

FrameWeb Editor: Uma Ferramenta CASE para suporte ao Método FrameWeb

Silas Louzada Campos

Núcleo de Estudos em Modelagem Conceitual e Ontologias (NEMO), Departamento de Informática, Universidade Federal do Espírito Santo (UFES) - Vitória, ES, Brasil
slcampos@inf.ufes.br

Vítor E. Silva Souza

Núcleo de Estudos em Modelagem Conceitual e Ontologias (NEMO), Departamento de Informática, Universidade Federal do Espírito Santo (UFES) - Vitória, ES, Brasil
vitorsouza@inf.ufes.br

ABSTRACT

The FrameWeb method defines a standard architecture to facilitate the integration with widely-used Web development frameworks, proposing a set of models that bring to architectural design the concepts inherent to these frameworks. In this paper, we present the *FrameWeb Editor*, a CASE tool built on top of the meta-models that define the syntax of the FrameWeb language, providing a graphical editor for the creation of valid models in this language and serving as foundation to other functionalities such as, e.g., code generation. The tool has been successfully used to model projects in the context of student course assignments, showing potential for its use in industrial settings.

KEYWORDS

Web engineering, Frameworks, Model-driven development, FrameWeb

1 INTRODUÇÃO

A Web tem sido amplamente utilizada como plataforma de implantação de sistemas de informação, tendo as várias tecnologias utilizadas para o desenvolvimento desses sistemas alcançado um alto nível de maturidade e sido utilizadas em larga escala pela comunidade de desenvolvimento de software. Com o objetivo de facilitar a integração entre tecnologias e também o reúso de soluções já implementadas é comum o uso de *frameworks*, que oferecem um conjunto de classes e bibliotecas que resolvem problemas comuns em diversas aplicações. Porém, apesar de afetarem diretamente a arquitetura dos sistemas, os *frameworks* geralmente não são considerados no processo de desenvolvimento até a fase de implementação.

Dentro deste contexto, destacamos o método FrameWeb [1, 16], um método baseado em *frameworks* para o desenvolvimento de sistemas de informação Web (*Web-based Information Systems* ou WISs). O método define uma arquitetura padrão para facilitar a integração com *frameworks* muito utilizados para o desenvolvimento nessa plataforma, além de propor um conjunto de modelos que traz para o projeto arquitetural do sistema conceitos inerentes a estes *frameworks*.

Baseado em técnicas MDD (*Model-Driven Development*) [15], a sintaxe da linguagem de modelagem do método FrameWeb é definida formalmente por meio de meta-modelos que permitem

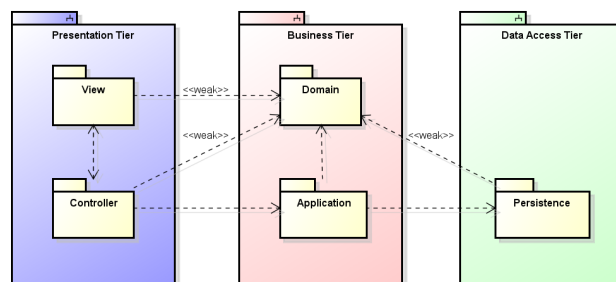


Figura 1: Proposta de arquitetura FrameWeb [16].

a inclusão de novas instâncias de *frameworks* e servem de base para a construção de ferramentas que verifiquem a sintaxe dos modelos construídos, ofereçam geração de código, dentre outras funcionalidades possíveis [11, 12].

Baseado nestes meta-modelos, este artigo propõe a ferramenta CASE (*Computer-Aided Software Engineering*) chamada *FrameWeb Editor*, que provê um ambiente gráfico para a criação de modelos FrameWeb, apresentando suas funcionalidades e aspectos relevantes de sua implementação: a Seção 2 introduz o referencial teórico, a Seção 3 apresenta a ferramenta, a Seção 4 compara trabalhos relacionados e a Seção 5 conclui.

2 REFERENCIAL TEÓRICO

O método FrameWeb (*Framework-based Design Method for Web Engineering*) [1, 16] é um método baseado em técnicas de Engenharia de Software orientadas a objetos, e é direcionado à construção de Sistemas de Informação Web (*Web-based Information Systems*, ou WIS) que possuam uma infra-estrutura arquitetônica baseada no uso de *frameworks*. O método propõe uma arquitetura de software padrão que divide os sistemas em camadas e uma extensão da UML para a construção de modelos de projeto que representem elementos comuns a estes *frameworks*.

A arquitetura lógica padrão para WISs definida pelo método FrameWeb é baseada no padrão arquitetônico *Service Layer* (Camada de Serviço), proposta por Randy Stafford em [7], que divide o sistema em três grandes camadas: lógica de apresentação, lógica de negócio e lógica de acesso aos dados, como mostra a Figura 1.

Com base nesta arquitetura, a linguagem FrameWeb define quatro tipos de diagramas, todos eles baseados no Diagrama de Classes da UML, que são utilizados para representar os componentes específicos de cada camada em questão, abrangendo componentes típicos da plataforma Web e relacionados aos *frameworks* utilizados.

A ferramenta proposta neste artigo (cf. Seção 3) oferece suporte para o desenvolvimento destes diagramas. São eles:

- **Modelo de Entidades** (originalmente Modelo de Domínio), que representa os objetos de domínio do problema e sua persistência por meio do *framework* de mapeamento objeto/relacional (e.g., Java Persistence API¹);
- **Modelo de Persistência**, referente à camada de acesso aos dados, seguindo o padrão de projeto *Data Access Object* (DAO) [3];
- **Modelo de Navegação**, para representar os diferentes componentes que, integrados ao *framework* Web (controlador frontal, e.g., JavaServer Faces²), formam a interface com o usuário e compõem a camada de Apresentação (Visão e Controle);
- **Modelo de Aplicação**, para representar as classes de serviço, responsáveis pela implementação dos casos de uso (funcionalidades do WIS), e suas dependências, gerenciadas pelo *framework* de injeção de dependências (e.g., Contexts and Dependency Injection for Java³).

Baseado em técnicas de Desenvolvimento Dirigido por Modelos (*Model-Driven Development* ou MDD) [15], a sintaxe da linguagem FrameWeb é formalmente definida por meio de meta-modelos que permitem a inclusão de novas instâncias de *frameworks*, dentro das categorias citadas acima (mapeamento objeto/relacional, controlador frontal, injeção de dependências). Tais meta-modelos podem também servir de base para a construção de ferramentas que verifiquem a sintaxe dos modelos construídos, ofereçam geração de código, dentre outras funcionalidades possíveis [11, 12].

Em sua extensão FrameWeb-LD [6], é proposta uma abordagem que integra ao modelo de domínio conceitos inerentes a *linked data*, auxiliando desenvolvedores na criação de WISs integrados à Web Semântica, que tem como principal ideia interligar dados ao invés de documentos e adicionar semântica (significado) a essas ligações.

Na sequência, apresentamos a ferramenta e sua arquitetura.

3 FRAMEWEB EDITOR

Esta seção apresenta o *FrameWeb Editor*. A Seção 3.1 apresenta uma visão geral da ferramenta, a Seção 3.2 descreve seu uso e a Seção 3.3 apresenta sua arquitetura. Um vídeo sobre a ferramenta e algumas de suas características encontra-se em <https://youtu.be/3cpkBcGlfUw>.

3.1 Visão Geral

O *FrameWeb Editor* oferece suporte à criação dos quatro tipos básicos de modelos FrameWeb, fornecendo um ferramental já conhecido em softwares de modelagem porém com características e propriedades próprias da linguagem. Ao contrário de editores UML de propósito geral, a ferramenta é capaz de fazer verificações no modelo e impedir ações inválidas (do ponto de vista da linguagem FrameWeb) como, por exemplo, a criação de uma classe fora de um pacote, ou associações indevidas entre componentes. Além disso, a ferramenta guia o desenvolvedor no uso dos elementos específicos

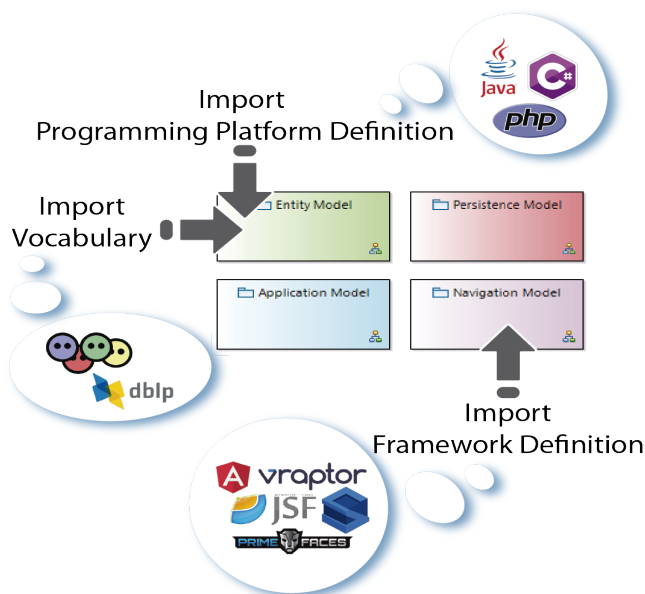


Figura 2: Visão geral da extensibilidade da ferramenta com relação aos *frameworks*, plataformas de programação e vocabulários *linked data*.

da linguagem FrameWeb, não disponibilizando conceitos da UML que não fazem parte do método.

O *FrameWeb Editor* dá suporte à característica de extensibilidade do FrameWeb, descrita na Seção 2, por meio da importação de arquivos de definição de *frameworks*, da plataforma de programação e de vocabulários *linked data*, como mostra a Figura 2. Além disso, a ferramenta permite também a criação e modificação destas definições dentro do próprio editor, provendo apoio a meta-modeladores que desejam estender a linguagem para dar suporte a um *framework*, uma plataforma ou um vocabulário ainda não incluídos.

Por exemplo, para um determinado projeto desenvolvido em Java, utilizando JSF e a biblioteca de componentes PrimeFaces e mapeando os dados ao vocabulário DBLP, tais definições seriam importadas no editor, permitindo que: (a) os componentes PrimeFaces sejam usados como tipos dos atributos dos formulários representados em Modelos de Navegação; (b) regras específicas do JSF se apliquem na relação entre páginas, formulários e controladores nestes mesmos modelos; (c) tipos básicos da linguagem Java sejam usados nos atributos das classes do Modelo de Entidades; (d) ligações entre as classes/atributos deste mesmo modelo sejam estabelecidas com classes e propriedades do vocabulário DBLP.

3.2 Principais funcionalidades

O *FrameWeb Editor* possui representações e painéis de criação de componentes diferentes para cada tipo de modelo: *Entity*, *Persistence*, *Application* e *Navigation*, além de um nível de projeto, que agrega estes quatro modelos, bem como os pacotes de linguagem e de definições de *frameworks*, como mostra a Figura 3. O projeto em questão utiliza a plataforma Java e a tecnologia JavaServer Faces (JSF) como *framework* Web.

¹<http://jcp.org/en/jsr/detail?id=338>

²<http://jcp.org/en/jsr/detail?id=344>

³<http://jcp.org/en/jsr/detail?id=346>

FrameWeb Editor: Uma Ferramenta CASE para suporte ao Método FrameWeb

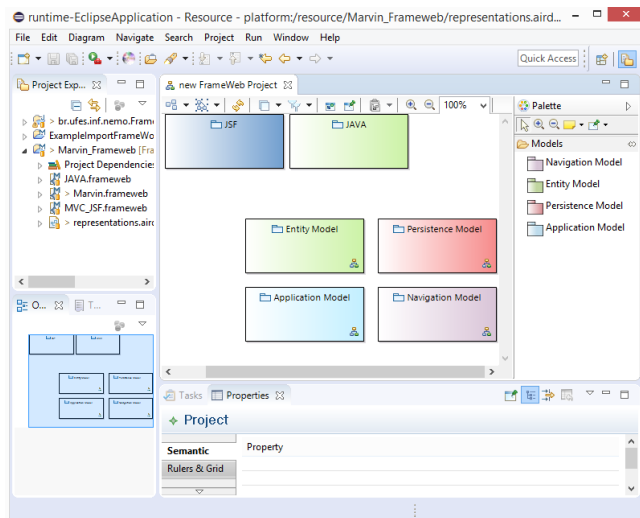


Figura 3: Visão geral de um WIS modelado com FrameWeb usando o editor.

Ainda na Figura 3, podemos analisar a interface principal da ferramenta, que possui no lado direito um painel as opções de componentes que podem ser criados no tipo de modelo em questão; na parte inferior uma lista de propriedades, onde podem ser alterados os atributos do componente selecionado; e no lado esquerdo o *Project Explorer*, para visualização do projeto. Um duplo-clique em um dos componentes de modelo na visão de projeto permite ao desenvolvedor construir o modelo em questão.

A Figura 4 mostra um fragmento de um Modelo de Navegação do *Marvin*,⁴ um sistema de apoio a atividades acadêmicas, utilizado como estudo de caso neste artigo. Seguindo a estrutura MVC, temos neste fragmento do modelo a página `index.xhtml`, composta de um formulário que envia os dados para o controlador `GenerateBibliographyController`, que por sua vez é associado à página `bibliography.xhtml` por meio do método `generate()` e novamente à página `index.xhtml` através do método `back()`. O modelo define, portanto, que dados são enviados das páginas Web às classes controladoras, quais os métodos destas classes são chamados e quais páginas são exibidas como resultado.

Neste estudo de caso, foi adicionado ao projeto um arquivo de definição de *framework* relacionado ao JSF, que permite a utilização de todas as *tags*, classes e propriedades definidas pelo *framework*. A Figura 4 mostra *tags* do tipo `inputNumber`, `selectOneMenu` dentre outras, que são próprias do JSF e especificadas neste arquivo.

Desta forma, o *FrameWeb Editor* visa auxiliar o desenvolvedor na criação de modelos válidos e adaptados aos *frameworks* e plataforma escolhidos. Destacamos como suas principais funcionalidades:

- Cada um dos quatro tipos básicos de modelos FrameWeb possui painéis de criação de componentes próprios com os componentes específicos do modelo em questão;
- Permite ao desenvolvedor apenas a criação de componentes e relações se esta for uma ação válida, garantindo a integridade do modelo;

⁴<https://github.com/dwvs-ufes/Marvin>

WebMedia'2017: Workshops e Pôsteres, WFA, Gramado, Brasil

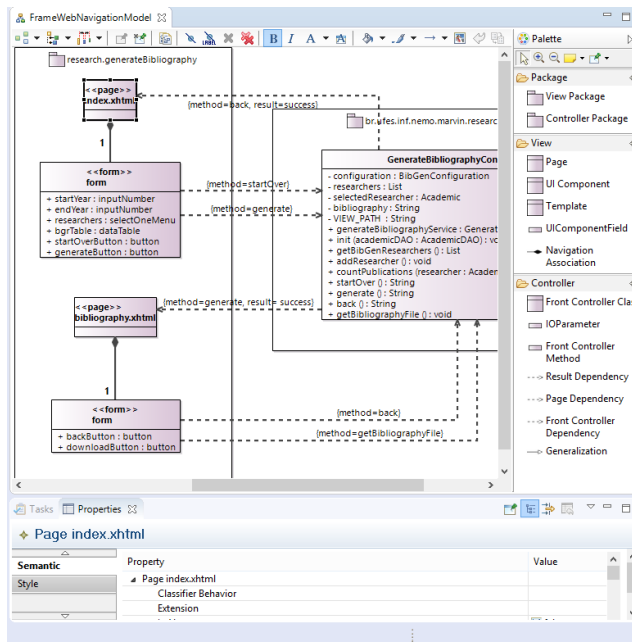


Figura 4: Diagrama de Navegação do Marvin modelado com o FrameWeb Editor.

- Relaciona os componentes gráficos diretamente com os componentes definidos na sintaxe da linguagem através dos meta-modelos, de forma que o modelo pode ser processado para a geração de código e outras funcionalidades análogas;
- Integra ao modelo de domínio conceitos inerentes a *linked data*, auxiliando na criação de WISs integrados à Web Semântica (segundo a abordagem FrameWeb-LD [6]);
- Oferece a possibilidade de importação de arquivos de definição de *frameworks*, plataforma de programação e vocabulário *linked data*, tornando fácil a integração e adaptação do modelo às tecnologias utilizadas;
- Possibilita a criação e atualização destes arquivos de definição na própria ferramenta, de forma a possibilitar que o desenvolvedor crie seus próprios arquivos caso os mesmos ainda não tenham sido definidos.

Os arquivos criados por meio da ferramenta são arquivos XML salvos no formato `.frameweb`, que podem ser processados posteriormente para diversas finalidades como, por exemplo, geração automática de código. Em um trabalho de mestrado em andamento está sendo desenvolvido um gerador próprio para a linguagem FrameWeb [2]. Também em [2], são relatados experimentos realizados com o editor na modelagem de WISs desenvolvidos por alunos de nossa universidade, no contexto de um curso sobre desenvolvimento Web.

A ferramenta encontra-se sob a licença MIT e o leitor interessado pode obter instruções detalhadas de instalação e uso na página do repositório do projeto, hospedada em <https://github.com/nemo-ufes/FrameWeb>.

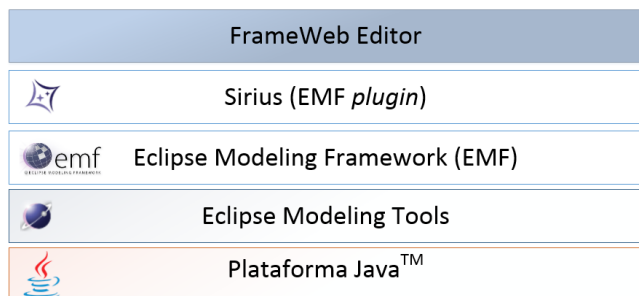


Figura 5: Arquitetura do *FrameWeb Editor*.

3.3 Arquitetura da Ferramenta

A arquitetura do *FrameWeb Editor* é ilustrada na Figura 5. A ferramenta foi desenvolvida com base na tecnologia Sirius [17], que oferece suporte à definição de representações gráficas de componentes e provê diversas funcionalidades comumente utilizadas em editores gráficos.

A construção de uma ferramenta de modelagem gráfica utilizando o *Sirius* passa pela definição de dois modelos relacionados ao domínio em questão (em nosso caso, *FrameWeb*). O modelo da sintaxe abstrata, ou meta-modelo, define os elementos que poderão ser criados no modelador gráfico, bem como suas propriedades e relações, e é especificada por meio do *Eclipse Modeling Framework (EMF)*, um *framework* de modelagem e geração de código para construção de ferramentas e outras aplicações com base em modelos estruturados de dados [8]. Tal meta-modelo foi inicialmente definido em [11, 12] e vem sendo refinado à medida do necessário. O modelo da sintaxe concreta, chamado pelo *Sirius* de *viewpoint specification*, define as características gráficas dos elementos definidos no meta-modelo. Ambos os modelos são especificados utilizando a sintaxe do *EMF*.

A Figura 6 mostra um fragmento da *viewpoint specification* definida no *Sirius*, em que o elemento gráfico *Page Node* e suas respectivas configurações de apresentação é associado à classe *Page* definida no meta-modelo e então utilizado no *FrameWeb Editor* como mostrado na Figura 4.

4 TRABALHOS RELACIONADOS

A seguir são apresentados alguns métodos que fazem uso de MDD no desenvolvimento de Sistemas de Informação Web e suas respectivas ferramentas, sendo apresentados com foco comparativo ao contexto de aplicação do método *FrameWeb* e da ferramenta apresentada neste artigo.

O método UWE (UML-based Web Engineering) [9] é um método MDD para desenvolvimento de aplicações Web e que possui a ferramenta CASE ArgoUWE,⁵ um plugin para a ferramenta ArgoUML⁶ em que os diagramas são construídos utilizando elementos de modelagem específicos para representar os conceitos introduzidos na metodologia UWE. Esse método assemelha-se ao *FrameWeb* quanto à separação da modelagem de conteúdo bem como a separação entre as partes independentes de plataforma e as dependentes. O

⁵<http://uwe.pst.ifi.lmu.de/toolargoUWE.html>

⁶<http://argouml.tigris.org/>

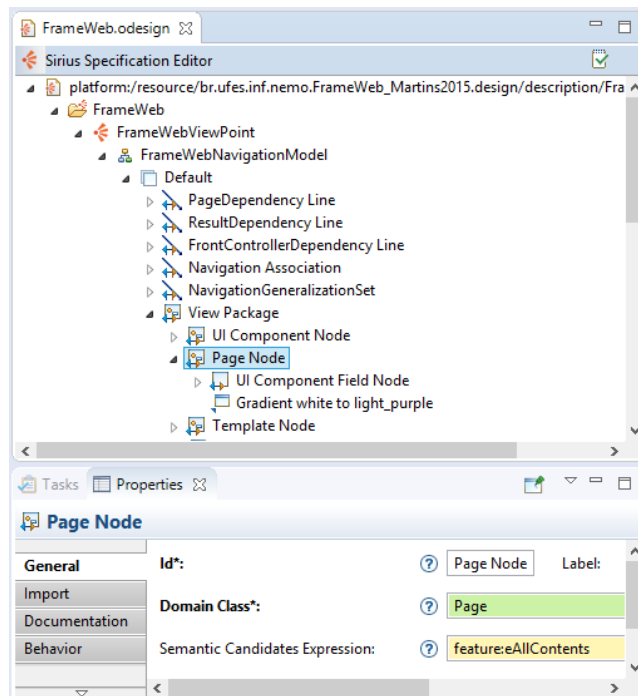


Figura 6: Fragmento do *FrameWeb Viewpoint Specification*, referente à classe *Page*.

grande diferencial se faz no fato do método não fazer menção aos conceitos e funcionalidades disponíveis com o uso de *frameworks*, que são incorporados aos modelos *FrameWeb* na ferramenta *FrameWeb Editor* por meio da importação dos arquivos de definições de *frameworks*, que podem ser criados e atualizados na própria ferramenta, permitindo a extensibilidade com relação à atualização e surgimento de novos *frameworks* disponíveis para utilização.

O RUX-Method [5] é um método para desenvolvimento de aplicações Web com características próprias para a criação de interfaces sensíveis a contexto. O método estrutura todo o ciclo de vida do desenvolvimento e oferece suporte à geração de código para diferentes dispositivos e plataformas. Este método também possui uma ferramenta implementada chamada RUX-Tool [10], que oferece suporte à modelagem e geração de código automática de interfaces com o usuário multiplataforma e para diferentes dispositivos. Por outro lado, assim como UWE, distancia-se do método *FrameWeb* e de seu editor principalmente com relação à extensibilidade, sendo muito dependente de sua biblioteca de componentes.

O padrão IFML [4] é uma linguagem visual independente de plataforma baseada em uma abordagem tradicional do padrão MVC, que traz aos modelos conceitos inerentes à navegabilidade das aplicações e operações que afetam seus conteúdos e estados de navegação considerando recursos integrados da *hypermedia*. O IFML possui uma ferramenta chamada IFML Editor,⁷ que possui a mesma arquitetura do *FrameWeb Editor*, sendo desenvolvida com base na tecnologia *Sirius* [17]. Porém como os demais métodos, não faz menção ou oferece funcionalidades referentes aos *frameworks*

⁷<http://ifml.github.io/>

FrameWeb Editor: Uma Ferramenta CASE para suporte ao Método FrameWeb

utilizados, abrangendo somente a parte independente de plataforma no desenvolvimento dos modelos.

O OOH4RIA [14] é um método baseado no *framework* da Google™ (Google Web Toolkit - GWT)⁸ para plataforma Java utilizando uma abordagem dirigida por modelos. O método é focado no desenvolvimento de RIAs (*Rich Internet Applications*) e possui a ferramenta OOH4RIA Tool [13], que permite representar os modelos e transformações para acelerar o processo de desenvolvimento de uma RIA implementada com o *framework* GWT. A principal limitação desta abordagem é que sob certo ponto de vista o fato de ser baseado no GWT enrijece o método, pois sua visão independente de plataforma não é de fato totalmente independente, já que está intimamente relacionada a aspectos da plataforma escolhida.

5 CONCLUSÕES

Em sua proposta original [1, 16], o método FrameWeb definia extensões leves (*lightweight extensions*) ao meta-modelo da UML para representar componentes típicos da plataforma Web e dos *frameworks* utilizados, sendo criado um perfil UML para a construção dos diagramas, de forma que os modelos eram criados utilizando-se qualquer editor UML.⁹ A principal limitação desta abordagem é o fato de o perfil UML não ser rigoroso, portanto as ferramentas de modelagem UML não possuem forma de prevenir que se incluam elementos que não pertencem aos modelos e ao fato dos componentes não estarem relacionados à sintaxe do método. Esta característica é apresentada no *FrameWeb Editor*, que por meio dos meta-modelos definidos em [11, 12], é capaz de gerar modelos FrameWeb válidos e relacionar os componentes gráficos diretamente com os componentes definidos na sintaxe da linguagem.

A ferramenta *FrameWeb Editor*, portanto, tem o intuito de suprir a necessidade do método por uma ferramenta CASE, permitindo que desenvolvedores criem modelos utilizando os elementos específicos da linguagem, de modo a oferecer métodos que auxiliem de fato desenvolvedores Web a produzirem WISs de qualidade, de forma eficiente e preparados para a Web Semântica.

Além de oferecer funcionalidades e ferramenta para a criação de modelos FrameWeb válidos e de dar suporte para extensão FrameWeb-LD [6], a ferramenta também serve de base para que sejam desenvolvidas outras funcionalidades. O *framework* EMF oferece plugins capazes não só de criar modelos, mas também de executar transformações de Modelo para Texto (*model to text - M2T*), transformações de Modelo para Modelo (*model to model - M2M*) e transformações de Texto para Texto (*text to text - T2T*), que podem ser utilizadas de forma integrada à ferramenta para a geração de código a partir dos modelos FrameWeb criados. Estamos atualmente trabalhando na integração do gerador de código FrameWeb [2] ao editor gráfico.

Além disso, o *FrameWeb Editor* traz ao usuário o potencial de extensibilidade característico ao método FrameWeb, dando suporte no editor à criação de arquivos de definição de *frameworks*, plataformas de programação e vocabulários *linked data* de forma gráfica, permitindo que usuários finais do editor integrem novos *frameworks*, plataformas e vocabulários ao editor e método FrameWeb.

⁸<http://www.gwtproject.org/>

⁹Uma lista de diversas ferramentas UML e suas respectivas plataformas, preços, companhias, dentre outras informações, pode ser encontrada em: http://www.objectsbydesign.com/tools/umltools_byCompany.html.

WebMedia'2017: Workshops e Pôsteres, WFA, Gramado, Brasil

Tem-se como trabalhos futuros a realização de avaliações e análises criteriosas de usabilidade da ferramenta, assim como experimentos comparativos com usuários usando a ferramenta proposta e usuários usando ferramentas CASE tradicionais no processo de projeto e desenvolvimento de sistemas de informação Web.

De modo geral, nossa perspectiva é que a ferramenta possa ser trabalhada para que evolua de protótipo para ferramenta comercial e ser usada por desenvolvedores Web na indústria, efetivamente realizando a transferência de tecnologia da academia.

ACKNOWLEDGMENTS

NEMO (<http://nemo.inf.ufes.br>) possui atualmente apoio da FAPES (# 0969/2015), do CNPq (# 485368/2013-7, # 461777/2014-2) e do FAP/UFES (# 6166/2015). Este trabalho foi realizado com bolsa do Programa Institucional de Iniciação Científica da UFES.

REFERÊNCIAS

- [1] Vitor E. S., Ricardo A. Falbo, and Giancarlo Guizzardi. 2009. Designing Web Information Systems for a Framework-based Construction. In *Innovations in Information Systems Modeling: Methods and Best Practices* (1 ed.), Terry Halpin, Eric Proper, and John Krogstie (Eds.). IGI Global, Chapter 11, 203–237.
- [2] Nilber Vittorazzi de Almeida, Silas Louzada Campos, and Vitor E. Silva Souza. 2017. A Model-Driven Approach for Code Generation for Web-based Information Systems built with Frameworks. In *Proceedings of the 23rd Brazilian Symposium on Multimedia and the Web*. ACM, (to appear).
- [3] Deepak Alur, John Crupi, and Dan Malks. 2003. *Core J2EE Patterns: Best Practices and Design Strategies* (2nd ed.). Prentice Hall / Sun Microsystems Press.
- [4] Luciano Baresi, Franca Garzotto, and Paolo Paolini. 2001. Extending UML for modeling web applications. In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*. IEEE, 10–pp.
- [5] Gaëlle Calvary, Joëlle Coutaz, David Thevenin, Quentin Limbourg, Laurent Bouillon, and Jean Vanderdonck. 2003. A unifying reference framework for multi-target user interfaces. *Interacting with computers* 15, 3 (2003), 289–308.
- [6] Danilo Ricardo Celino, Luana Vettler Reis, Beatriz Franco Martins, and Vitor E. S. Souza. 2016. A Framework-based Approach for the Integration of Web-based Information Systems on the Semantic Web. In *Proc. of the 22nd Brazilian Symposium on Multimedia and the Web*. ACM, 231–238.
- [7] Martin Fowler. 2002. *Patterns of Enterprise Application Architecture* (1 ed.). Addison-Wesley.
- [8] Richard C. Gronback. 2009. *Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit* (1 ed.). Addison-Wesley Professional.
- [9] Nora Koch, Alexander Knapp, Gefei Zhang, and Hubert Baumeister. 2008b. Web Engineering: Modelling and Implementing Web Applications, chap. UML-Based Web Engineering. In *UML-Based Web Engineering*. Springer London, London, UK.
- [10] Marino Linaje, Juan Carlos Preciado, Rober Morales-Chaparro, Roberto Rodriguez-Echeverria, and Fernando Sánchez-Figueroa. 2009. Automatic Generation of RIAs Using RUX-Tool and Webratio. In *Web Engineering: 9th International Conference, ICWE 2009, Proceedings*. Springer Berlin Heidelberg, San Sebastián, Spain, 501–504. https://doi.org/10.1007/978-3-642-02818-2_48
- [11] Beatriz Franco Martins. 2016. *Evolução do Método FrameWeb para o Projeto de Sistemas de Informação Web Utilizando uma Abordagem Dirigida a Modelos*. Master's thesis. Programa de Pós-Graduação em Informática, Universidade Federal do Espírito Santo.
- [12] Beatriz Franco Martins and Vitor E Silva Souza. 2015. A Model-Driven Approach for the Design of Web Information Systems based on Frameworks. In *Proceedings of the 21st Brazilian Symposium on Multimedia and the Web*. ACM, 41–48.
- [13] Santiago Meliá, José Javier Martínez Domene, Álvaro Pérez, and Jaime Gómez. 2009. OOH4RIA Tool: Una Herramienta basada en el Desarrollo Dirigido por Modelos para las RIAs.. In *JISBD*. 219–222.
- [14] Santiago Meliá, Jaime Gómez, Sandy Pérez, and Oscar Díaz. 2008. A model-driven development for GWT-based rich internet applications with OOH4RIA. In *Web Engineering, 2008. ICWE'08. Eighth International Conference on*. IEEE, 13–23.
- [15] Oscar Pastor, Sergio España, José Ignacio Panach, and Nathalie Aquino. 2008. Model-driven development. *Informatik-Spektrum* 31 (2008), 394–407.
- [16] Vitor E Silva Souza. 2007. *FrameWeb: um Método baseado em Frameworks para o Projeto de Sistemas de Informação Web*. Master's thesis. Programa de Pós-Graduação em Informática, Universidade Federal do Espírito Santo.
- [17] Vladimir Viyovic, Milan Maksimovic, and Branko Perisic. 2014. Sirius: A rapid development of DSM graphical editor. In *Intelligent Engineering Systems (INES), 2014 18th International Conference on*. IEEE, 233–238.