os objetivos, as tarefas e os requisitos de todos os *stakeholders*. Além disso, na atividade de Planejamento Inicial (2), os analistas e engenheiros da

Web devem prever certos tipos de tarefas não-técnicas, como busca por conteúdos, que serão conduzidas

posteriormente, cujo momento melhor para identificá-las é esse.

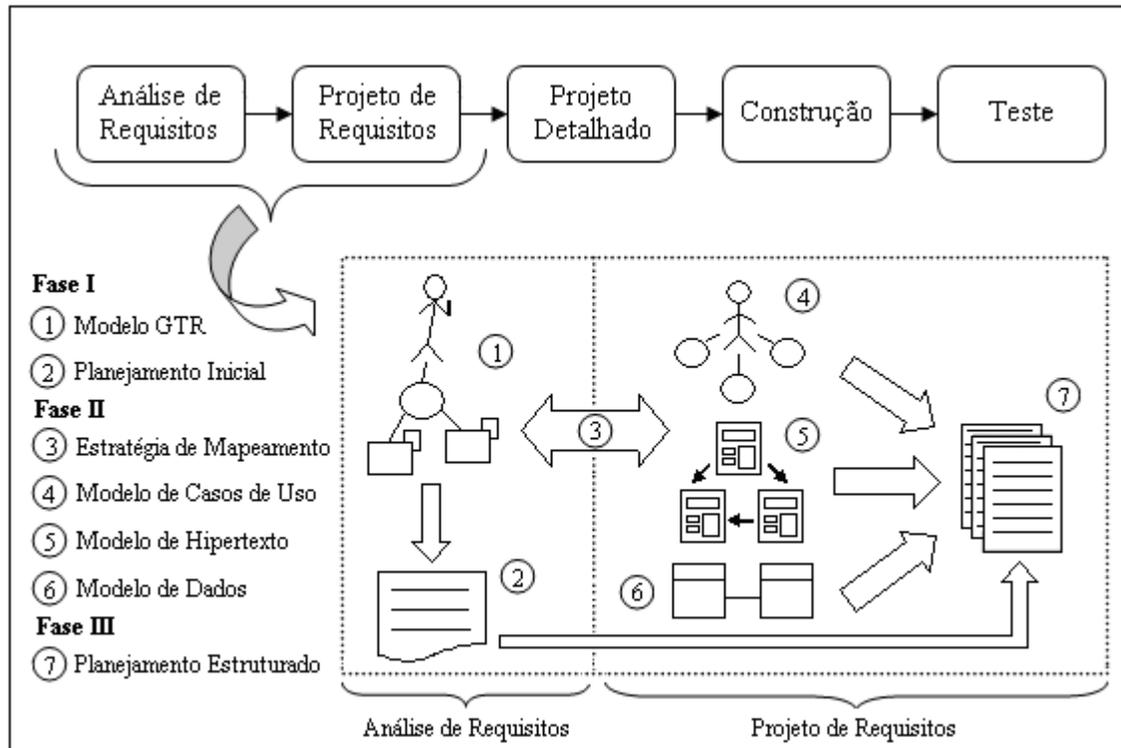


Figura 1. Descrição do Método Web-SEMP.

Devido à natureza abstrata de objetivos e requisitos [10], a Fase II do método Web-SEMP estabelece uma estratégia de mapeamento (3) que permite derivar soluções de projeto de alto-nível focando em aspectos cruciais da aplicação.

A estratégia de mapeamento possibilita construir modelos de casos de uso navegacionais e funcionais (4), modelos de estrutura de hipertexto (5), e identificar entidades e relacionamentos mais relevantes, compondo o modelo de dados (6). A Fase III do método Web-SEMP está relacionada à aplicação do modelo de métrica (7) aos modelos conceituais elaborados na fase anterior, e não será tratada neste artigo.

A seguir, as duas primeiras fases do método Web-SEMP são descritas em detalhes.

3.1. Fase I – Elicitação de Requisitos

A Fase I do método Web-SEMP é composta por duas etapas: etapa de elicitação de requisitos e etapa de planejamento inicial. Neste artigo, como as questões de planejamento não serão abordadas, será descrita apenas a etapa de elicitação de requisitos, que é dividida em dois passos:

- **Passo 1:** identificar cada uma das visões da aplicação Web a ser desenvolvida, de acordo com as categorias de público-alvo. Assim, é possível determinar a necessidade de modelar uma visão pública e visões privadas, cujo conteúdo é específico para cada classe de usuários finais.
- **Passo 2:** para cada visão identificada no passo anterior, aplicam-se os seguintes sub-passos:
 - **2.1:** identificar todos os *stakeholders* que possuem objetivos em relação a uma dada visão.
 - **2.2:** para cada *stakeholder* identificado, elaborar o diagrama GTR proposto pelo modelo AWARE. Durante a construção dos diagramas GTR, é importante, muitas vezes, atribuir prioridades a objetivos, *stakeholders* e requisitos.

3.2. Fase II – Aplicação da Estratégia de Mapeamento

A Fase II do método Web-SEMP é composta por três etapas: identificação de aspectos de composição, identificação de aspectos de navegação e identificação de aspectos de funcionalidade, as quais devem ser realizadas para cada visão identificada na fase anterior.

Assim, adota-se o princípio fundamental da separação de interesses, de maneira que cada etapa

represente uma dimensão diferente da aplicação. A separação de interesses permite que analistas dêem foco em dimensões cruciais de projeto, de acordo com as características da aplicação Web a ser desenvolvida. Esse princípio ainda permite inspecionar os diagramas GTR e avaliar a sua completeza, influenciando, assim, na usabilidade e na qualidade final da aplicação.

Etapa 1 – Identificação de Aspectos de Composição – tem o objetivo de compreender os requisitos relacionados a conteúdo nos diagramas GTR de uma dada visão.

O artefato de modelagem gerado na Etapa 1 é o sub-modelo de composição da WebML estendida, isto é, nessa etapa são identificadas áreas, páginas e unidades de conteúdo que constituem cada uma dessas páginas.

A Etapa 1 deve ser executada seguindo-se 7 passos:

- **Passo 1:** numerar todos os requisitos de hiperlinks que constam nos diagramas GTR para uma dada visão.
- **Passo 2:** identificar os temas nos diagramas GTR para uma dada visão, analisando-se requisitos relacionados a conteúdo, ou seja, requisitos que possuem a notação “C” ou “S”. Temas são os principais tópicos cujas informações devem ser mantidas pela aplicação Web. O formulário TRS – Tema x Requisito x Stakeholder – deve ser preenchido com essas três informações e mais a informação se o tema é Dinâmico (informações que possuem várias propriedades ou atributos e devem constar no modelo de dados, representando entidades) ou Estático (idéias ou mensagens transmitidas pela aplicação Web que dificilmente se modificarão ao longo do tempo).
- **Passo 3:** relacionar junto a cada tema do formulário TRS os requisitos de conteúdo (do tipo “C” ou “S”) que estão associados ao mesmo tema.

Para cada tema listado no formulário TRS repetir os passos 4 a 7:

- **Passo 4:** mapear requisitos de conteúdo referentes a um tema em unidades de dados definidas na WebML estendida. Caso o tema seja estático, requisitos de conteúdo serão mapeados em *text units*. Caso o tema seja dinâmico, requisitos de conteúdo serão mapeados em *data units* ou em *multidata units*. Em qualquer um dos casos, uma página principal deve ser criada para o tema.
- **Passo 5:** analisar requisitos de estrutura de conteúdo referentes a um tema. É necessário verificar como a mensagem referenciada pelo requisito deve ser apresentada ao usuário. Requisitos desse tipo podem ser satisfeitos aplicando-se as mesmas regras do Passo 4 (para conteúdo textual) ou as mesmas regras do Passo 6 (para conteúdo não-textual).
- **Passo 6:** analisar todos os requisitos de apresentação mapeando-os em *image units*, *multiimage units*, ou

animation units. Essa regra se aplica para requisitos referentes a temas estáticos e a temas dinâmicos. Requisitos indicando características gerais de apresentação (leiaute, cores, estilos de fonte, etc.) devem servir de entrada para a Fase de Projeto Detalhado, no processo de desenvolvimento, a qual não é abordada pelo Web-SEMP.

- **Passo 7:** analisar todos os requisitos de interação, os quais podem expressar a necessidade de criação de animações mais complexas – *animation units* –, ou operações que exigem recursos específicos.

Etapa 2 – Identificação de Aspectos de Navegação – tem o objetivo de analisar e identificar todos os requisitos de navegação e de caminhos de acesso ao conteúdo nos diagramas GTR, para cada uma das visões, gerando um único modelo de hipertexto, formado pelos sub-modelos de composição, aos quais se adicionam os aspectos de navegação.

Essa etapa recebe como entrada os diagramas GTR e o sub-modelo de composição elaborado na etapa anterior. Nesse sub-modelo, são introduzidos os relacionamentos semânticos entre temas assim como as estruturas de acesso disponíveis ao usuário. Isso corresponde a identificar *links* não-contextuais (entre páginas), *links* contextuais (entre unidades) e unidades de acesso tais como *index units* e *scroller units*.

A Etapa 2 deve ser executada por meio de 3 passos:

- **Passo 1:** identificar todos os requisitos de navegação nos diagramas GTR. Temas devem ser relacionados por meio de *links* contextuais ou não-contextuais, seguindo-se as regras de validade do modelo de hipertexto, definidas em [28]. Requisitos de navegação devem ser mapeados em *links* não-contextuais – quando há, pelo menos, um tema estático envolvido no relacionamento –, ou em *links* contextuais – quando os dois temas envolvidos são dinâmicos. Nesse último caso, o requisito de navegação serve como entrada para se identificar relacionamentos no modelo de dados da aplicação.
- **Passo 2:** identificar todos os requisitos de caminhos de acesso ao conteúdo, os quais devem ser mapeados em estruturas de acesso apontando para o tema referenciado pelo requisito. Essas estruturas de acesso incluem *index units*, *scroller units* ou ainda *entry units*, ligadas a outras unidades de conteúdo.
- **Passo 3:** identificar casos de uso navegacionais. A especificação de temas, assim como relacionamentos semânticos entre eles, auxilia efetivamente na identificação de casos de uso navegacionais.

Etapa 3 – Identificação de Aspectos de Funcionalidade – tem como objetivo analisar e identificar todos os requisitos de operação de usuário e requisitos de operação de sistema. Assim, essa etapa permite identificar casos de uso funcionais, bem como relacionamentos entre eles.

4. Exemplo de Aplicação do Método Web-SEMP

Neste exemplo o método Web-SEMP foi aplicado durante a análise de requisitos de uma aplicação Web para um fornecedor de produtos para combate a formigas, denominado Mirex. A Mirex é uma empresa brasileira voltada ao desenvolvimento de tecnologias para o controle de formigas cortadeiras. Em busca de maiores benefícios para o seu negócio, a empresa solicitou o desenvolvimento de um site Web, disponibilizando informações sobre produtos e serviços oferecidos. O estudo de caso apresentado foi realizado na empresa Linkway, localizada na cidade de São Carlos – SP, Brasil.

Esse estudo de caso tem a vantagem de representar uma aplicação real; no entanto, ele não ilustra todas as características do método proposto. Nas próximas subseções, as duas primeiras fases do método Web-SEMP são exemplificadas.

4.1. Aplicação da Fase I

Passo 1:

De acordo com as primeiras reuniões realizadas entre a empresa Linkway e a Mirex, foram identificados três *stakeholders* para a aplicação Web: um dos diretores da Mirex (representando a empresa), um representante do departamento de *marketing* e um representante de balconista. Também se identificou um perfil de usuário, denominado cliente Mirex, representando o público-alvo da empresa.

A partir de discussões iniciais com os principais *stakeholders* identificados, foi possível determinar a necessidade de modelar uma visão pública e duas visões privadas, sendo uma voltada para balconistas e a outra voltada para o gerenciamento de conteúdos.

Passo 2:

A Tabela 3 mostra cada visão identificada no passo anterior e os *stakeholders* associados a cada uma delas (Passo 2.1).

Tabela 3. Visões da aplicação e stakeholders.

Visão da Aplicação	Stakeholders
Visão para Clientes (Visão Pública)	Diretor da Mirex Representante Marketing Cliente Mirex
Visão para Balconistas (Visão Privada)	Diretor da Mirex Representante Balconista
Visão para Gerenciamento (Visão Privada)	Diretor da Mirex

Para os *stakeholders* de cada visão, elaboraram-se os diagramas GTR (Passo 2.2). Por razões de falta de

espaço, serão descritos alguns objetivos e requisitos referentes ao diretor da Mirex e ao cliente Mirex com relação à visão pública e alguns requisitos com relação à visão privada. As Figuras 2, 3 e 4 mostram, respectivamente, trechos dos diagramas GTR para a visão pública e para a visão privada.

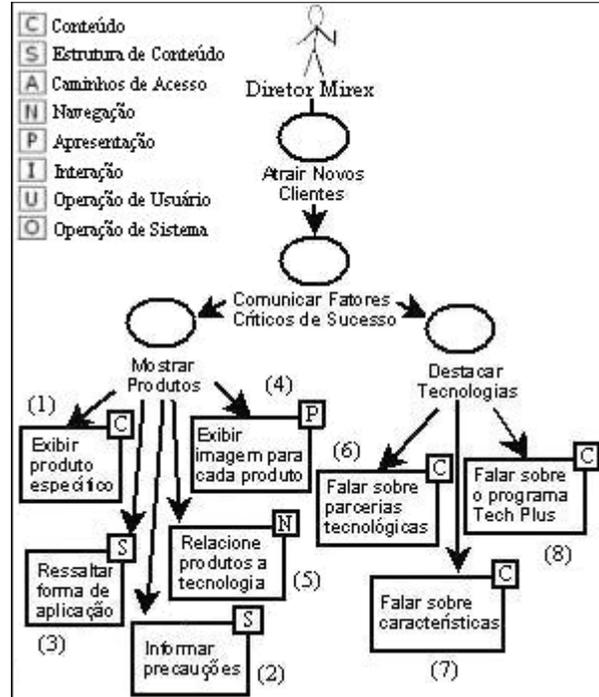


Figura 2. Diagrama GTR para o Diretor da Mirex – Visão Pública.

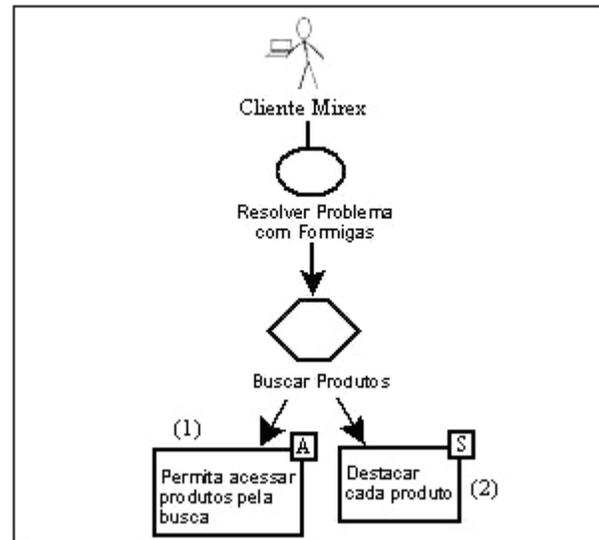


Figura 3. Diagrama GTR para o Cliente Mirex – Visão Pública.

Dentre outros objetivos (não representados no diagrama), o principal objetivo do diretor da Mirex

com relação à visão pública era “atrair novos clientes”. A partir desse objetivo de mais alto-nível, derivou-se o sub-objetivo “comunicar fatores críticos de sucesso”, ressaltando as principais características dos produtos fornecidos e as tecnologias usadas no combate às formigas (Figura 2). Do ponto de vista do cliente Mirex, foram identificados novos requisitos de hiper-mídia, originados a partir do objetivo “resolver problema com formigas”, conforme mostra a Figura 3.

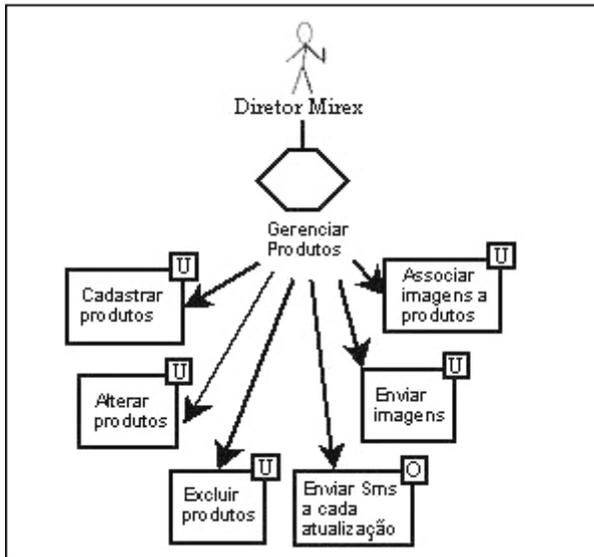


Figura 4. Diagrama GTR para o Diretor da Mirex - Visão Privada.

Para a visão privada, a única tarefa representada pela Figura 4 é “gerenciar produtos”, decomposta em 7 requisitos de operação de usuário e um requisito de operação de sistema.

4.2. Aplicação da Fase II

Para mostrar a aplicação da Fase II, as visões públicas das Figuras 2 e 3 foram submetidas às Etapas 1 e 2, enquanto a visão privada da Figura 4 foi submetida apenas à Etapa 3.

A **Etapa 1** foi conduzida de acordo com os 7 passos definidos no método. Nas Figuras 5 e 6, os passos dessa etapa estão destacados, com linhas pontilhadas e os números entre parênteses são os mesmos dos diagramas GTR, representados nas Figuras 2 e 3. Esses números são usados apenas para explicar a associação entre cada requisito que compõe os diagramas GTR e trechos do sub-modelo de composição e, portanto, não fazem parte da aplicação da Fase II propriamente.

Passo 1:

Todos os requisitos de hiper-mídia em cada um dos diagramas GTR foram numerados, como mostrado nas Figuras 2 e 3.

Passos 2 e 3:

A partir dos diagramas GTR, identificaram-se os temas “produto” e “tecnologia”, de acordo com os requisitos de conteúdo (C) e de estrutura de conteúdo (S), os quais aparecem em negrito na Tabela 4. Além disso, eles foram marcados como dinâmicos (D) ou estáticos (E) (**Passo 2**). Em seguida, os mesmos requisitos foram analisados para serem associados a um desses dois temas. (**Passo 3**). Como exemplo, ao tema “produto” foram associados os seguintes requisitos: “produto específico”, “precauções” e “forma de aplicação”. A Tabela 4 representa o resultado final da aplicação dos passos 2 e 3.

Tabela 4. Formulário RTS preenchido.

Tema	Requisito	Stakeholder	E/D
Produto	-Produto específico	Diretor Mirex	D
	-Precauções -Forma de Aplicação	Cliente Mirex	
Tecnologia	-Parcerias tecnológicas	Diretor Mirex	E
	-Características		
	-Programa TechPlus		

A partir desse ponto, cada tema – “produto” e “tecnologia” no exemplo apresentado – foi analisado para a derivação de aspectos de projeto.

Passos 4, 5, 6 e 7:

As Figuras 5 e 6 mostram trechos do sub-modelo de composição para os temas “produto” e “tecnologia”, respectivamente.

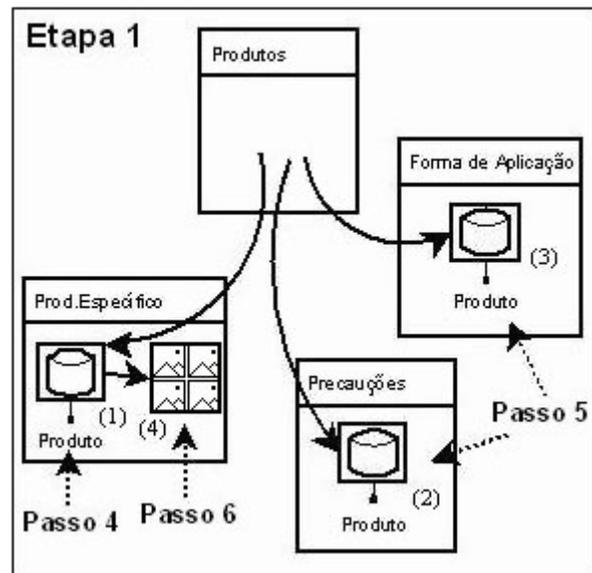


Figura 5. Trecho do Sub-Modelo de Composição para Produtos após a aplicação da Etapa 1.

Para o tema “produto”, apenas o passo 7 não foi conduzido, pois não há requisitos de interação nos diagramas GTR relacionados a esse tema. Já para o tema “tecnologia”, apenas o passo 4 foi conduzido, pois há apenas requisitos de conteúdo relacionados a esse tema, os quais foram mapeados em *text units*, como pode ser observado na Figura 6.

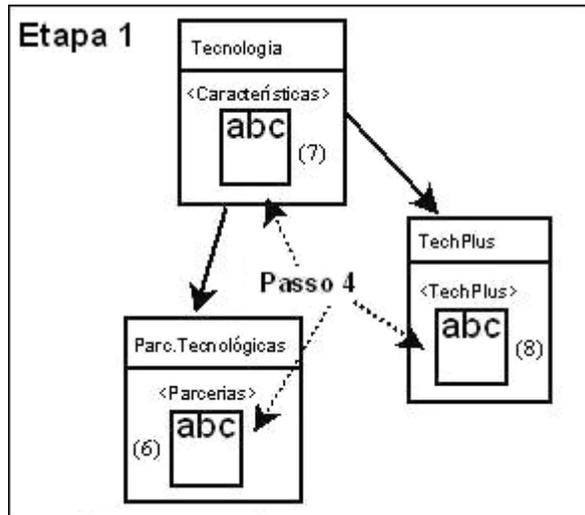


Figura 6. Trecho do Sub-Modelo de Composição para Tecnologias após a aplicação da Etapa 1.

A **Etapa 2** foi conduzida de acordo com os 3 passos definidos no método. Assim, para cada diagrama GTR (Figuras 2, 3 e 4), tentaram-se aplicar os Passos 1, 2 e 3 embora nem sempre os quesitos procurados pelos passos tenham sido observados em todos os diagramas. O resultado se resume no seguinte:

Passo 1:

Foi identificado apenas um requisito de navegação (marcado com “N” na Figura 2), relacionando os dois temas: produto e tecnologia. Esse requisito foi modelado por meio de um *link* não-contextual entre as páginas principais correspondentes aos dois temas, conforme mostra a Figura 7.

Passo 2:

Foi identificado um requisito de caminho de acesso ao conteúdo (Figura 3, marcado com “A”), relacionado ao tema produto. Para resolver esse requisito, adicionou-se uma página para busca de produtos. Além disso, para dar acesso a todas as informações sobre um produto, adicionou-se uma *index unit* na página principal do tema “produto”. Uma página “Home” foi criada para a visão pública dando acesso a todos os temas principais da aplicação Web – incluindo os temas “produto” e “tecnologia”, como pode ser observado na Figura 7.

Passo 3:

Foram identificados apenas dois casos de uso navegacionais, “acessar produtos” e “acessar

tecnologias”, cada qual relacionado a um tema do modelo de hipertexto da Figura 7. Esses casos de uso estão delimitados com uma linha tracejada na própria Figura 7.

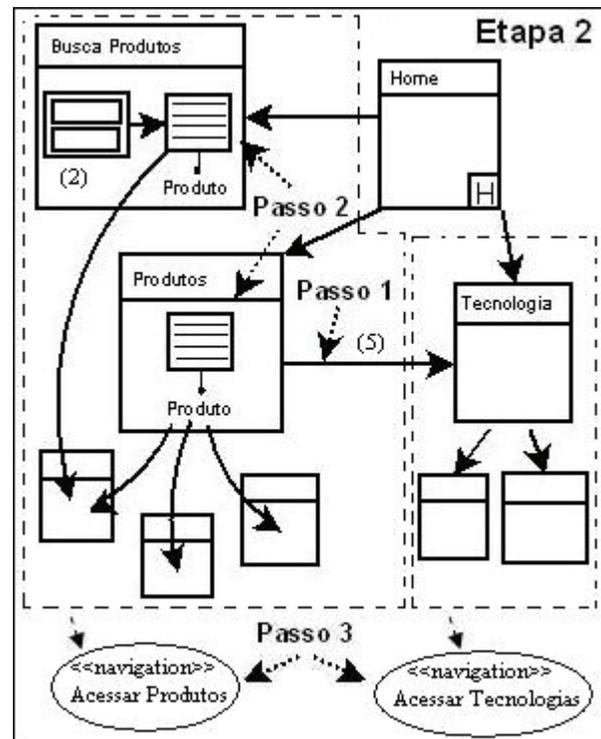


Figura 7. Trecho do Modelo de Hipertexto após a Aplicação da Etapa 2.

A **Etapa 3** não foi conduzida para o trecho da visão pública (Figuras 2 e 3) porque não havia quaisquer funcionalidades. Para o trecho da visão privada, cujo diagrama GTR é apresentado na Figura 4, a Etapa 3 levou à elaboração de um modelo inicial de casos de uso funcionais, conforme mostra a Figura 8.

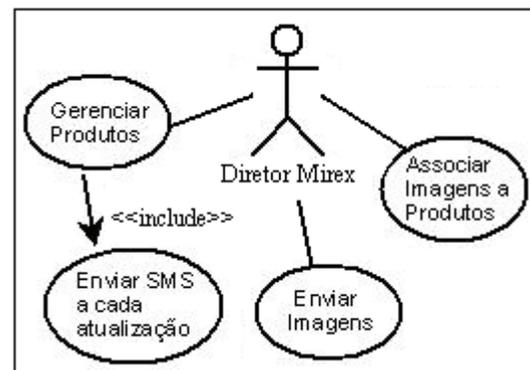


Figura 8. Casos de Uso Funcionais.

É importante ressaltar que o trecho da visão privada apresentado não foi submetido às Etapas 1 e 2, pelo fato de ser um exemplo pequeno. A partir do estudo de

caso realizado, foi possível observar que os modelos construídos facilitaram a comunicação entre todos os envolvidos no projeto, independentemente do seu conhecimento técnico. Embora o método Web-SEMP tenha exigido um treinamento inicial para ser adotado, não se encontraram dificuldades durante a sua aplicação e as suas representações permitiram que todos os *stakeholders* discutissem em torno dos modelos e fornecessem mais requisitos.

5. Conclusões

Este artigo apresentou as Fases I e II do método Web-SEMP, cujo objetivo é dar apoio às atividades de elicitação, modelagem e planejamento de aplicações Web.

Esse método foi definido por observarem-se, na literatura, algumas lacunas tais como: i) dentre vários métodos de ER para Web, não se encontrou algum que tratasse a fase de ER como um todo, dando suporte à elicitação de requisitos e propondo uma migração bem definida para a modelagem desses requisitos; ii) a maioria dos métodos é aplicada de forma isolada, e não integra algumas das práticas e princípios da engenharia de software.

Para suprir essas lacunas, o método Web-SEMP agrega métodos propostos na literatura, os quais possuem, individualmente, características que dão um bom suporte para aplicações Web. Esses métodos são: 1) AWARE, que é baseado em objetivos, usado na fase de levantamento de requisitos, 2) WebML, que dá apoio à elaboração do modelo de dados e modelo navegacional, mas que foi estendida, para retratar alguns itens ainda não considerados, e 3) modelo de casos de uso funcionais e navegacionais, pois estes permitem representar os dois enfoques que podem ser encontrados em uma aplicação Web.

Apresentou-se também um exemplo de aplicação das Fases I e II do método em um sistema Web real, desenvolvido em uma empresa. Com base nesse exemplo, pode-se observar que o objetivo de manter sob controle, em um único processo, as atividades de análise e projeto de requisitos é alcançado, contribuindo para a melhoria da qualidade do processo de desenvolvimento de aplicações Web.

Como trabalhos futuros, pretende-se desenvolver um ambiente automatizado que dê apoio às etapas pertinentes do método Web-SEMP, bem como utilizá-lo em mais aplicações Web a fim de avaliar com maior precisão a sua efetividade. Além disso, pretende-se definir atividades de inspeção ao longo do processo, com o objetivo de aumentar a qualidade dessas fases iniciais de desenvolvimento de aplicações Web.

6. Agradecimentos

Agradecemos à equipe de desenvolvimento da empresa Linkway pela participação no estudo de caso e a CAPES pelo apoio financeiro.

7. Referências

- [1] Pressman, R. S. *Engenharia de Software*. 6. ed., McGraw-Hill, 2006. 720 p.
- [2] J. Zhang, C. K. Chang, and J.-Y. Chung “Mockup-driven Fast-prototyping Methodology for Web Requirements Engineering”. In: *Proceedings of the 27th Annual International Conference on Computer Software and Applications (COMPSAC), Washington, DC*. IEEE Computer Society, 2003.
- [3] J. Zhang, and U. Buy. “A Framework for the Efficient Production of Web Applications”. In: *Proceedings of the 8th IEEE International Symposium on Computers and Communications, Washington, DC*. IEEE Computer Society, 2003, pp. 419-424.
- [4] Y. Deshpande, and S. Hansen. “Web Engineering: Creating a Discipline among Disciplines”. *IEEE MultiMedia* 8, 2001, pp. 82-87.
- [5] Kappel, G., Proll, B, Reich, S., and Retschitzegger, W. *Web Engineering: The Discipline of Systematic Development of Web Applications*. John Wiley & Sons, 2006. 366 p.
- [6] D. Bolchini, P. Paolini, and G. Randazzo. “Adding Hypermedia Requirements to Goal-Driven Analysis”. In: *Proceedings of the 11th IEEE International Conference on Requirements Engineering*. RE. IEEE Computer Society, Washington, DC, 127, 2003.
- [7] D. Bolchini, and P. Paolini. “Goal-driven requirements analysis for hypermedia-intensive Web applications”. *Requir. Eng.*, 2004, pp. 85–103.
- [8] V. Perrone, and D. Bolchini. “Designing Communication Intensive Web Applications: Experience and Lessons from a Real Case”. In: *Proceedings of WER 2004*, Tandil, Argentina, 2004.
- [9] V. Perrone, D. Bolchini, A. Rastellini, and L. Dragone. “Shaping Requirements for Institutional Web Applications: Experience from an Industrial Project”. In: *Proceedings of the 13th IEEE International Conference on Requirements Engineering (Re'05), Washington, DC*. IEEE Computer Society, 2005. pp. 221-230.
- [10] V. Perrone, D. Bolchini, and P. Paolini. “A stakeholders centered approach for conceptual modeling of communication-intensive applications”. In: *Proceedings of the 23rd Annual International Conference on Design of*

Communication: Documenting & Designing For Pervasive information. SIGDOC '05. ACM, New York, NY, 2005, pp. 25-33.

[11] A. Dardenne, A. van Lamsweerde, and S. Fickas. "Goal-directed requirements acquisition". In: *Selected Papers of the Sixth international Workshop on Software Specification and Design* M. Sintzoff, C. Ghezzi, and G. Roman, Eds. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, 1993, pp. 3-50.

[12] H. Al-Subaie, and T. Maibaum. "Evaluating the Effectiveness of a Goal-Oriented Requirements Engineering Methodology", In: *IEEE International Workshop on Comparative Evaluation in Requirements Engineering (CERE)*, Los Alamitos, California: IEEE Computer Society Press, 2006, pp. 8-19.

[13] E. Yu. "Modeling organizations for information systems requirements engineering". In: *Proceedings of the 1st international symposium on requirements engineering*, RE'93, San Jose, USA, 1993.

[14] J. Castro, M. Kolp, and J. Mylopoulos. "Towards requirements driven information systems engineering: the TROPOS project". *Inform Syst*, 2002, pp. 365-389.

[15] A. I. Antón, and C. Potts. "The use of goals to surface requirements for evolving systems". In: *Proceedings of the 20th international Conference on Software Engineering*. International Conference on Software Engineering. IEEE Computer Society, Washington, DC, 1998, pp. 157-166.

[16] L. S. Al-Salem, and A. B. Samaha. "Eliciting Web application requirements - an industrial case study". *J. Syst. Softw.* 80, 3, 2007, pp. 294-313.

[17] S. J. Bleistein, K. Cox, and J. Verner. "Modeling Business Strategy in E-Business Systems Requirements Engineering," *Lecture Notes in Computer Science, Volume 3289*, Berlin: Springer, 2004, pp. 617-628.

[18] J. Gordijn, and J. M. Akkermans. "Value-based requirements engineering: exploring innovative e-commerce ideas". *Requir. Eng.* 2003, pp.114-134.

[19] M. Meldrum, and J. Rose. "Activity based generation of requirements for Web-based information systems: the SSM/ICDT approach". In: *The 12th European Conference on Information Systems*, Turku Finland, 2004.

[20] A. Redouane. "Experience Using Formal Methods for Capturing Requirements of Web-Based Applications". In: *Proceedings of the 1st IEEE international Conference on Cognitive informatics (ICCI)*, Washington, DC. IEEE Computer Society, 2002, pp. 213-221.

[21] A. Redouane. "Towards a New Method for the Development of Web-Based Applications". In: *Proceedings of the 3rd IEEE international Conference on Cognitive informatics (Icci'04)*, Washignton, DC. IEEE Computer

Society, 2004, pp. 116-122.

[22] D. Bolchini, and J. Mylopoulos. "From Task-Oriented to Goal-Oriented Web Requirements Analysis". In: *Proceedings of the Fourth international Conference on Web information Systems Engineering*. WISE. IEEE Computer Society, Washington, DC, 2003, 166.

[23] I. Jacobson, M. Christerson, P. Jonsson, and G. Overgaard. *Object-Oriented Software Engineering – A Use Case Driven Approach*. Addison-Wesley Publish Company, 1992. 528 p.

[24] D. Leffingwell, and D. Widrig. *Managing Software Requirements: a Use Case approach*. 2. ed. Addison-Wesley, 2003, 800 p.

[25] L. Baresi, F. Garzotto, and P. Paolini. "Extending UML for Modeling Web Applications". In: *Proceedings of the 34th Annual Hawaii international Conference on System Sciences*. HICSS. IEEE Computer Society, Washington, DC, 2001.

[26] N. Koch, and A. Kraus. "The Expressive Power of UML-Based Web Engineering", In: *Proceedings of 2nd International Workshop on Web-Oriented Software Technology*, Malaga, Spain, 2002.

[27] G. Rossi, D. Schwabe, and R. Guimaraes. "Designing personalized web applications", In: *Proceedings of WWW10*, Hong Kong, 2001.

[28] S. Ceri, P. Fraternali, and A. Bongio, A. "Web Modeling Language (WebML): a modeling language for designing Web sites". In: *Proceedings of the 9th international World Wide Web Conference on Computer Networks : the international Journal of Computer and Telecommunications Netowrking*. North-Holland Publishing Co., Amsterdam, The Netherlands, 2000, pp.137-157.

[29] S. Ceri, F. Daniel, and M. Matera. "Extending WebML for modeling multi-channel context-aware web applications", In: *Proceedings of Fourth International Conference on Web Information Systems Engineering Workshops (WISEW'03)*, Rome, Italy, IEEE Press, 2003, pp. 225-233.

[30] J. Conallen. *Building Web Applications With UML*, 2. ed. Reading, MA: Addison-Wesley, 2003.

[31] S. D. Martino, F. Ferrucci, L. Paolino, M. Sebillio, G. Vitiello, and G. Avagliano. "A WebML-based Visual Language for the Development of Web GIS Applications". In: *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing*. VLHCC. IEEE Computer Society, Washington, DC, 2007, pp. 209-214.

[32] M. Pang, S. Woojong, K. Jongho, and L. Heeseok. "A Benchmarking-Based Requirement Analysis Methodology for Improving Websites". *KMIS International Conference*, Jeju, Korea, 2005, pp. 680-685.