

Extensão de um metamodelo de Aplicações Baseadas na Web considerando Ajax

Euler H. Marinho^{1,2}, Rodolfo F. Resende¹

¹Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)
31.270-010 – Belo Horizonte – MG – Brasil

²Departamento de Ciências Exatas e Aplicadas
Universidade Federal de Ouro Preto (UFOP)
35.931-008 – João Monlevade – MG – Brasil

{eulerhm, rodolfo}@dcc.ufmg.br

***Abstract.** This paper presents some concepts of web-based applications and the characteristics of RIA applications that distinguish them of traditional web-based applications. The role of metamodels in various activities of software development process is described. In this context, an extended metamodel of web-based applications is showed, gathering two different structures. This metamodel has been important in our effort to prepare a support infrastructure for the web-based applications test.*

***Resumo.** Esse artigo apresenta alguns conceitos de aplicações baseadas na web e as características de aplicações RIAs que as distinguem de aplicações tradicionais baseadas na web. O papel dos metamodelos em diversas atividades do processo de desenvolvimento de software é descrito. Nesse contexto, um metamodelo estendido para as aplicações baseadas na web é apresentado, agrupando duas diferentes estruturas. Este metamodelo tem sido importante em nosso esforço para preparar uma infraestrutura de suporte ao teste de aplicações baseadas na web.*

1. Introdução

A World Wide Web (WWW), originalmente, tinha por objetivo a apresentação de informação que consistia basicamente em documentos no formato hipertexto [Connalen 2002]. A partir de 1993, com a apresentação do navegador Mosaic [Andreessen e Bina 1994], o primeiro navegador popular, o uso da Web experimentou um grande crescimento.

O paradigma web tradicional de desenvolvimento de aplicações baseadas na web era baseado na apresentação de várias páginas para o usuário (multipáginas) e permitia uma interação pouco flexível. O surgimento das chamadas Aplicações Ricas para Internet (*Rich Internet Applications* - RIAs) foi possibilitado pelas tecnologias de desenvolvimento como Ajax (*Asynchronous Javascript and XML*) e permite uma interação com o usuário mais avançada e sofisticada [Marchetto *et al.* 2008]

Conforme ressaltado por Di Lucca e Fasolino [Di Lucca e Fasolino 2006], uma aplicação baseada na web pode ser considerada um tipo de sistema distribuído, com uma arquitetura cliente-servidor ou multi-camadas com as seguintes características:

- um grande número de usuários distribuídos;
- ambiente de execução heterogêneo composto de vários tipos de hardware, conexões de rede, sistemas operacionais, servidores e navegadores;
- natureza extremamente heterogênea que depende da grande variedade de componentes que a constituem. Estes componentes podem ser escritos em diferentes linguagens de programação e serem de natureza diferente (por exemplo, componentes legados, COTS);
- a habilidade de gerar componentes em tempo de execução de acordo com a entrada do usuário e status do servidor.

O desenvolvimento das RIAs trouxe novos desafios aos desenvolvedores [Lawton 2008]. Como exemplo, seja o fato de que as aplicações baseadas em AJAX usam intensivamente a comunicação assíncrona e que o programador faça a suposição de que a resposta do servidor será recebida imediatamente após o envio da requisição, com nenhum evento acontecendo entre o envio de cada mensagem. Nessa situação, poderão ocorrer intercalação das mensagens do servidor, chamadas retornadas (*callbacks*) trocadas e execução da aplicação sob um estado inválido do DOM (*Document Object Model*) [Marchetto *et al.* 2008]. Ademais, existem várias tecnologias (Ajax, Silverlight, Flex, AIR, Ruby on Rails, JavaFX) e diversos arcabouços (*frameworks*) [AjaxPatterns 2011] disponíveis para implementação desse tipo de aplicação. Dessa maneira, torna-se necessária uma melhor compreensão do domínio dessas aplicações.

Nesse contexto, o uso de metamodelos mostra-se importante para a análise e compreensão da estrutura das aplicações baseadas na web, com o objetivo de realizar tarefas de manutenção, teste e melhoria da qualidade [Ricca e Tonella 2002, Pág. 95; DiLucca *et al.* 2004, Pág. 76]. Assim, nesse trabalho, será apresentada uma extensão do metamodelo de DiLucca e outros [DiLucca *et al.* 2004] com a incorporação da estrutura das RIAs implementadas em Ajax.

Este trabalho está organizado como segue. Na seção 2 serão apresentados alguns conceitos de aplicações baseadas na web. Na seção 3, comenta-se alguns aspectos práticos e operacionais do desenvolvimento de aplicações RIAs, com a menção de alguns arcabouços. Na seção 4, são apresentadas as características que distinguem as RIAs das aplicações tradicionais baseadas na web, mencionando as vantagens e as desvantagens dessas características. Na seção 5, será apresentado um metamodelo estendido para aplicações baseadas na web. Por último, na seção 6, faz-se a conclusão.

2. Conceito de Aplicações Baseadas na Web

Alalfi e outros [Alalfi *et al.* 2009] apresentam uma aplicação baseada na web como uma aplicação de software acessível por um *thin-client* (por exemplo um navegador) através de uma rede como a Internet ou através de intranets. Sua arquitetura mais comum é de três camadas, conforme ilustrado pela Figura 1. A primeira camada é representada pelo navegador. O lado servidor (web server), que apresenta componentes em diversas

tecnologias tais como CGI, PHP, Java Servlets ou Active Server Pages (ASP), juntamente com os componentes que interagem com o banco de dados, constituem a camada intermediária. A terceira camada corresponde ao servidor do sistema de banco de dados, constituído pelos dados da aplicação e o sistema de gerência de banco de dados.

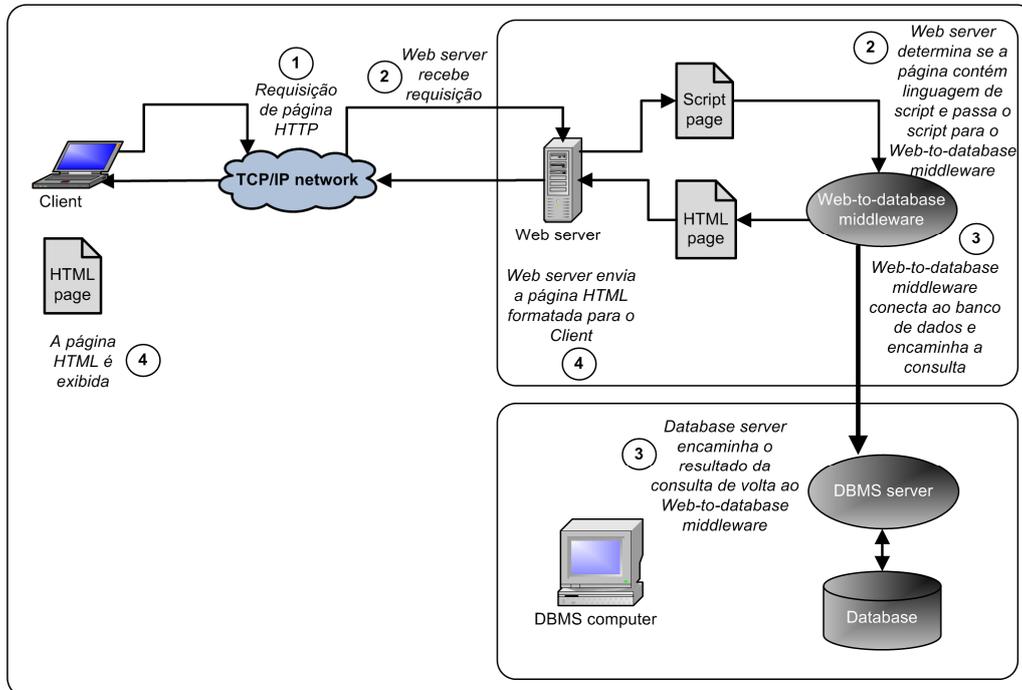


Figura 1. Componentes de uma aplicação baseada na web - adaptada de Rob e Coronel [Rob e Coronel 2009, Pág. 588].

Aplicações tradicionais baseadas na web geram páginas que consistem em diferentes tipos de informação tais como texto, imagens e formulários. Essas páginas podem ser estáticas ou dinâmicas. Páginas estáticas, que são providas por um servidor web, contêm apenas código em HTML e códigos executáveis no lado cliente (por exemplo, Javascript). Por outro lado, páginas dinâmicas, providas pelo servidor da aplicação, resultam da execução de vários scripts e componentes.

As aplicações tradicionais baseadas na web foram classificadas por Tilley e Huang [Tilley e Huang 2001] em três classes com complexidade crescente. A classe 1 é constituída por aplicações estáticas implementadas em HTML e que não disponibilizam interatividade com o usuário. A classe 2 provê interação do cliente com páginas HTML dinâmicas, que associam ações implementadas em scripts à eventos gerados pelo usuário. Por sua vez, a classe 3 reúne as aplicações de conteúdo dinâmico cujas páginas podem ser criadas em tempo de execução, de acordo com a interação do usuário com a aplicação. Usualmente, na implementação dessas aplicações são empregadas várias tecnologias tais como Java Server Pages (JSP), Active Server Pages (ASP) Java Servlets, PHP, CGI, JDBC.

2.1. Metamodelo de uma aplicação tradicional baseada na web

Um metamodelo de aplicação tradicional baseada na web pertinente a classe 3 foi apresentado por Di Lucca e outros [Di Lucca *et al.* 2004] com base no modelo conceitual discutido por Conallen [Conallen 1999; Conallen 2002]. O diagrama de classes que representa este metamodelo é ilustrado na Figura 2, onde diferentes classes e associações representam as diferentes categorias de componentes e associações entre estes.

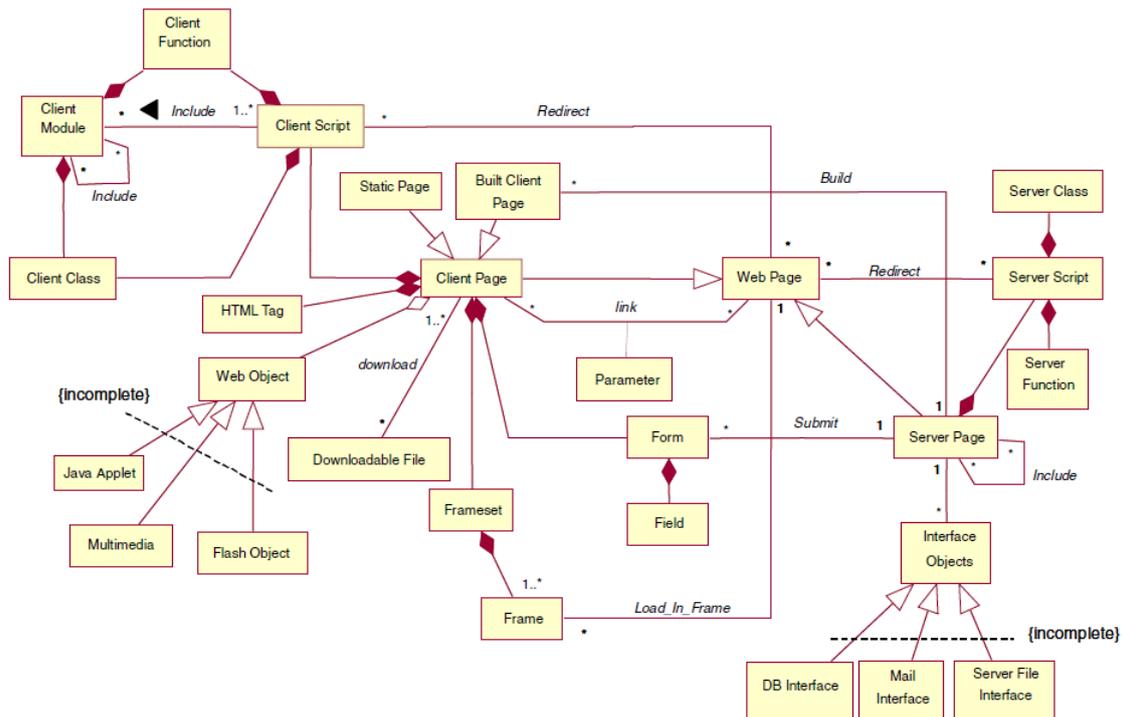


Figura 2. O metamodelo de uma aplicação tradicional baseada na web apresentado por DiLucca e outros [Di Lucca *et al.* 2004, Pág. 75].

De acordo com esse metamodelo, uma aplicação web é constituída de Páginas da Web (*Web Pages*), que podem ser do tipo Páginas do Servidor (*Server Pages*), ou seja, páginas que estão no servidor web, ou Páginas do Cliente (*Client Pages*), páginas que o servidor web envia para o cliente em resposta a uma requisição. As Páginas do Cliente podem ser classificadas como Páginas Estáticas (*Static Pages*), quando seu conteúdo é fixo, ou Páginas Geradas para o Cliente (*Built Client Pages*), quando são geradas sob demanda por uma Página do Servidor. Uma Página do Cliente é composta de Marcadores de HTML (*HTML Tags*) e pode conter Conjuntos de Quadros (*Framesets*), constituídos por 1 ou mais Quadros (*Frames*) onde cada Página da Web pode ser carregada. Outros elementos de uma Página do Cliente: Scripts do Cliente (*Client Scripts*), que implementam alguma ação de processamento, Objetos da Web (*Web Objects*) tais como *Java Applets*, elementos Multimídia (*Multimedia*) (sons, vídeos ou imagens) ou conteúdos *Flash*. Um Script do Cliente pode incluir alguns Módulos do Cliente (*Client Modules*) e, ambos, podem conter Funções do Cliente (*Client Functions*) ou Classes do Cliente (*Client Classes*). Um Script do Cliente pode redirecionar a navegação para outra Página da Web. Pode haver um elo de hiperligação entre uma Página do Cliente e outra Página da Web, que pode ser caracterizado por

algum Parâmetro (*Parameter*). Um Arquivo que Pode Ser Copiado (*Downloadable File*) pode estar associado a uma Página do Cliente que, por sua vez, pode incluir um Formulário (*Form*), composto de diferentes tipos de Campos (*Fields*) (tais como caixas de seleção, botões, áreas de texto etc). Formulários são utilizados para agrupar entradas do usuário e repassa-las (*submit*) a uma Página do Servidor. Uma Página do Servidor pode ser composta de alguns Scripts do Servidor (*Server Scripts*) (que podem incluir Classes do Servidor (*Server Classes*) ou Funções do Servidor (*Server Functions*)) implementando algumas ações de processamento, as quais podem redirecionar a requisição para outra Página da Web, ou dinamicamente construir uma Página Gerada para o Cliente. Por último, uma Página do Servidor pode incluir outras Páginas do Servidor, e pode estar associada a outros Objetos de Interface (*Interface Objects*), ou seja, objetos que permitem a conexão da aplicação web com servidores de banco de dados, email, arquivos.

HTML é uma linguagem de marcação de hipertexto, que informa a estrutura e a aparência de documentos acessíveis por um navegador. A versão 4.0 desse padrão trouxe suporte para folhas de estilo para formatação e apresentação de conteúdo, proporcionando ao desenvolvedor e ao usuário controle sobre a aparência do documento, bem como o suporte a scripts [W3C 1999]. Na Tabela 1 são apresentados alguns dos novos elementos especificados no padrão HTML 5 [W3C 2010a; W3C 2010b].

Tabela 1. Elementos novos do padrão HTML 5

Elemento	Propósito	Descrição
<article>	Estrutural	Representa um pedaço independente de conteúdo, tal como uma entrada em um blog ou artigo de jornal.
<section>	Estrutural	Representa uma seção genérica de um documento ou aplicação.
<header>	Estrutural	Representa um grupo de ajuda introdutória ou navegacional.
<footer>	Estrutural	Representa um rodapé de uma seção e pode conter informações sobre o autor, copirraite etc
<hgroup>	Estrutural	Representa um cabeçalho de uma seção.
<aside>	Estrutural	Representa uma parte de conteúdo que está ligeiramente relacionada ao resto da página.
<figure>	Estrutural	Representa um conteúdo de fluxo auto-contido, tipicamente referenciado como uma única unidade no fluxo principal do documento. Conteúdo de fluxo representa a maioria dos elementos utilizados no corpo do documento (, <table>, <math>, <p> etc)
<video>	Conteúdo	Provê uma API (<i>Application Programming Interface</i>) para a execução de vídeos e filmes.
<audio>	Conteúdo	Provê uma API para a execução de sons e fluxos de áudio.
<canvas>	Conteúdo	Utilizado para renderização sob demanda de gráficos bitmap, como jogos, imagens etc.
<embed>	Conteúdo	Representa um ponto de integração para uma aplicação externa (não HTML) ou conteúdo interativo.
<command>	Conteúdo	Representa um comando que pode ser invocado pelo usuário.

2.2. Metamodelo de uma aplicação RIA

Ajax, termo proposto por Garrett [Garrett 2005], tornou-se uma tecnologia proeminente na implementação de aplicações baseadas na web [Mesbah e van Deursen 2009].

Mesbah e van Deursen [Mesbah e van Deursen 2008] propõem um metamodelo para aplicações Ajax, segundo o estilo arquitetural denominado SPIAR (Single Page Internet Application aRchitecture). Esse estilo abstrai as características comuns de vários arcabouços e enfatiza o desenvolvimento baseado em componentes de interfaces com o usuário, a comunicação delta entre componentes cliente/servidor e notificação de mudanças de estado através dos componentes. A comunicação delta é aquela em que apenas informações sobre estado são trocadas entre o cliente e o servidor, evitando o fluxo de informações redundantes. Segundo os autores, essas características do estilo arquitetural SPIAR teriam por objetivo a melhoria da interatividade do usuário, a diminuição da latência percebida pelo usuário, o aumento da coerência dos dados e maior facilidade de desenvolvimento. O diagrama de classes, representando o metamodelo, é apresentado na Figura 3.

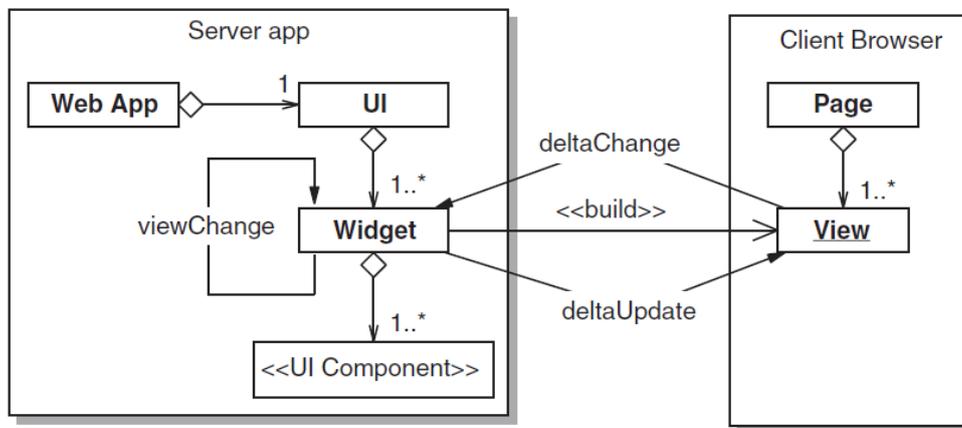


Figura 3. O metamodelo de uma aplicação RIA proposto por Mesbah e van Deursen [Mesbah e van Deursen 2008].

De acordo com esse metamodelo, a interface com o usuário (*UI*) é composta por Elementos de Interface (*Widgets*). Mudanças delta e de visões ocorrem ao nível dos Elementos de Interface. Uma mudança de visão pode ser vista como uma navegação entre os Elementos de Interface disponíveis.

3. Aspectos práticos e operacionais

Conforme definido originalmente por Garrett [Garrett 2005], Ajax fundamenta-se sobre padrões como XHTML e CSS, interações e apresentações dinâmicas utilizando DOM (*Document Object Model*), intercâmbio e manipulação de dados utilizando XML e XSLT, recuperação assíncrona de dados utilizando o objeto XMLHttpRequest e JavaScript. Assim, Ajax representa uma abordagem para o desenvolvimento baseado na web fundamentada em tecnologias bem estabelecidas [Mesbah e van Deursen 2008].

Aplicações Ajax podem ser desenvolvidas e mantidas pelo uso de arcabouços [AjaxPatterns 2011]. Alguns arcabouços têm por objetivo disponibilizar elementos de interface com o usuário avançados enquanto outros focalizam na simplificação do acesso a funções do servidor [Serrano e Aroztegi 2007]. A seguir são descritos brevemente três arcabouços.

GWT

O GWT (Google Web Toolkit) [GWT 2011] é um arcabouço que tem por objetivo facilitar o desenvolvimento de interfaces com o usuário de uma forma similar às APIs (*Application Programming Interface*) AWT e Swing da linguagem Java. Em vez de manter os componentes de interface com o usuário no servidor e realizar comunicações de troca de estado, a interface com o usuário é escrita na linguagem Java e o compilador GWT converte o código resultante para código em JavaScript e HTML compatível com os navegadores. Dessa maneira, toda a funcionalidade da interface com o usuário é disponibilizada no cliente, minimizando a comunicação com o servidor [Mesbah e van Deursen 2007a].

Dojo

O Dojo Toolkit [DJT 2011] é um arcabouço da linguagem JavaScript para desenvolvimento de aplicações Ajax. Este arcabouço oferece vários controles para leiaute e formulários, baseando-se num modelo de elementos de interface extensível [Livshits e Erlingsson 2007]. O Dojo utiliza JavaScript para a definição de diferentes padrões de comportamento e controle para os elementos de interface.

Echo3

O Echo3 [EWF 2011] é arcabouço de código aberto que disponibiliza componentes de interface com o usuário para implementação de aplicações Ajax. Essas aplicações podem ser implementadas como aplicações do lado servidor, pelo uso da linguagem Java e programação dirigida por eventos, ou como aplicações do lado cliente, pelo uso de JavaScript, utilizando XML, JSON (JavaScript Object Notation) ou outro protocolo sobre HTTP para comunicação com o servidor. Nesse caso, a comunicação somente acontece na invocação de serviços web ou no acesso ao sistema de gerência de banco de dados.

De maneira similar, existem arcabouços para o desenvolvimento de aplicações tradicionais baseadas na web como Spring [SSF 2011], Struts [ASF 2011] e Seam [JBS 2011].

4. Peculiaridades das aplicações RIAs

O termo RIA designa uma classe heterogênea de soluções, caracterizada pelo propósito comum de adicionar novas características a aplicações tradicionais baseadas na web. Essas características se fundamentam na combinação de três aspectos fundamentais: a arquitetura distribuída mais leve, a interatividade semelhante à das aplicações desktop e a melhor utilização da capacidade de computação do cliente. Essa combinação traz melhorias para vários aspectos de uma aplicação tradicional baseada na web [Fraternali *et al.* 2010]. Os aspectos, que distinguem as RIAs das aplicações baseadas na web tradicionais [Preciado *et al.* 2007], bem como as vantagens e desvantagens relacionadas [Brambilla *et al.* 2008; Mahemoff 2006; Busch e Koch 2009], encontram-se descritos na Tabela 2.

Tabela 2. Aspectos aprimorados pelas RIAs

Aspecto	Descrição	Vantagens	Desvantagens
Distribuição de Dados	Nas RIAs, os dados da aplicação podem ser armazenados tanto no cliente quanto no servidor. No caso dos dados serem armazenados no cliente, estes são enviados ao servidor ao final de cada operação.	- Uso <i>offline</i> da aplicação. - Validação e preparação de dados no cliente.	- Replicação de dados. - Políticas de alocação. - Consistência de dados.
Distribuição da Lógica de negócio	As RIAs possuem uma estrutura de navegação diferente das aplicações tradicionais baseadas na web. Devido à disponibilidade de processamento do cliente, tanto este quanto o servidor podem realizar operações complexas.	- Uso <i>offline</i> da aplicação. - <i>Live validation</i> . - <i>Page Rearrangement</i> . - <i>Display Morphing</i> .	- Cliente requer dados adicionais.
Comunicação Cliente-Servidor	As RIAs permitem tanto a comunicação síncrona quanto a comunicação assíncrona. A distribuição de dados e funcionalidade entre cliente e servidor amplia as características dos eventos produzidos. Esses podem ser originados, detectados e processados de diferentes formas.	- Atualização parcial de página. - <i>Page Rearrangement</i> . - <i>Display Morphing</i> .	- Incremento do esforço de desenvolvimento.
Aprimoramento da Interface de Usuário	As RIAs possibilitam apresentação e interações com o usuário avançadas (por exemplo, facilidades do tipo “arrastar e soltar”, menor tempo de resposta).	- Operação como aplicação de página única. - Evita atualização de página inteira. - Carregamento progressivo de elementos da página.	- Problemas de desempenho. - Incompatibilidades com navegadores.

Na Tabela 2, *Page Rearrangement* e *Display Morphing* designam padrões de projeto que permitem, respectivamente, alterações em elementos individuais da página e alterações na estrutura da página.

5. Um metamodelo estendido para aplicações baseadas na web

Ricca e Tonella [Ricca e Tonella 2002, Pág. 94] ressaltam que os metamodelos desempenham um papel central no desenvolvimento de aplicações baseadas web. Estes são úteis para o entendimento do sistema existente, avaliação e implementação de mudanças nos requisitos e como ponto de partida para as atividades de teste.

Considerando os metamodelos apresentados na seção 2, constata-se que nenhum deles oferece uma visão unificada de uma aplicação web, sendo restritos a aplicações com estrutura particular. Entretanto, em nossos esforços para elaboração do arcabouço WATEST para suporte ao teste de aplicações baseadas na web, existe a demanda por tal metamodelo de forma a centralizar as diferentes estruturas das aplicações baseadas na web.

Aplicações implementadas em Ajax, apresentam uma estrutura de página única ao contrário das aplicações tradicionais que, normalmente, apresentam estrutura de múltiplas páginas [Zucker 2007]. Neste caso, pode-se acrescentar elementos ao metamodelo de forma a exibir as especificidades da estrutura de cada aplicação. Assim, na Figura 4, é apresentado o diagrama de classes de um metamodelo estendido a partir do metamodelo de DiLucca e outros [DiLucca *et al.* 2004, Pág. 75], com a estrutura de aplicações Ajax que seguem o estilo arquitetural SPIAR. Nessa figura, foram ocultados alguns elementos para melhor apresentação da mesma.

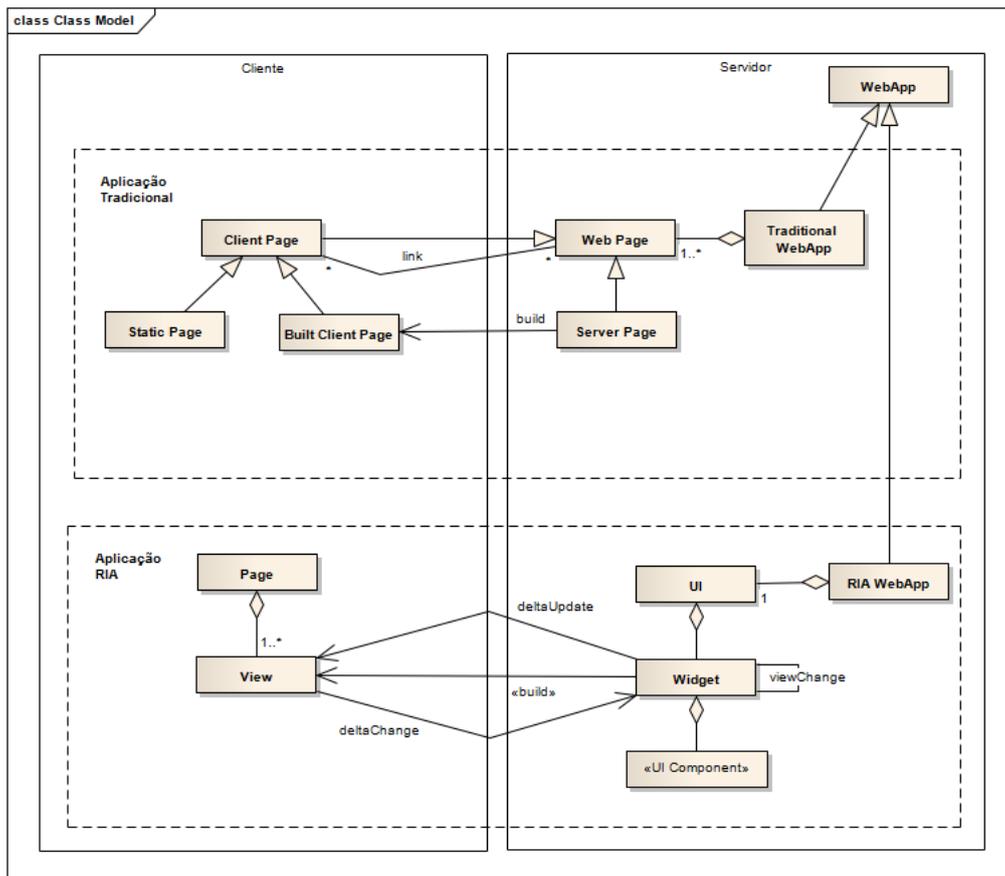


Figura 4. O metamodelo estendido de uma aplicação tradicional baseada na Web com a estrutura de uma aplicação RIA implementada em Ajax.

Conforme pode-se verificar pela Figura 4, houve a inserção de uma classe *WebApp*, que representa a aplicação baseada na web. Isso permite a especialização do metamodelo para aplicações com estrutura particular. Além disso, manter ambas as estruturas mostra-se adequado em processos de migração de aplicações entre tecnologias. Por exemplo, Mesbah e van Deursen [Mesbah e van Deursen 2007b] analisam como aplicações multipáginas poderiam ser refatoradas para aplicações de página única em Ajax, por meio de técnicas de engenharia reversa.

O metamodelo de DiLucca e outros [DiLucca *et al.* 2004, Pág. 75] mostra-se extensível também para representar a estrutura de aplicações RIAs implementadas através de tecnologias como Silverlight e JavaFX que executam em ambientes de tempo de execução específicos. Para a execução desses tipos de aplicações a partir de um navegador, é necessária a instalação de *plugins*. Nesse caso, a extensão do metamodelo para essa classe de aplicações poderia ser realizada pela inserção de classes especializadas a partir da classe *Web Object*. Na Figura 5, é apresentado o metamodelo estendido, considerando aplicações RIAs implementadas em JavaFX.

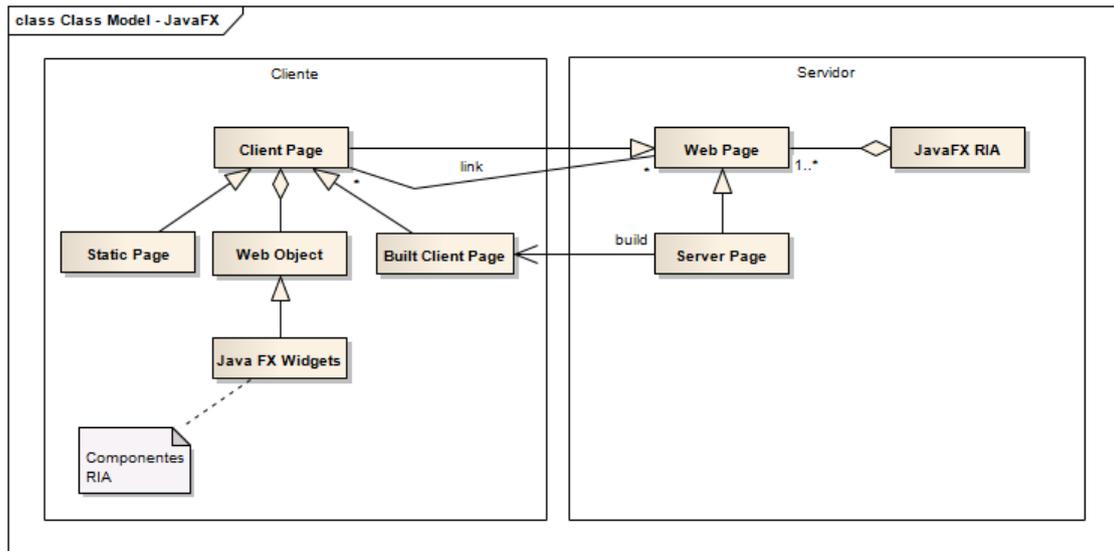


Figura 5. O metamodelo estendido de uma aplicação tradicional baseada na Web com a estrutura de uma aplicação RIA implementada em JavaFX.

6. Conclusão

Nesse trabalho, foram vistos alguns conceitos de aplicações baseadas na web e foram apresentadas as características de aplicações RIAs que as distinguem de aplicações tradicionais baseadas na web. Foi mencionado que os metamodelos desempenham um papel importante em várias atividades no processo de desenvolvimento de software. Um metamodelo para aplicações tradicionais baseadas na web foi proposto por DiLucca e Fasolino [DiLucca e Fasolino, Pág. 75], sendo utilizado na proposta de um arcabouço [DiLucca e Di Penta 2005] para apoio à compreensão de aplicações baseadas na web, visando atividades de manutenção e testes. Entretanto, um metamodelo unificado e extensível torna-se importante para as aplicações baseadas na web, que podem apresentar estruturas próprias dependendo das tecnologias utilizadas.

Nesse contexto, foi proposto um metamodelo estendido a partir do metamodelo mencionado com a incorporação da estrutura das RIAs implementadas em Ajax. Também foi mencionado como o metamodelo original poderia ser estendido para as RIAs implementadas em outras tecnologias como Silverlight e JavaFX. Este metamodelo estendido tem sido útil em nossos esforços para elaboração do arcabouço WATEST para suporte ao teste de aplicações baseadas na web. Além disso, acreditamos que o mesmo poderá ser importante em outros esforços para o suporte dessa classe de aplicação.

Referências

- AjaxPatterns (2011). Frameworks. < <http://ajaxpatterns.org/Frameworks>>. Acesso em 09 de Fevereiro de 2011.
- Alalfi, M. H., Cordy, J. R., Dean, T. R. (2009). Modelling Methods for Web Application Verification and Testing: State of the Art. *Software Testing, Verification and Reliability*, 19(4): 265-296.

- Andreessen, M., Bina, E. (1994). NCSA Mosaic: A Global Hypermedia System. *Internet Research*, 4(1): 7-17.
- ASF (2011). Apache Struts Framework. <<http://struts.apache.org/>>. Acesso em 11 de Fevereiro de 2011.
- Brambilla, M., Preciado, J. C., Linaje, M., Sanchez-Figueroa, F. (2008). Business Process-Based Conceptual Design of Rich Internet Applications. In *ICWE 2008: 8th International Conference on Web Engineering*, pages 155-161, New York, USA.
- Busch, M., Koch, N. (2009). Rich Internet Applications. Technical Report 0902, Institute for Informatics, Ludwig-Maximilians-Universität München, Germany.
- Conallen, J. (1999). Modeling Web Application Architectures with UML. *Communications of the ACM*, 42 (10): 63-70.
- Conallen, J. (2002). *Building Web applications with UML*. Addison-Wesley, 2nd edition.
- Di Lucca, G. A., Di Penta, M. (2005). Integrating Static and Dynamic Analysis to Improve the Comprehension of Existing Web Applications. In *WSE 2005: 7th International Symposium on Web Site Evolution*, pages 87-94, Budapest, Hungary.
- Di Lucca, G. A., Fasolino, A. R. (2006). Testing Web-Based Applications: The State of the Art and Future Trends. *Information and Software Technology*, 48: 1172-1186.
- Di Lucca, G. A., Fasolino, A. R., Tramontana, P. (2004). Reverse Engineering Web Applications: the WARE Approach. *Journal of Software Maintenance and Evolution: Research and Practice*, 16: 71-101.
- DJT (2011). Dojo Toolkit. <<http://dojotoolkit.org>>. Acesso em 11 de Fevereiro de 2011.
- EFW (2011). Echo Web Framework. <<http://echo.nextapp.com/site/>>. Acesso em 11 de Fevereiro de 2011.
- Fraternali, P., Rosi, G., Sánchez-Figueroa, F. (2010). Rich Internet Applications. *IEEE Internet Computing*, 14(3): 9-12.
- Garrett, J. (2005). Ajax: A New Approach to Web Applications. *Adaptive Path*. <<http://www.adaptivepath.com/ideas/essays/archives/000385.php>>. Acesso em 09 de Fevereiro de 2011.
- GWT (2011). Google Web Toolkit. <<http://code.google.com/webtoolkit/>>. Acesso em 11 de Fevereiro de 2011.
- JBS (2011). JBoss Seam Framework. <<http://www.seamframework.org/>>. Acesso em 11 de Fevereiro de 2011.
- Lawton, G. (2008). New Ways to Build Rich Internet Applications. *Computer*, 41(8): 10-12.
- Livshits, B., Erlingsson, U. (2007). Using Web Application Construction Frameworks to Protect Against Code Injection Attacks. In *PLAS 2007: ACM SIGPLAN Workshop on Programming Languages and Analysis for Security*, pages 95-103, San Diego, USA.
- Mahemoff, M. (2006). *Ajax Design Patterns*. O'Reilly Media, 1st edition.

- Marchetto, A., Ricca, F., Tonella, P. (2008). A Case Study-Based Comparison of Web Testing Techniques Applied to Ajax Web Applications. *International Journal on Software Tools for Technology Transfer*, 10: 477-492.
- Meliá, S., Gómez, J., Pérez, S., Díaz, O. (2010). Architectural and Tehnological Variability in Rich Internet Applications. *IEEE Internet Computing*, 14(3): 24-32.
- Mesbah, A., van Deursen, A. (2007a). An Architectural Style for Ajax. In *WICSA 2007: 6th Working IEEE/IFIP Conference on Software Achitecture*, pages 44-53, Mumbai, India.
- Mesbah, A., van Deursen, A. (2007b). Migrating Multi-page Web Applications to Single-page Ajax Interfaces. In *CSMR 2007: 11th European Conference on Software Maintenance and Reengineering*, pages 181-190, Amsterdam, The Netherlands.
- Mesbah, A., van Deursen, A. (2008). Component- and push-based architectural style for Ajax Applications. *The Journal of Systems and Software*, 81: 2194-2209.
- Mesbah, A., van Deursen, A. (2009). Invariant-Based Automatic Testing of Ajax User Interfaces. In *ICSE 2009: 31st International Conference on Software Engineering*, pages 210-220, Vancouver, Canada.
- Preciado, J. C., Linaje, M., Comai, S., Sanchez-Figueroa, F. (2007). Design Rich Internet Applications with Web Engineering Methodologies. In *WSE 2007: 9th IEEE International Workshop on Web Site Evolution*, pages 23-30, Paris, France.
- Ricca, F., Tonella, P. (2002). Testing Processes of Web Applications. *Annals of Software Engineering*, 14: 93-114.
- Rob, P., Coronel, C. (2009). *Database Systems: Design, Implementation and Management*. Course Technology, 8th edition.
- Serrano, N., Aroztegi, J. P. (2007). Ajax Frameworks for Interactive Web Apps. *IEEE Software*, 24(5): 12-14.
- SSF (2011). Spring Framework. <<http://www.springsource.org/>>. Acesso em 11 de Fevereiro de 2011.
- Tilley, S., Huang, S. (2001). Evaluating the Reverse Engineering Capabilities of Web Tools for Understanding Site Content and Structure: A Case Study. In *ICSE 2001: 23rd International Conference on Software Engineering*, pages 514-523, Toronto, Canada.
- W3C (1999). HTML 4.01 Specification. <<http://www.w3.org/TR/html4/>>. Acesso em 25 de Novembro de 2010.
- W3C (2010a). HTML 5: A Vocabulary and Associated APIs for HTML e XHTML. <<http://www.w3.org/TR/html5/>>. Acesso em 25 de Novembro de 2010.
- W3C (2010b). HTML 5 Differences from HTML 4. <<http://dev.w3.org/html5/html4-differences/>>. Acesso em 25 de Novembro de 2010.
- Zucker, D. (2007). What Does Ajax Mean for You?. *Interactions*, 14(5): 10-12.