

Estudo Exploratório no Refinamento de uma DSL para Versões Baseadas em EMF, EMF Forms e Angular

Esther Salgado Favero, Igor Ademilson de Oliveira,
Pedro Sebastian Zanella Nuñez, Fábio Paulo Basso

¹Universidade Federal do Pampa (UNIPAMPA)
Engenharia de Software
Alegrete, Rio Grande do Sul, Brasil
{estherfavero97@gmail.com, igornnt@gmail.com ,
pedrosebastian90@gmail.com, fabiobasso@unipampa.edu.br}

Abstract. *This work presents an exploratory study of technologies for building Domain Specific Languages (DSLs) including EMF, EMF Forms and Angular. Our goal is to explore three versions of a DSL designed to represent business models created for the validation of startup models. Our domain model is conceptually directed to represent business plans for all areas of the labor market. This research performs a "black box" analysis, investigating in practice these technologies as a way of finding the best option for generation of a business plan management information system. Our conclusions are: 1) The development of systems based on DSL concepts requires elevated knowledge for domain analysis; 2) the tools investigated bring a facility for the integration of each technology, and; 3) Visual results are not pleasing to an end user, thus opening window for new research for usability.*

Resumo. *Este trabalho apresenta um estudo exploratório de tecnologias para a construção de Linguagens Específicas de Domínio (DSL), incluindo EMF, EMF Forms e Angular. Nosso objetivo é explorar três versões de uma DSL concebida para representação de modelos de negócio criados para a validação de modelos de startups. Nosso modelo de domínio é conceitualmente direcionado para representação de planos de negócios para todas as áreas do mercado de trabalho. Esta pesquisa realiza uma análise "black box", investigando na prática estas tecnologias como forma de encontrar a melhor opção para um sistema de informação gerencial de planos de negócio. Nossas conclusões são: 1) O desenvolvimento de sistemas com base em conceitos de DSL requer um nível elevado de conhecimento para análise de domínio; 2) as ferramentas investigadas trazem uma facilidade para a integração de cada tecnologia, e; 3) os resultados visuais não são agradáveis para um usuário final, portanto abrindo espaço para novas pesquisas em usabilidade.*

1. Introdução

Sabe-se que “startup” é o termo utilizado para definir uma empresa recém criada, que apresenta um modelo de negócios considerado inovador e que ainda não fora validado totalmente junto ao mercado (Giardino et al. 2014). Empresas novas no mercado tendem a ter inseguranças a respeito de produtos, serviços ou afins que a mesma desenvolve. Este

cenário de incerteza significa dizer que não há uma garantia de que a ideia ou o projeto da empresa dará certo.

Modelos de planos de negócio servem para guiar uma startup no desenvolvimento e gerenciamento de seu negócio. Voltados para a tomada de decisão, tais modelos são caracterizados por Sistemas de Informação Gerenciais (SIG) (Giardino et al. 2014), que são munidos de bancos de dados e estruturas para cadastro e busca de informações. Eles também são caracterizados por cenários dinâmicos cujos dados tendem à sofrer mudanças conforme a inovação é confrontada perante o tempo de mercado (Basso et al. 2015). Tendo isto em vista, é recomendada a utilização de SIGs para a elaboração e gestão de modelos para plano de negócios.

Usando-se de geração automática de código e refinamentos (Basso et al. 2016), SIGs podem ser produzidos por meio de técnicas e ferramentas para Prototipação Rápida de Aplicações (RAP). Este trabalho discute sobre a geração e refinamento de três diferentes versões de um SIG, cujo modelo de domínio é focado em negócios. Para este domínio, é recomendável o uso de características de usabilidade como padrões de design (Welie 2018). Ou seja, levando-se em conta que o SIG deve oferecer aos empreendedores a possibilidade de criação de um modelo de negócio genérico que abranja todas as áreas de mercado de trabalho, investiga-se as características de usabilidade oferecidas por três ferramentas de RAP.

Nossa contribuição é uma descrição de benefícios e limitações observadas dessas ferramentas de RAP e que suportam a Engenharia Dirigida a Modelos (MDE) (Brambilla and Fraternali 2014). Tais ferramentas tem o intuito de utilizar modelos em diferentes níveis de abstração para a construção de Linguagens Específicas de Domínio (DSL) (Voelter 2009). Estes modelos são gerados automaticamente com transformadores de modelos (Basso et al. 2017) e refinados manualmente para aderir com alguns padrões de design (Welie 2018).

Este artigo é organizado da seguinte forma: na Seção 2 está disposta a Metodologia, discutindo a abordagem utilizada para o problema proposto. Na seção 3 apresenta-se as principais características da DSL proposta. A Seção 4 relata a execução do estudo. A discussão que sumariza nossos achados é apresentada na Seção 5 e a Seção 6 expõe as lições aprendidas e respostas às questões de pesquisa.

2. Metodologia

Como parte de uma disciplina optativa de nosso curso de Engenharia de Software, denominada "Engenharia Dirigida por Modelos", explorou-se o desenvolvimento deste sistema sob a forma de uma DSL construída sobre o Eclipse Modeling Framework (EMF) (Steinberg et al. 2008). DSLs são comumente conhecidas na literatura como formas de se representar uma arquitetura de software usando-se de diagramas (Kelly and Tolvanen 2008). No entanto, para o domínio do problema proposto, a DSL deve ser construída sob a forma de um sistema de informação web (Brambilla and Fraternali 2014), como por exemplo, composta de um conjunto de telas que permitam a realização de tarefas do tipo CRUD (Create, Read, Update e Delete) e outras estruturas disponíveis em SIGs (Welie 2018).

Neste sentido, investigou-se alguns atributos de qualidade associados com algumas tecnologias de MDE que afetam a criação deste sistema como uma DSL. Buscamos

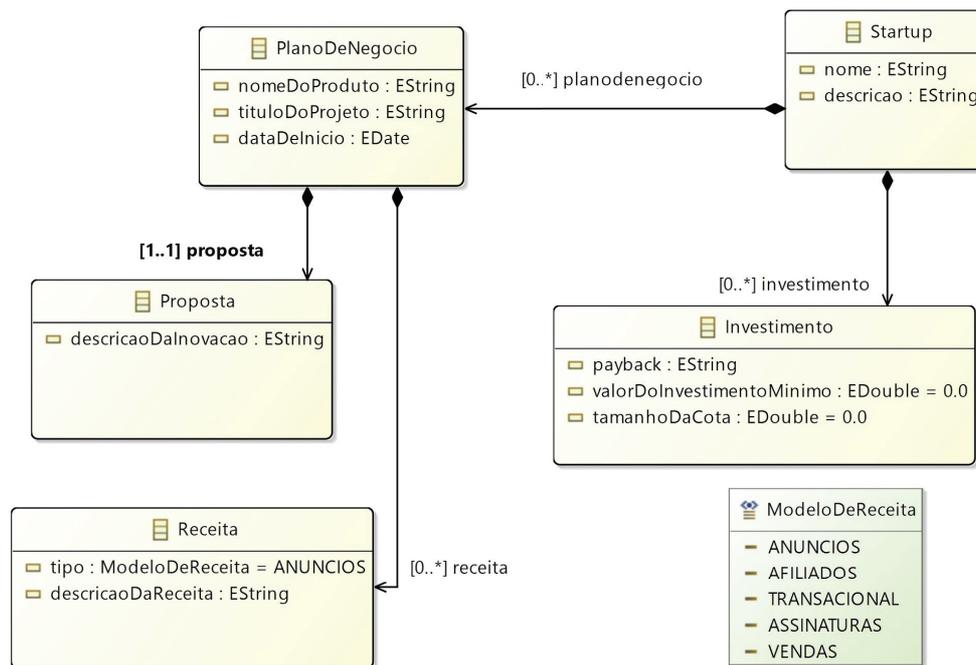


Figura 1. Primeira parte do diagrama de classes da DSL proposta

assim conduzir uma pesquisa exploratória do tipo "black-box" para responder às seguintes questões: 1) Como usuários inexperientes de tecnologias para MDE, qual é a nossa percepção sobre as dificuldades na geração de uma DSL com a tecnologia EMF? 2) Uma vez que a DSL precisa ter o front-end semelhante aos sistemas de informações gerenciais, qual é a nossa percepção sobre as dificuldades na adaptação desta DSL com a tecnologia EMF Forms? e 3) Já que a DSL precisa ter a sua execução na web, atendendo um critério definido internamente como requisito não funcional desejável desse sistema, qual é a nossa percepção sobre as dificuldades na adaptação desta DSL com tecnologias para javascript como Angular 5?

3. Principais Características da DSL Proposta

O modelo de negócio proposto faz o uso do MDE com sua respectiva DSL, à qual é responsável por especificar o modelo de domínio através do EMF (Steinberg et al. 2008), uma ferramenta para metamodelagem. Neste mapeamento, foram selecionados artigos que tratavam de assuntos relacionados ao modelo de negócio. A partir deles, se embasou a construção do metamodelo parcialmente apresentado na Figura 1.

Visto que o trabalho trata da área "business", a representação do modelo de domínio (metamodelo) também foi inspirado em uma ferramenta já existente, o MakeMoney <makemoney.starta.com.br/>. O MakeMoney é um software que permite uma empresa testar suas idéias de produto, analisar a possibilidade de inovação e aumentar sua competitividade no mercado.

4. Relato do Estudo Exploratório

Com base nestas questões de pesquisa, a construção da DSL foi feita utilizando-se de técnicas para MDE disponíveis com suporte ferramental na distribuição Eclipse Modelling

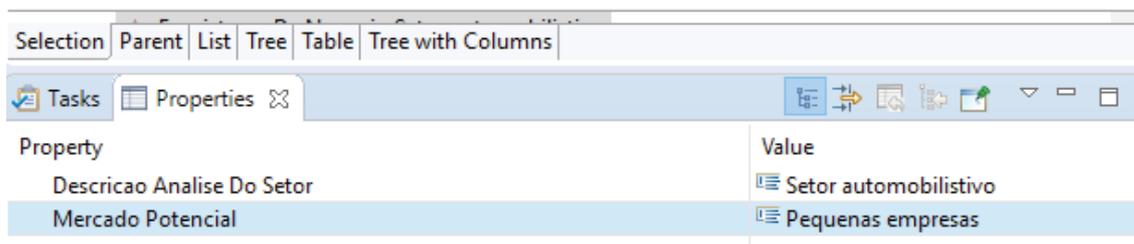


Figura 2. DSL Versão 1 como um plugin gerado pelo EMF

Tools (EMT) (Steinberg et al. 2008). O EMT é uma ferramenta que dispõe do modelador gráfico Ecore, isto é, um diagrama de classes utilizado na construção de metamodelos. Além disso, este é um ecossistema de ferramentas integradas de MDE, cujos plugins são usados para a geração de código Java para aplicativos construídos sobre o guarda-chuva do EMF.

Usando um ecossistema maduro para a criação de DSLs, nosso desafio de pesquisa resumiu-se à explorar diferentes tecnologias/plugins para a geração de três diferentes versões da DSL proposta, buscando aquela que melhor atende aos requisitos do sistema proposto. Além disso, no MDE, o aumento ou redução do nível de abstração é realizado através de transformações dos modelos que representam as versões da nossa DSL. Quer dizer, o modelo da DSL proposta, originalmente em nível independente de uma plataforma web, deve ser transformado em modelos de nível dependente de plataforma de execução web.

4.1. Geração da Versão 1

Utilizando o EMF, gerou-se o modelo de domínio mostrado na Figura 1. Então, transformou-se esta representação para uma DSL ilustrada na Figura 2. Esta figura apresenta o editor disponível no EMF para especificar os dados de um modelo de negócios. Além deste formulário, o EMF também disponibiliza uma árvore de elementos para o usuário final interagir sobre elementos do plano de negócio. Claramente, o resultado final não é amigável e não nos agradou, o que nos impulsionou à procurar por alternativas para agregar à esta DSL uma melhor usabilidade.

4.2. Transformações de Refinamentos da Versão 2

Em uma perspectiva de melhorar os aspectos visuais da DSL na Versão 1, foi utilizado o EMF Forms <www.eclipse.org/ecp/emfforms/> para a geração de uma Versão 2. O plugin do eclipse EMF Forms é uma ferramenta desenvolvida para a criação de interfaces de usuário para modelos EMF, mas cuja representação é baseada em formulários. Nesta pesquisa, encontramos um plugin do EMT que possibilita a geração do modelo conforme o metamodelo EMF Forms de maneira automática. Assim, o passo seguinte foi a utilização da ferramenta de editoração do EMF Forms, que permite que as interfaces sejam customizadas de uma maneira simples, sem codificação. Além disso, este aparato ferramental proporciona aparência e comportamento uniformes para operações do tipo CRUD. Portanto, o resultado final da versão 2 de nossa DSL é mostrado na Figura 3.

4.3. Estudo Black-box de Plugins Eclipse Para Web Front-End

Para se chegar até a tecnologia para criação dos formulários para a web foi realizada uma simples busca na internet utilizando o Google com ferramenta de pesquisa. Os cinco

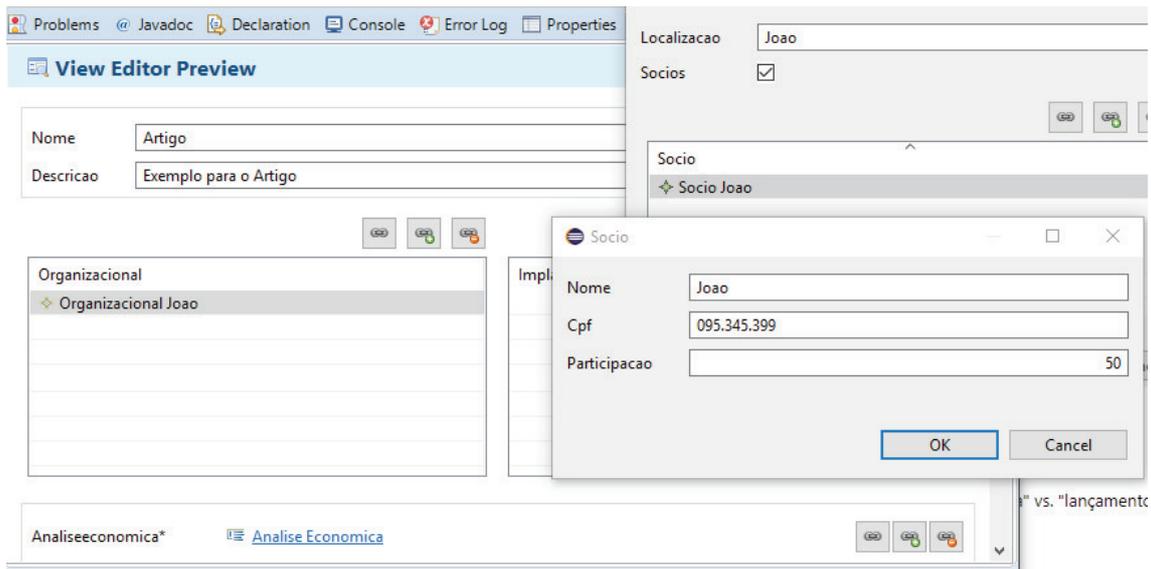


Figura 3. DSL Versão 2 como formulários gerados com base no EMF Forms

Tabela 1. Plugins Eclipse para a Execução de uma DSL em Ambiente Web

Plugin	URL do Plugin	Tipo
JSON Forms	https://eclipsesource.com/blogs/tutorials/emf-forms-and-json-forms-integration-guide/	SI & CRUD
Web Modeling Framework	https://projects.eclipse.org/proposals/web-modeling-framework	Diagrama
EMF REST	https://modeling-languages.com/emf-rest-restful-apis-from-models/	Diagrama
EMF Parsley	http://www.rcp-vision.com/emf-parsley-a-web-application-is-a-few-steps/	Árvore EMF
MPS da jetbrains	https://www.jetbrains.com/mps/	DSL Textual

resultados encontrados são apresentados na Tabela 1, que caracteriza cada opção para diferentes tipos de interação do usuário final com a DSL.

Através dessa análise black-box, chegou-se até a ferramenta JSON Forms, que é um framework que possibilita a geração de interfaces para Sistemas de Informação (SI) web baseadas em formulários. Tal framework utiliza arquivos em formato JSON para representar o esquema da interface e a estrutura da visão (view). Através da pesquisa foi descoberta a possibilidade de renderizar esses formulários utilizando frameworks e bibliotecas web de uma forma mais simples através de tags html personalizadas (web components).

Assim, selecionou-se o framework JSON Forms para o desenvolvimento da terceira versão da DSL proposta. Esses componentes web processam arquivos do JSON Forms e renderizam um formulário com estilo CSS, executando algumas validações. Por fim, as tecnologias encontradas na execução da DSL na web são: AngularJs - utilizando a biblioteca, Angular Schema Form; Angular 5 - com a biblioteca Angular JSON Schema Form; e o React.js - com a biblioteca react-jsonschema-form.

4.4. Transformações e Refinamentos da Versão 3

Por meio de um plugin disponível no EMT, foi possível gerar arquivos JSON (JSON Schemas), juntamente com as Views (UI Schemas) respectivas. Também gerou-se arqui-

vos com schemas JSON tendo como entrada os modelos conforme o EMF Forms. Como resultado, obteve-se os esquemas de formulário e de estrutura de view com base em JSON. Posteriormente, através desses arquivos JSON gerados à partir das View Forms da aplicação, foi realizado o refinamento manual dos artefatos gerados, criando a aplicação final com o framework de desenvolvimento de aplicações e interfaces para web Angular, como ilustrado na Figura 4.

Com o intuito de se realizar uma avaliação prática para se verificar a viabilidade de se construir interfaces web para o usuário, optamos por refinar o resultado final com o Angular 5 e suas tecnologias relacionadas. O motivo é que o mesmo apresenta-se como um framework robusto para a criação de aplicações corporativas e com tecnologias atuais. Assim, o AngularJs foi desconsiderado, pois foi descontinuado pelos mantenedores em detrimento do novo Angular. A biblioteca React é também uma biblioteca moderna, sendo hoje a mais utilizada do mercado para dentre as demais concorrentes, porém seu foco não é para criação de aplicações comerciais, mas sim para criação de interfaces.

4.5. Resultado Final

Como lição aprendida, observamos que essas tecnologias integradas trazem a possibilidade de se criar de forma rápida formulários sem que o desenvolvedor invista muito tempo na tarefa de programação. Com uma simples tag html e alguns atributos é possível renderizar na tela formulários razoavelmente complexos. Um exemplo da utilização do componente web para renderização é mostrado na Figura 4.

Angular JSON Schema Form Material Design App

Análise Econômica:
Data De Inicio
dd/mm/aaaa --:--

Equipe
Nome
Funcao
Custo

Add to Equipe

Faturamento
Data Do Recurso
dd/mm/aaaa --:--
Valor

Figura 4. DSL Versão 3 como um formulário web gerado com base em Angular e JSON Forms

5. Discussões

Com base no trabalho realizado, observa-se a facilidade com que se pode construir uma DSL. Entretanto, várias questões no que tange a qualidade de software ainda não foram bem abordadas, como a questão da experiência do usuário e, talvez, questões de arquitetura e integração entre sistemas diversos. As aplicações geradas através do EMF Forms são difíceis de serem utilizadas em cenários reais e a solução para gerar formulários web, nesse momento, apenas pode dar uma produtividade maior na construção de parcelas de um determinado sistema.

Porém, mesmo essa questão precisa ser bem avaliada. A tarefa de criação de simples formulários web normalmente é delegada a desenvolvedores front-end, que geralmente não tem conhecimento sobre criação e modelagem de domínio, essencial para a criação de DSLs. Em um contexto empresarial, a tarefa de criação de formulários com base no EMF Forms exigiria algum treinamento, ou que exista um profissional capacitado e especializado, o que talvez não seja viável para projetos pequenos.

Esperava-se, a princípio, que para a execução de nosso estudo, já existissem bibliotecas, frameworks ou softwares capazes de gerar sistemas web com base em EMF, com pouca ou nenhuma programação. Para levar nossa DSL da versão 2 para a versão 3, muito esforço manual foi necessário. Assim, existe um gap de pesquisa que merece ser investigado na transformação automática de modelos EMF Forms para ambiente web. Também, é preciso uma forma de representar os formulários mais ricos do que os formatos existentes no EMF Forms, como os padrões de projeto de desenho e interação (Welie 2018).

Essa pilha de tecnologias necessárias para a execução da DSL na web também é um desafio de pesquisa. É importante que se gere o código completo para que as tarefas de refinamento não sejam tão demoradas. Esta pilha de tecnologias do EMT auxilia times na realização de tarefas mais triviais, poupando esforços e recursos para funcionalidades chave da DSL. No entanto, é preciso a geração de interfaces com maior usabilidade.

Outra questão importante é que realmente não existem ferramentas de fácil utilização para modelagem de negócios e validação de startups no mercado. Porém, por ser um problema recorrente para a maioria dos empreendedores, a criação de tal sistema seria um bom investimento. Por ser uma área bem conhecida e estruturada, criá-la como uma DSL torna-se completamente viável, necessitando de um time de bons engenheiros e de especialistas de domínio, além de melhorias nas pesquisas para a transformação para diferentes versões de DSLs rodando em diferentes plataformas.

6. Conclusão

Concluimos, com base num estudo exploratório, que a geração e refinamento de sistemas de informações, baseados em DSLs para formulários, pode ser um grande desafio em termos de engenharia. O sucesso para desenvolvimento das diferentes versões da DSL explorada exigem profissionais treinados e uma grande proximidade com especialistas de domínio. No entanto, isto caracteriza um grande potencial de pesquisa e inovação para fomentar o crescimento neste mercado. O motivo é que as tecnologias estudadas trazem a possibilidade de se resolver problemas específicos e com real foco no domínio.

As tecnologias investigadas são devotadas para um dos grandes desafios das tecnologias utilizadas atualmente no mercado: o desenvolvimento rápido de sistemas orien-

tados à objetos. Assim, respondendo as questões de pesquisa voltadas para este desafio, pode-se destacar que: 1) Apesar das facilidades trazidas pelo EMT e pelo EMF, ainda não há um bom suporte para a construção de soluções baseadas na diversidade de padrões de design e interação em (Welie 2018); 2) Neste sentido, as interfaces de usuário geradas e refinadas com EMF Forms não são visualmente agradáveis e tem pouco foco em questões como usabilidade e experiência de uso e, o mais importante; 3) Por fim, as tecnologias disponíveis não possibilitam de forma eficiente e robusta a geração de interfaces web para javascript, o que em caso contrário motivaria uma maior utilização para tais recursos.

Como um trabalho futuro, pretende-se desenvolver uma alternativa para as transformações de modelos EMF para ambientes web. Para tanto, no lugar de EMF Forms, pretende-se utilizar a ferramenta MockupToME (Basso et al. 2016), que permite elaborar formulários com uma interação mais rica, e o desenvolvimento de transformações customizáveis para múltiplas combinações de tecnologias (Basso et al. 2017).

Por fim, acredita-se que a execução deste trabalho, além de nos proporcionar uma visão panorâmica do estado da arte disponível no ecossistema do Eclipse para geração de DSLs para SIGs, também abre novos estudos visando migrar automaticamente as DSLs desenvolvidas em nosso grupo de pesquisa que possuem esta mesma característica. Destaca-se o RAS++, uma DSL para a representação de ativos para negócios envolvendo a transferência de tecnologia em cenários de coopetição na Engenharia de Software.

Referências

- [Basso et al. 2017] Basso, F. P., Oliveira, T. C., Werner, C. M., and Becker, L. B. (2017). Building the foundations for ‘mde as service’. *IET Software*, 11:195–206.
- [Basso et al. 2016] Basso, F. P., Pillat, R. M., Oliveira, T. C., Roos-Frantz, F., and Frantz, R. Z. (2016). Automated design of multi-layered web information systems. *Journal of Systems and Software*, 117:612 – 637.
- [Basso et al. 2015] Basso, F. P., Pillat, R. M., Roos-Frantz, F., and Frantz, R. Z. (2015). Combining mde and scrum on the rapid prototyping of web information systems. *International Journal of Web Engineering and Technology*, 10(3):214–244.
- [Brambilla and Fraternali 2014] Brambilla, M. and Fraternali, P. (2014). Large-scale model-driven engineering of web user interaction: The webml and webratio experience. *Science of Computer Programming*, 89, Part B:71 – 87. Special issue on Success Stories in Model Driven Engineering.
- [Giardino et al. 2014] Giardino, C., Unterkalmsteiner, M., Paternoster, N., Gorschek, T., and Abrahamsson, P. (2014). What do we know about software development in startups? *Software, IEEE*, 31(5):28–32.
- [Kelly and Tolvanen 2008] Kelly, S. and Tolvanen, J.-P. (2008). *Domain Specific Modeling: Enabling Full Code Generation*. IEEE Computer Society - John Wiley & Sons.
- [Steinberg et al. 2008] Steinberg, D., Budinsky, F., Paternostro, M., and Merks, E. (2008). *EMF: Eclipse Modeling Framework (2nd Edition)*. Addison-Wesley Professional.
- [Voelter 2009] Voelter, M. (2009). Best practices for dsls and model-driven development. *Journal of Object Technology*, 8(6):79–102.
- [Welie 2018] Welie (2018). Patterns and interaction design <<http://www.welie.com/patterns/>>. Acessado em: 10 de Julho de 2018.