

Usa-DSL: Um Framework para Avaliação de Usabilidade de Linguagens Específicas de Domínio*

Ildevana Poltronieri Rodrigues¹, Márcia de Borba Campos¹, Avelino F. Zorzo¹,
Maicon Bernardino², Elder de Macedo Rodrigues²

¹Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Av. Ipiranga, 6681, Partenon – Porto Alegre – RS – Brasil

²Universidade Federal do Pampa (UNIPAMPA)
Laboratory of Empirical Studies in Software Engineering (LESSE)
Av. Tiarajú, 810, Ibirapuitã – Alegrete, RS – Brasil

ildevana@gmail.com, marciabcampos@hotmail.com, avelino.zorzo@pucrs.br
bernardino@acm.org, elderrodrigues@unipampa.edu.br

Abstract. *The lack of well-defined methods to assess usability criteria for Domain-Specific Languages (DSL) is a problem for DSL developers. Therefore, software engineers usually evaluate usability in an ad-hoc and subjective manner. In this paper, we present the first version of a DSL usability assessment framework, called Usa-DSL. We also present a preliminary evaluation of this framework.*

Resumo. *Estudos mostraram não haver métodos bem definidos para avaliar critérios de usabilidade em linguagens específicas de domínio (Domain-Specific Languages - DSL). Desta forma, engenheiros de software avaliam de forma experimental, até mesmo subjetivamente, se uma DSL é fácil de ser utilizada e de fácil entendimento, por exemplo. Este trabalho descreve a primeira versão de um framework de avaliação de usabilidade de DSL denominado Usa-DSL. Este artigo também apresenta uma avaliação preliminar do Usa-DSL.*

1. Introdução

Em desenvolvimento de software geralmente Linguagens de Propósito Geral (*General-Purpose Languages - GPL*) são utilizadas, por exemplo, Java, C#, Ruby, Python, dentre outras. Porém, existem algumas aplicações de domínios específicos que se beneficiam de linguagens com características específicas, que podem gerar maior desempenho, representações diferenciadas, abstrações no domínio do negócio, melhor comunicação entre os desenvolvedores e os analistas de negócio, entre outros aspectos. Para isso, os engenheiros de sistemas buscam, por meio do desenvolvimento de diferentes linguagens, facilitar cada vez mais a concepção de novos conceitos e teorias. Neste sentido, se verifica a existência de linguagens de programação que, aliadas à complexidade do desenvolvimento de aplicações, podem apresentar certas dificuldades às empresas no que se refere à modelagem, implementação, avaliação e manutenção de sistemas, e, com isso, causar diferentes problemas e comprometer a qualidade do projeto dos mesmos.

*Agradecimentos: DELL e CAPES pela concessão de bolsas de mestrado e doutorado.

Sendo assim, para minimizar dificuldades inerentes ao desenvolvimento de aplicações, destaca-se a possibilidade de uso do desenvolvimento dirigido por modelos (*Model-Driven Development* - MDD) [Mernik et al. 2005], que, a partir de modelos do sistema, permite a geração automática de códigos. Em específico para este artigo, a abordagem de MDD está baseada em Linguagens Específicas de Domínio (*Domain-Specific Languages* - DSLs) [Fowler 2010]. Pode-se destacar que tem se investido em diferentes propósitos para o desenvolvimento de DSL, que variam conforme o domínio da aplicação. Por exemplo, existem DSLs que podem ser aplicadas nos domínios de robótica [Conrado 2012], anomalias em arquitetura de software [Albuquerque et al. 2015] e teste de desempenho [Bernardino et al. 2016], entre outros. O diferencial entre as DSLs é definido pela sintaxe e semântica, que são determinadas pelo domínio do problema a ser resolvido. Ainda, um domínio pode utilizar mais de uma DSL para modelar as suas características, sem haver sobreposição.

Desta forma, acredita-se que o atendimento a recomendações de usabilidade em uma DSL possa aumentar seu uso no desenvolvimento de sistemas. Para que os usuários de domínios específicos possam utilizar estas linguagens de forma mais independente e com facilidade, os engenheiros de linguagens precisam preocupar-se com critérios de usabilidade e ainda com a diversidade de contextos e domínios, levando em conta a satisfação no uso durante a interação destes usuários e os diferentes sistemas. Sendo assim, de posse de diferentes conceitos sobre DSL, Teste de Desempenho, Estudos Empíricos em Engenharia de Software e Avaliação de Usabilidade, este artigo visa apresentar o *Framework Usa-DSL para Avaliação de Usabilidade de Linguagens Específicas de Domínio*. Também apresenta a metodologia e resultados de uma avaliação preliminar, que foi realizada utilizando a técnica de Entrevista.

Este artigo está organizado da seguinte forma: na Seção 2 são apresentados trabalhos relacionados referentes a DSLs e os métodos e técnicas aplicadas na avaliação de usabilidade no contexto de DSL. A Seção 3 descreve o *Framework Usa-DSL*, na qual será apresentada sua estrutura que é composta por etapas, fases e atividades. A Seção 4 apresenta uma avaliação preliminar realizada por meio de uma entrevista. Finalmente, a Seção 5 apresenta a conclusão e os trabalhos futuros.

2. Trabalhos Relacionados

Com o propósito de embasar conceitos e definições utilizadas no desenvolvimento do *Framework Usa-DSL* e unificar o entendimento dos tópicos relacionados ao estudo, são apresentados alguns trabalhos identificados na revisão sistemática [Rodrigues et al. 2017]. Estes artigos estão diretamente relacionados ao processo de avaliação de DSL e consideram questões referentes à usabilidade, bem como técnicas e abordagens de usabilidade que foram utilizadas nas avaliações.

O trabalho de Barisic *et al.* [Barisic et al. 2014] considera fundamental avaliar critérios de qualidade de uso no desenvolvimento de uma DSL, para que esta possa melhor atender as necessidades do usuário final. O estudo indica que a avaliação de critérios de usabilidade deve ser aplicada durante todas as etapas do processo de concepção, tais como: projeto, análise, desenvolvimento e avaliação de uma DSL. Para melhor atender o usuário final de uma DSL, Barisic *et al.* [Barisic et al. 2014] sugere que a avaliação de Interação Humano Computador (IHC) inclua métodos quantitativos e qualitativos, tais

como: observação do usuário e entrevistas.

Com relação a importância dada a usabilidade de DSL, o estudo de Albuquerque [Albuquerque et al. 2015] destaca a preocupação com a realização de uma análise quantitativa no que se refere a aspectos de usabilidade. Para isso, realizaram uma instanciamento do *Cognitive Dimensions of Notations framework* (CDN), proposto por Blackwell and Green em 2003. Este *framework* foi desenvolvido para discutir o desenvolvimento e avaliação de ferramentas por *designers*, já a sua instância CDN foi desenvolvida para detectar aspectos de usabilidade em artefatos de manutenção de tarefas em DSL. O artigo ainda relata que uma DSL compreende quatro aspectos importantes: expressividade, concisão, integração e desempenho.

Do mesmo modo, Barisic *et al.* [Barisic et al. 2011] define um modelo cognitivo para a linguagem baseado no contexto de cenários de usuários. As atividades cognitivas envolvidas na linguagem são: aprendizado da sintaxe e semântica, composição da sintaxe necessária para desempenhar uma função, a compreensão da sintaxe desenvolvida por outra pessoa, depuração da sintaxe e a modificação de uma função escrita pelo próprio desenvolvedor ou por outro. Referente a avaliação desta DSL foi realizado um estudo de caso com 6 participantes, que utilizaram a DSL Pheasant (PHysicist's EAasy Analysis Tool) e uma *baseline* GPL em C++. Ainda com relação ao método de avaliação, Teruel *et al.* [Teruel et al. 2014], como os autores Barisic *et al.* [Barisic et al. 2012] e outros apresenta a condução de um experimento para avaliar a usabilidade, no caso de Teruel o domínio da linguagem se refere a modelagem colaborativa de requisitos de sistema.

A avaliação de usabilidade no estudo de Sinha e Smidts [Sinha and Smidts 2006] está baseado em quatro das dez heurísticas propostas por Nielsen, sendo que para cada uma das heurísticas avaliadas foram definidas métricas. Sendo estas: aprendizado, eficiência, índice de erros, satisfação e facilidade. No final, a avaliação quantitativa considerou cada uma das métricas definidas para avaliar a DSL.

Quanto às técnicas e métodos, estudos apresentam a adaptação ou utilização de conjuntos de métricas de usabilidade. Apesar dos esforços, ainda é necessário um grande esforço para conceber uma DSL de maneira fácil, com maior compreensão e expressividade em relação ao domínio que se pretende representar. Além disto, é necessário também desenvolver processos, métodos e técnicas que auxiliem em uma melhor avaliação de usabilidade de DSLs.

3. Usa-DSL: *Framework* para Avaliação de Usabilidade de DSL

Pensando na necessidade de avaliar linguagens para domínio específicos, buscou-se na literatura por meio da execução de uma Revisão Sistemática da Literatura - RSL [Rodrigues et al. 2017] e da consulta de bibliografias consolidadas na área de IHC [Rogers et al. 2013] [Barbosa and da Silva 2010] [Stone et al. 2005] [Nielsen 1993], diferentes métodos e *frameworks* que realizassem avaliação de usabilidade. Na RSL [Rodrigues et al. 2017] foram identificado diferentes estudos, que discutem e apresentam preocupações relacionadas a aspectos de usabilidade e destacam a importância deste tipo de avaliação durante o projeto, análise e desenvolvimento das DSLs, porém nenhum *framework* ou método para realização avaliação de usabilidade foi apresentado. Sendo assim, este trabalho propõe um *framework* para avaliação de usabilidade de DSL denominado Usa-DSL, que é constituído por etapas, fases e atividades, conforme descrito

nas próximas subseções.

3.1. Estrutura do *Framework* Usa-DSL

A estrutura do *framework* Usa-DSL é um legado do processo de ciclo de vida de projeto [Stone et al. 2005]. Está organizada em etapas, fases e atividades, como pode ser visto na Figura 3.1. As etapas estão definidas em: Perfil do Avaliador, Responsabilidades Éticas e Legais, Tipo de Dados, Métodos Empíricos (ES), Método de Avaliação (IHC), Métricas, Instrumentos de Coleta, Instrumentos de Instrução e Treinamento, Condução da Avaliação, Empacotamento dos Dados e Relatório da Avaliação. Já as fases são compostas por: Definição, Execução, Análise e Resultados. As atividades são formadas por um conjunto de 32 conceitos que são distribuídas entre as etapas e fases, e estão dispostas no *link* http://tiny.cc/ERES2017_Usa-DSL.

A estrutura do *framework* Usa-DSL foi planejada para ser utilizada a partir da necessidade de cada avaliação. Desta forma, por exemplo, o avaliador pode escolher iniciar o planejamento de sua avaliação a partir do perfil do avaliador, ou pelo tipo de dado (qualitativo ou quantitativo), pelo tipo de método empírico ou de avaliação que deseja realizar. Com isso, o avaliador tem a liberdade de desenvolver a avaliação conforme o tipo de linguagem que deseja avaliar e de uma forma iterativa, na qual ele poderá seguir diferente instâncias/caminhos na avaliação da DSL.

3.2. Etapas do *Framework* Usa-DSL

O *framework* Usa-DSL é constituído por 11 etapas, que podem ser selecionadas e executadas de acordo com o tipo de linguagem que será avaliada. As etapas do *Framework* Usa-DSL são descritas a seguir (ver Figura 3.1):

- Etapa 1** *Perfil do Avaliador*: define os perfis de avaliador, que podem ser Experiente em IHC, Analista do Domínio, Desenvolvedor do Domínio ou Testador do Domínio, por exemplo. Estes perfis foram definidos a partir do Mapeamento da RSL, no qual os autores [Cuenca et al. 2015], [Ewais and De Troyer 2014], [Gibbs et al. 2015], [Albuquerque et al. 2015], [Barisic et al. 2014] descrevem ter utilizado especialistas no domínio para realização de avaliações.
- Etapa 2** *Responsabilidades Éticas e Legais*: toda pesquisa que envolve seres humanos deve atender a questões éticas. Sendo assim, o Usa-DSL está baseado no *framework* DECIDE, que sugere que seja fornecido ao participante um documento, no qual estará explicado o objetivo da pesquisa, e ainda certifica a segurança do participante prevendo o sigilo dos seus dados não associando suas respostas ao seu perfil e nem relatando em qualquer documento associado a pesquisa. Também está baseado no código da Association for Computing Machinery (ACM)¹.
- Etapa 3** *Tipo de Dados*: os dados quantitativos são resultados numéricos que preveem a quantidade de respostas atribuídas para determinado item de uma questão. Já os dados qualitativos constituem informações subjetivas referentes a opinião do participante sobre o objeto estudado. Estes dados buscam prever que tipo de informação o avaliador pretende obter, Albuquerque *et al.* [Albuquerque et al. 2015] sugere o uso dos dois tipos de dados, a fim de obter

¹<https://www.acm.org/about-acm/acm-code-of-ethics-and-professional>

Etapas	Fases			
	Definição	Execução	Análise	Resultados
1- Perfil do Avaliador	D1 Definir o perfil do avaliador	E1 Aplicar os instrumento para levantar perfil	A1 Analisar o perfil levantado	R1 Reportar o perfil levantado
2- Responsabilidades Éticas e Legais	D2 Definir o documento de consentimento de participação	E2 Apresentar o documento e coletar assinaturas dos participantes		R2 Reportar o número de participantes e o documento utilizado
3 - Tipo de Dados	D3 Definir o tipo de dado que será coletado			
4 - Métodos Empíricos (ES)	D4 Definir a metodologia empírica que será utilizada no estudo	E4 Desenvolver o protocolo e executar estudo	A4 Analisar o protocolo desenvolvido	R4 Reportar o protocolo desenvolvido
5 - Método de Avaliação	D5 Definir o tipo de avaliação de usabilidade que será utilizada	E5 Preparar e conduzir a avaliação		R5 Reportar a condução da avaliação
6 - Métricas	D6 Definir métricas para validação da linguagem			
7 - Instrumentos de Coleta	D7 Definir os instrumentos de coleta de dados	E7 Coletar os dados	A7 Analisar os dados coletados	R7 Reportar a análise dos dados
8 - Instruções sobre o Sistema	D8 Definir os instrumentos de instrução e treinamento	E8 Apresentar os instrumentos de instrução e realizar o treinamento		R8 Reportar os instrumentos utilizados
9 - Condução da Avaliação	D9 Definir o local de execução	E9 Execução da tarefa prevista para os participantes	A9 Analisar as tarefas realizadas	R9 Reportar a análise das tarefas
10 - Empacotamento dos Dados	D10 Definir como serão armazenados os dados	E10 Armazenar os dados e informações obtidas		
11 - Relatório da Avaliação	D11 Definir como serão relatados dos dados e os resultados do estudo		A11 Analisar a documentação utilizada na avaliação	R11 Reportar os dados levantados e os resultados obtidos
Atividades				

Figura 1. Estrutura da Versão Preliminar do Framework Usa-DSL

uma visão mais abrangente e completa sobre a opinião dos participante. Já Barisic *et al.* [Barisic et al. 2014] utiliza em seus experimentos dados quantitativos e considera que sejam suficientes para o objetivo da sua pesquisa. Desta forma, o pesquisador deve analisar a linguagem a ser avaliada e definir qual tipo de dado melhor atende aos objetivos de sua avaliação.

Etapa 4 Métodos Empíricos (Engenharia de Software): Wohlin *et al.* [Wohlin 2014] sugerem o uso de *survey*, estudo de caso e experimento. Estes métodos podem ser definidos a partir do objeto de estudo, do perfil dos participantes da avaliação e/ou dos dados que se pretende coletar.

Etapa 5 Método de Avaliação: podem ser realizados junto a usuários finais (observação de uso, teste de usabilidade, por exemplo) ou a especialistas (avaliação heurística, Percurso cognitivo - *Cognitive Walkthrough*, por exemplo).

Etapa 6 Métricas: foram definidas a partir da RSL ([Rodrigues et al. 2017]). São sugeridas as seguintes características: compreensão/aprendizado, facilidade no uso, esforço/tempo de conclusão, complexidade percebida, taxa de conclusão, taxa de erro na tarefa, eficiência e eficácia. Estas métricas devem orientar a construção dos instrumentos de coleta de dados, a serem utilizados durante a avaliação.

- Etapa 7** *Instrumentos de Coleta*: estão baseados em [Rodrigues et al. 2017], [Rogers et al. 2013] e [Rouly et al. 2014]. Sugerem-se *checklist* para avaliação heurística, *checklist* ergonômico, questionários, entrevistas, observação de uso e gravação de ação do usuário.
- Etapa 8** *Instruções sobre o Sistema*: conforme Wohlin *et al.* [Wohlin 2014], as instruções do sistemas constituem em manual de uso, apresentação dos instrumentos, e tarefa a ser executada. Estes materiais devem ser distribuídos e utilizados ao realizar um experimento, *survey* ou estudo de caso e servem para orientar e esclarecer à dúvidas dos participantes.
- Etapa 9** *Condução da Avaliação*: é nessa etapa que se aplica o que foi definido nas etapas anteriores. Sendo assim, é necessário que as demais etapas tenham sido cuidadosamente definidas e testadas antes de envolver os participantes da avaliação. Caso não tenha bem definido o que deve ser feito e como será executado, o pesquisador poderá perder a amostra e invalidar os dados coletados. Por esta razão, os dados que deseja coletar, a tarefa e os materiais que serão aplicados devem ter passado por um teste piloto anteriormente e somente após obter uma versão estável, a avaliação poderá ser executada. Outro aspecto se refere à confirmação dos participantes, de forma que o avaliador deve buscar garantir que o número necessário de participantes compareceram às atividades da fase de execução.
- Etapa 10** *Empacotamento dos Dados*: ao finalizar a avaliação aconselha-se que os materiais utilizados no treinamento e na coleta dos dados sejam armazenados em local seguro e de fácil acesso, para que o estudo possa ser replicado quando necessário e assim conduzir uma reavaliação da linguagem que está sendo testada. Ainda, o armazenamento dos dados coletados também são úteis para uma posterior comparação com avaliações que poderão vir a ser conduzidas até a finalização e/ou estabilização da linguagem.
- Etapa 11** *Relatório da Avaliação*: deve seguir o método de avaliação escolhido. Cada método gera um tipo de documento para relatar a avaliação, este documento deve conter a descrição da DSL analisada, objetivo da avaliação, descrição do método aplicado, perfil dos participantes/especialistas, instrumentos aplicados, dados coletados, análise dos dados, problemas identificados, soluções propostas, dentre outros.

3.3. Fases do *Framework Usa-DSL*

As fases são compostas pela definição, execução, análise e resultados da avaliação, como podem ser observadas na Figura 3.1. Cada Fase possui um conjunto de atividades, que se relacionam à etapa que está sendo desenvolvida.

- Fase 1** *Definição*: esta fase tem como propósito a organização da pesquisa por parte do avaliador, a definição do que será realizado para avaliar a DSL. Nesta fase devem ser definidos e construídos documentos, tomadas decisões sobre os dados que se deseja obter, qual o perfil de usuário que fará parte da avaliação. Ou seja, nesta fase se estrutura e se planeja a avaliação da DSL.
- Fase 2** *Execução*: nesta fase se concretiza a avaliação. São aplicados os documentos para coleta dos dados e conduzida a avaliação, conforme protocolo que fora definido.
- Fase 3** *Análise*: terá como objetivo realizar a análise dos artefatos gerados tanto na fase de Definição como na fase de Execução. Na fase de definição está análise é feita

para que sejam ajustados os documentos e decisões tomadas quanto ao desenvolvimento da avaliação. Sobre a fase de Execução à análise se concentra nos dados e tarefas geradas durante a realização da avaliação.

Fase 4 Resultados: nesta fase, o avaliador realiza o registro do protocolo utilizado, dos artefatos gerados, dados analisados. Estas informações podem ser reportadas por meio de relatório no caso da indústria ou artigo científico no caso de pesquisa acadêmica. O formato do relatório também depende do tipo de avaliação que foi adotada.

3.4. Atividades do *Framework Usa-DSL*

As atividades do *framework Usa-DSL* são formadas por um conjunto de teorias e práticas utilizadas para realização de avaliação de software pelas áreas de Interação Humano-Computador e Engenharia de Software. Os conceitos utilizados foram identificados a partir de uma Revisão Sistemática da Literatura [Rodrigues et al. 2017], e autores das áreas envolvidas. O quadro com a estrutura das atividades pode ser visto na Figura 3.1 e suas descrições no repositório online, por meio do *link* http://tiny.cc/ERES2017_Usa-DSL.

4. Avaliação Preliminar

Esta avaliação foi realizada por meio da técnica de entrevista, e teve como objetivo apresentar o *framework Usa-DSL*, e obter a opinião de um desenvolvedor de DSL. Este desenvolvedor criou uma DSL para os domínios de visualização de dados geoespaciais (DSL GMaVis) e abstração do paralelismo em aplicações de *streaming* (DSL SPar). No primeiro momento desta entrevista, foi solicitado ao participante que expressasse a sua concordância em participar, por meio do Termo de Consentimento Livre e Esclarecido (TCLE). Após, foram repassadas a ele, a descrição do *framework Usa-DSL*, um exemplo de uso, e as informações contidas no glossário. Estes e outros documentos usados na avaliação estão disponíveis para consulta no repositório online.

Após os procedimentos iniciais, foi solicitado ao participante que respondesse a um questionário de caracterização, que pode ser acessado no *link* http://tiny.cc/ERES2017_USA-DSL. Com base nas respostas do entrevistado, foram destacadas as seguintes informações sobre o perfil do participante: atua profissionalmente como pesquisador e professor por um período de 7 anos; possui 2 anos de experiência em análise, desenvolvimento e teste de DSL; não possui experiência quanto à avaliação de usabilidade; já participou de métodos experimentais empíricos, e *i.e.* realizou experimento controlado e *survey*.

Ao ser questionado sobre quais critérios de usabilidade considera importante no desenvolvimento de uma DSL, o participante citou: efetividade, eficiência, satisfação de uso, prevenção de erros, facilidade de uso, facilidade de aprender como usar, facilidade de lembrar como usar, segurança, e, ainda, acrescentou o critério “desempenho”. Como justificativa para importância de tais critérios, afirmou que se o usuário não estiver satisfeito com a DSL que foi projetada, mesmo que possua todas as funcionalidades que necessita, buscará por uma ferramenta que atenda a critérios de usabilidade.

Em seguida, o avaliador apresentou ao participante o exemplo de uso, DSL Canopus, bem como esclareceu dúvidas sobre a composição e funcionamento do *framework*.

Ao final, quando o participante considerou que estava apto para responder ao questionário, o avaliador fez as perguntas, que foram gravadas em áudio e registradas em um questionário escrito. Dentre as repostas desta entrevista, destacam-se: **a) A importância em avaliar critérios de usabilidade em DSLs gráficas:** o participante informou ser importante avaliá-los, e afirmou que o usuário precisa se sentir satisfeito para continuar usando a DSL; caso contrário, irá buscar outra ferramenta, mesmo que não seja para o domínio específico. **b) Durante o processo de desenvolvimento de uma DSL o framework Usa-DSL poderia ser usado:** o participante afirmou que usaria o Usa-DSL. E justifica que, por não ter conhecimentos na área de IHC, o *framework* ajudaria no processo de avaliação de sua DSL, pois oferece uma metodologia a ser seguida. **c) O entendimento do framework é fácil:** o participante confirmou que a descrição é de fácil entendimento, mas não soube precisar se seria fácil aplicá-lo. Assim, o participante considerou o *framework* Usa-DSL de fácil compreensão, mas não soube, naquele momento, se seria de fácil uso. **d) O uso do framework facilita o planejamento, execução e análise da avaliação de DSLs:** o participante afirmou que o *framework* Usa-DSL oferece informações de forma clara. Porém, não saberia avaliar se, de fato, seria fácil de ser utilizado, embora as descrições pareçam facilitar a avaliação. Novamente, percebe-se a preocupação do participante no que se refere à facilidade de uso do *framework*. **e) O uso do framework torna a avaliação de DSL mais produtiva:** embora não tenha utilizado o *framework* Usa-DSL, ainda assim, achou que seria mais produtivo o seu uso na construção e execução da avaliação de uma DSL. E complementa que, ao utilizá-lo, não precisaria aprender sobre uma área diferente de sua pesquisa (no caso, IHC), e que certamente levaria mais de um ano para compreender os conceitos. Por fim, justifica que, esta curva de aprendizagem é o que o afasta de realizar tais avaliações. **f) Informações que adicionaria no framework:** informações sobre recrutamento de participantes, pois considerou que é difícil encontrar pessoas adequadas para participar de avaliações de DSL, por necessitar de um perfil mais específico. **g) Informações que retiraria do framework:** informou que não excluiria informação, que estão sendo apresentadas no Usa-DSL. **h) Perfis de usuário para avaliar uma DSL gráfica:** primeiramente indica a consulta a alguém com experiência em IHC, para resolver problemas básicos de interface com o usuário. E, após as modificações apontadas pelo avaliador de IHC, aplicaria uma segunda avaliação com usuários do domínio. O participante acredita que ao realizar a avaliação com alguém experiente em IHC, evitaria que o usuário do domínio realizasse várias avaliações. Relatou, que ao utilizar os dois perfis seria mais produtivo, pois é uma tarefa difícil conseguir voluntários para realização de tais avaliações.

5. Considerações Finais e Trabalhos Futuros

Nos últimos anos é possível identificar uma aumento em relação ao desenvolvimento de DSLs. A literatura menciona que, para as DSLs se tornarem mais atrativas e com maior qualidade de uso, é necessário dedicar mais esforço na avaliação de usabilidade destas DSLs [Quiñones and Rusu 2017].

Neste sentido, busca-se com o *framework* Usa-DSL auxiliar os engenheiros de linguagem com a avaliação das DSLs desenvolvidas e minimizar dificuldades inerentes a utilização das aplicações construídas. O *framework* Usa-DSL ainda busca tornar a avaliação de DSLs mais fácil, produtiva e, conseqüentemente, procura alcançar critérios de usabilidade e de satisfação de uso [Sinha and Smidts 2006]. Com o uso do Usa-DSL,

busca-se facilitar o planejamento, execução e análise da avaliação de DSLs, oferecendo informações de forma clara e objetivas para que a equipe de avaliação (analista, desenvolvedor, testador) possa planejar e avaliar a DSL.

Após a avaliação preliminar apresentada na Seção 4, foi realizado um Grupo Focal envolvendo participantes com experiência em IHC, Engenharia de Software e Teste de Desempenho. Esta avaliação teve como objetivo apresentar o *framework* Usa-DSL, e obter a opinião dos participantes quanto a sua clareza, facilidade no uso e entendimento. Os resultados obtidos e as alterações sugeridas serão apresentadas em outro artigo, que está em processo de submissão. Referente aos trabalhos futuros, pretende-se aplicar o *framework* Usa-DSL na avaliação de diferentes DSLs, a partir de diferentes métodos de avaliação de usabilidade; refinar os materiais informativos do Usa-DSL, incluindo exemplos de uso; utilizar o *framework* Usa-DSL com diferentes perfis de avaliadores de forma a refinar o *framework* e sua descrição.

Referências

- Albuquerque, D., Cafeo, B., Garcia, A., Barbosa, S., Abrahão, S., and Ribeiro, A. (2015). Quantifying usability of domain-specific languages: An empirical study on software maintenance. *Journal of Systems and Software*, 101:245 – 259.
- Barbosa, S. and da Silva, B. (2010). *Interação humano-computador*. Campus.
- Barisic, A., Amaral, V., Goulão, M., and Aguiar, A. (2014). Introducing usability concerns early in the dsl development cycle: Flowsl experience report. In *Proceedings of the 17th International Conference on Model Driven Engineering Languages and Systems*.
- Barisic, A., Amaral, V., Goulão, M., and Barroca, B. (2011). Quality in use of domain-specific languages: A case study. In *Proceedings of the 3rd ACM SIGPLAN Workshop on Evaluation and Usability of Programming Languages and Tools*, pages 65–72, New York, NY, USA. ACM.
- Barisic, A., Amaral, V., Goulão, M., and Barroca, B. (2012). Evaluating the usability of domain-specific languages. *Formal and practical aspects of domain-specific languages: Recent developments*.
- Bernardino, M., Zorzo, A., and Rodrigues, E. (2016). Canopus: A Domain-Specific Language for Modeling Performance Testing. In *Proceedings of the 9th International Conference on Software Testing, Verification and Validation (ICST)*, pages 157–167. IEEE.
- Conrado, D. B. F. (2012). Abordagem para criação de linguagem específica de domínio para robótica móvel. Master's thesis, Universidade Federal de São Carlos, São Paulo.
- Cuenca, F., Bergh, J. V. d., Luyten, K., and Coninx, K. (2015). A user study for comparing the programming efficiency of modifying executable multimodal interaction descriptions: A domain-specific language versus equivalent event-callback code. In *Proceedings of the 6th Workshop on Evaluation and Usability of Programming Languages and Tools*, pages 31–38, New York, NY, USA. ACM.

- Ewais, A. B. and De Troyer, O. (2014). A usability evaluation of graphical modelling languages for authoring adaptive 3d virtual learning environments. In *Proceedings of the 6th International Conference on Computer Supported Education*, pages 459–466.
- Fowler, M. (2010). *Domain Specific Languages*. Addison-Wesley Professional, 1 edition.
- Gibbs, I., Dascalu, S., Harris, F. C., and Jr. (2015). A separation-based ui architecture with a dsl for role specialization. *Journal of Systems and Software*, 101:69 – 85.
- Mernik, M., Heering, J., and Sloane, A. M. (2005). When and how to develop domain-specific languages. *ACM Computing Surveys*, 37(4):316–344.
- Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann, San Francisco, USA.
- Quiñones, D. and Rusu, C. (2017). How to develop usability heuristics: A systematic literature review. *Computer Standards & Interfaces*, 53:89–122.
- Rodrigues, I., Campos, M., and Zorzo, A. (2017). Usability evaluation of domain-specific languages: a systematic literature review. pages 522–534.
- Rogers, Y., Sharp, H., and Preece, J. (2013). *Design de interação: além da interação humano-computador*. Bookman.
- Rouly, J. M., Orbeck, J. D., and Syriani, E. (2014). Usability and suitability survey of features in visual ides for non-programmers. In *Proceedings of the 5th Workshop on Evaluation and Usability of Programming Languages and Tools*, pages 31–42, New York, NY, USA. ACM.
- Sinha, A. c. and Smidts, C. d. (2006). An experimental evaluation of a higher-ordered-typed-functional specification-based test-generation technique. *Empirical Software Engineering*, 11(2):173–202.
- Stone, D., Jarrett, C., Woodroffe, M., and Minocha, S. (2005). *User Interface Design and Evaluation*. Interactive Technologies. Elsevier Science.
- Teruel, M. A., Navarro, E., López-Jaquero, V., Montero, F., and González, P. (2014). A csw requirements engineering case tool: Development and usability evaluation. *Information and Software Technology*, 56(8):922 – 949.
- Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, page 38. ACM.