

A framework for visualizing HCI pattern languages through network diagrams

DIEGO MOREIRA DA ROSA, Pontifícia Universidade Católica do Rio Grande do Sul

ANDREA GNECCO, Pontifícia Universidade Católica do Rio Grande do Sul

MILENE SILVEIRA, Pontifícia Universidade Católica do Rio Grande do Sul

CHRISTIAN MATTJIE, Pontifícia Universidade Católica do Rio Grande do Sul

RODRIGO C. BARROS, Pontifícia Universidade Católica do Rio Grande do Sul

SOFIA APUZZO, Pontifícia Universidade Católica do Rio Grande do Sul

MARCIO SARROGLIA PINHO, Pontifícia Universidade Católica do Rio Grande do Sul

ISABEL MANSSOUR, Pontifícia Universidade Católica do Rio Grande do Sul

Defined in the late 1970s by architect Christopher Alexander, the concepts of patterns and pattern language have been adopted by specialists in several areas, including Software Engineering and Human-Computer Interaction (HCI). The hierarchical structure of pattern languages predicted by Alexander favors their representation through network diagrams, a common practice among HCI researchers. Despite the advantages of using graphs to elicit pattern interrelationships within a language, it was not possible to find a structured and documented method for visualizing pattern languages in modern network diagram tools. In this work, we present a framework that describes the steps for the visualization of HCI pattern languages through network diagrams. As part of the framework, a tool capable of converting pattern description XML files into a graph description file ready to be interpreted by feature-rich visualization applications was developed. The framework was tested with data from Welie.com, an online library containing 132 interaction patterns, and the resulting graph was visualized in two well-known applications for designing network structures (Gephi and InfraNodus). The results indicate that the proposed solution is a viable strategy for the visualization of pattern languages.

Categories and Subject Descriptors: H.1.2 [**Models and Principles**]: User/Machine Systems—*Human factors*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Evaluation/methodology*

General Terms: Human Factors

Additional Key Words and Phrases: visualization, patterns, pattern language, HCI, network diagram, graph

ACM Reference Format:

Da Rosa, D. M. et al. 2023. A framework for visualizing HCI pattern languages through network diagrams. HILLSIDE Proc. of Conf. on Pattern Lang. of Prog. V (January 2023), 11 pages.

1. INTRODUÇÃO

Os padrões (*patterns*), também conhecidos na área da Computação como padrões de projeto (*design patterns*), são definidos como uma solução geral para um problema recorrente em um determinado contexto [Alexander 1977]. Com o passar dos anos, o conceito foi adotado por especialistas e pesquisadores de outras áreas, incluindo

Authors' address: Escola Politécnica, PUCRS, Av. Ipiranga 6681, 90619-900 Porto Alegre, RS, Brazil. Email: diego.rosa81@edu.pucrs.br; andrea.gnecco@edu.pucrs.br; milene.silveira@pucrs.br; christian.oliveira95@edu.pucrs.br; rodrigo.barros@pucrs.br; sofia.apuzzo@edu.pucrs.br; marcio.pinho@pucrs.br; isabel.manssour@pucrs.br.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission. A preliminary version of this paper was presented in a writers' workshop at the 29th Conference on Pattern Languages of Programs (PLoP). PLoP'22, October 17-24, Virtual Online. Copyright 2022 is held by the author(s). HILLSIDE 978-1-941652-18-3

a Engenharia de Software [Gamma et al. 1995] e a Interação Humano-Computador (IHC) [Bayle et al. 1998; Borchers 2000]. Em IHC, os padrões se popularizam a partir do final da década de 1990, se tornam frequentes em trabalhos científicos envolvendo o *design* de interação, e recebem a nomenclatura de padrões de interação ou, simplesmente, padrões de IHC.

De acordo com Alexander, «nenhum padrão é uma entidade isolada» e «cada padrão pode existir no mundo, apenas na medida em que ele é apoiado por outros padrões: os padrões maiores nos quais ele está inserido, os padrões de mesmo tamanho que o rodeiam, e os padrões menores que estão inseridos nele» [Alexander 1977]. Um conjunto de padrões inter-relacionados e associados a um determinado contexto forma uma linguagem de padrões. Já a característica hierárquica destas relações faz com que as linguagens tenham a estrutura de uma rede [Alexander 1977]. Desde o início da adoção das linguagens de padrões na área de IHC, esta característica foi reconhecida e destacada, sendo comum entre os pesquisadores a representação visual das linguagens através de diagramas de rede (ver Seção 2).

O uso de grafos para representar visualmente as linguagens de padrões permite que sejam facilmente identificadas as inter-relações entre os padrões e a estrutura hierárquica existente entre eles, facilitando assim a compreensão e o uso da linguagem na prática. Apesar desta vantagem, não foram encontrados métodos estruturados e devidamente documentados para a visualização de linguagens de padrões através de aplicações modernas para desenho de diagrama de rede, como por exemplo Gephi¹ e InfraNodus². O objetivo deste trabalho é propor um *framework*, ou uma sequência de passos estruturados, para a visualização de linguagens de padrões de IHC em ferramentas de diagrama de rede.

Como parte do *framework* proposto, foi desenvolvida uma ferramenta que converte automaticamente arquivos de descrição de padrões em um arquivo de descrição de grafo pronto para ser utilizado como entrada nas aplicações de visualização. Como fonte de dados para os testes, foi utilizada a biblioteca *online* Welie.com³, uma coleção de 132 padrões de interação, a qual inclui as descrições dos padrões em formato PLML (*Pattern Language Markup Language*)⁴. Posteriormente, o arquivo GEXF (*Graph Exchange XML Format*)⁵ resultante da execução da ferramenta e contendo a descrição do grafo foi utilizado para produzir visualizações nas aplicações Gephi e InfraNodus, amplamente utilizadas para o desenho de grafos. Os resultados obtidos indicam que a sequência de passos utilizada, incluindo o processamento dos arquivos através da ferramenta, compõem um *framework* viável para a visualização de linguagens de padrões de IHC através de diagramas de rede.

Este artigo é composto de cinco seções incluindo esta introdução. A Seção 2 apresenta trabalhos relacionados e outras definições a respeito de padrões e visualização de linguagens de padrões. A metodologia do trabalho é apresentada na Seção 3. A Seção 4 descreve o framework e os resultados obtidos, enquanto a Seção 5 apresenta considerações finais e trabalhos futuros.

2. TRABALHOS RELACIONADOS

Nesta seção, são apresentados quatro trabalhos relacionados ao tema da visualização de linguagens de padrões. O primeiro estudo apresenta uma revisão histórica e uma proposta de definição formal para linguagens de padrões no contexto do *design* de interação [Borchers 2000]. Já o segundo trabalho detalha, no mesmo contexto, a organização das linguagens de padrões em estruturas de rede [Van Welie and Van der Veer 2003]. Uma ferramenta básica para a visualização de linguagens de padrões é apresentada no terceiro estudo [Schobert and Schümmer 2006], enquanto o quarto artigo apresenta uma proposta de aplicação de análise de redes sociais a uma rede de padrões [Park 2015]. Por fim, esta seção encerra-se com uma discussão das relações entre os trabalhos mencionados anteriormente e o presente estudo.

¹<https://gephi.org/>

²<https://infranodus.com/>

³<http://www.welie.com/>

⁴PLML é uma linguagem de marcação baseada na linguagem XML (*eXtensible Markup Language*).

⁵Assim como a PLML, a GEXF também é uma linguagem de marcação baseada na linguagem XML.

2.1 Padrões no *Design* de Interação

Borchers é o autor de uma das primeiras tentativas de definir formalmente uma linguagem de padrões de IHC [Borchers 2000]. Seu trabalho inicia com uma revisão histórica de padrões desde a época de Alexander até suas primeiras aplicações no *design* de interação. Conforme descreve o autor, um padrão é uma solução geral para um problema que ocorre com frequência dentro de um determinado contexto. Já uma linguagem de padrões é um conjunto organizado e coerente de padrões relacionados entre si. Após terem se consolidado na área de Engenharia de Software no início da década de 1990, os dois conceitos passam a atrair a atenção de pesquisadores de IHC a partir do final da mesma década, sendo tema de diversos workshops na série de Conferências sobre Fatores Humanos em Sistemas Computacionais (CHI) da ACM a partir de 1997 [Bayle et al. 1998; Griffiths et al. 2000; Van Welie et al. 2002; Fincher et al. 2003; Schümmer et al. 2004].

No mesmo trabalho, Borchers define quais seriam os atributos básicos para a descrição de um padrão: **nome**, **ranking** (nível de confiança), **ilustração**, **problema**, **forças**, **exemplos**, **solução** e **diagrama** [Borchers 2000]. Além disso, estabelece uma estrutura de rede para indicar as relações entre os padrões de uma linguagem. De acordo com a sua definição, uma linguagem é representada por um grafo direcionado, no qual cada nó representa um dos padrões da coleção. O **contexto** de um padrão é indicado pelas arestas que chegam ao nó. Já as arestas que partem do nó indicam quais os **padrões relacionados**, ou seja, quais outros padrões podem ser utilizados na implementação do padrão representado pelo nó. A Fig. 1 exibe a estrutura proposta por Borchers usando como exemplo uma linguagem para exibições interativas.

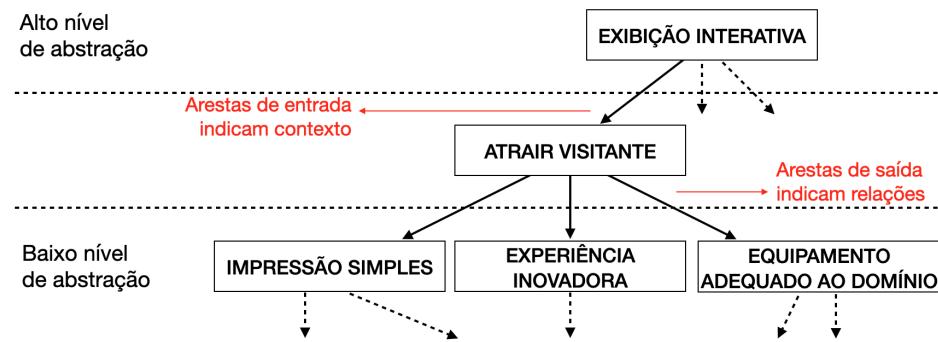


Fig. 1. Estrutura de rede proposta por Borchers para linguagens de padrões (adaptada do artigo original [Borchers 2000])

2.2 Estrutura e organização de linguagens de padrões

Van Welie e van der Veer detalham ainda mais a estrutura e organização das linguagens de padrões de IHC [Van Welie and Van der Veer 2003]. Embasados pelo texto original de Alexander [Alexander 1977], os autores afirmam que um padrão, por si só, é apenas um pequeno pedaço de todo o conhecimento de *design*, ou seja, uma peça de um quebra-cabeça maior. Um padrão individual pode ser muito valioso para os *designers*, mas, quando os padrões estão relacionados a outros, pode-se alcançar uma compreensão mais completa do problema. Neste trabalho, os autores reforçam a visão de Alexander (e Borchers) de que a linguagem “produz” o *design* ao percorrer dos padrões de mais alto aos padrões de mais baixo nível, ou ainda, «do *design* de cidades detalhando até o *design* do assento da janela, uma hierarquia de escala» [Van Welie and Van der Veer 2003]. Além de ressaltar a importância da estrutura de rede das linguagens de padrões, os autores ampliam a definição de Borchers, descrevendo três tipos diferentes de relações entre os padrões de interação:

- **Agregação:** utilizada para descrever relações todo-parte, ou seja, quando um padrão maior contém ou agrupa outros padrões menores (relação “tem-um”).
- **Especialização:** utilizada nos casos em que um padrão é uma versão especializada de outro padrão, ou seja, herda características de um padrão mais genérico e adiciona opções estendidas (relação “é-um”).
- **Associação:** utilizada para descrever casos em que padrões são utilizados em contextos semelhantes ou fornecem soluções alternativas para um mesmo tipo de problemas (relação “está-relacionado-a”).

Ainda no mesmo trabalho, van Welie e van der Veer apresentam a estrutura e organização da biblioteca de padrões para *web design* Welie.com (recém criada na época e a mesma selecionada para a realização dos testes no presente estudo) [Van Welie and Van der Veer 2003]. No diagrama de rede que apresenta a estrutura da linguagem, os autores destacam as diferentes camadas que podem ser identificadas quando os padrões são organizados do mais alto para o mais baixo nível dentro da hierarquia.

2.3 Visualização de linguagens de padrões

Embora descrevam como formalizar e organizar a estrutura de rede das linguagens de padrões de IHC, os trabalhos anteriores não abordam a questão do desenho do diagrama de rede. Uma das primeiras ferramentas de software para a visualização de linguagens de padrões foi proposta por Schobert e Schümmer em 2006 [Schobert and Schümmer 2006]. Os autores destacam a ausência, na época, de uma ferramenta que auxiliasse a criação das descrições textuais e do mapa de padrões simultaneamente. A partir disso, os autores propõem a ferramenta *Collaborative Pattern Editor* (CoPE) para preencher esta lacuna.

Nesta ferramenta, os diagramas são gerados automaticamente a partir das descrições dos padrões, as quais podem ser escritas na sintaxe Wiki em um editor de texto ou importadas de arquivos XML no formato PLML. A ferramenta permite ainda que os autores trabalhem colaborativamente nas linguagens de padrões. Cores podem ser atribuídas manualmente a regiões do diagrama de forma a explicitar melhor as relações entre os padrões.

2.4 Análise de rede aplicada a linguagens de padrões

Uma proposta de visualização para a linguagem de padrões de Alexander [Alexander 1977] através do método de Social Network Analysis (SNA) é o foco do estudo de Park [Park 2015]. A partir da construção de um mapa de rede (*Network Map*) para os padrões, o método SNA foi executado com três diferentes critérios de importância: **grau**, **proximidade**, e **intermediação**. Park implementou uma pontuação para cada um destes critérios dentro da linguagem utilizada, obtendo, ao final da análise, conjuntos de padrões com distintas pontuações.

O resultado desta classificação pode ser visualizado em forma de uma rede de padrões interconectados, na qual: padrões de maior pontuação para o critério **grau** possuem mais conexões; padrões de maior pontuação para o critério **proximidade** estão mais próximos uns dos outros; e, por último, padrões com maiores pontuações para o critério de **intermediação**, estão em posições mais centralizadas do grafo. Ao mesmo tempo, padrões com maior pontuação geral (considerando os três critérios), foram categorizados como «padrões-chave».

2.5 Discussão sobre os trabalhos relacionados

A análise dos três primeiros artigos apresentados indica uma ênfase dos desenvolvedores de padrões de IHC na representação visual das linguagens [Borchers 2000; Van Welie and Van der Veer 2003; Schobert and Schümmer 2006]. De fato, diagramas de rede são comuns em estudos de IHC que propõem linguagens de padrões. Embora Alexander já tivesse previsto as inter-relações entre os padrões e uma organização hierárquica das linguagens, a mesma ênfase na representação visual através de grafos não é observada, por exemplo, nas áreas da Arquitetura e da Engenharia de Software. Esta característica das publicações de IHC reforça a relevância do *framework* proposto no presente estudo para a geração de visualizações das linguagens de padrões.

No trabalho de Schobert e Schümmer, é apresentada uma solução semelhante à solução proposta no presente estudo [Schobert and Schümmer 2006]. No entanto, a ferramenta desenvolvida pelos autores 16 anos atrás não

encontra-se mais disponível e o software parece ter sido descontinuado. Já o *framework* proposto no presente trabalho funciona em conjunto com ferramentas atuais de visualização ricas em funcionalidades, tais como Gephi e InfraNodus. O uso destas ferramentas permite, além da geração de diferentes tipos de diagramas de rede altamente configuráveis, a realização de análises mais complexas das relações entre os padrões, tais como a SNA proposta por Park no quarto estudo [Park 2015].

3. METODOLOGIA

A Fig. 2 resume as etapas desenvolvidas ao longo deste estudo. Em um primeiro momento foi realizado um planejamento das atividades, a escolha do *dataset* e o pré-processamento dos dados. Logo após, foi realizado o desenvolvimento da ferramenta *HCIPatternsConverter*. A ferramenta foi validada através da conversão dos arquivos e da geração das visualizações dos diagramas de rede. Por fim, as etapas foram organizadas e detalhadas dando origem ao *framework* proposto.

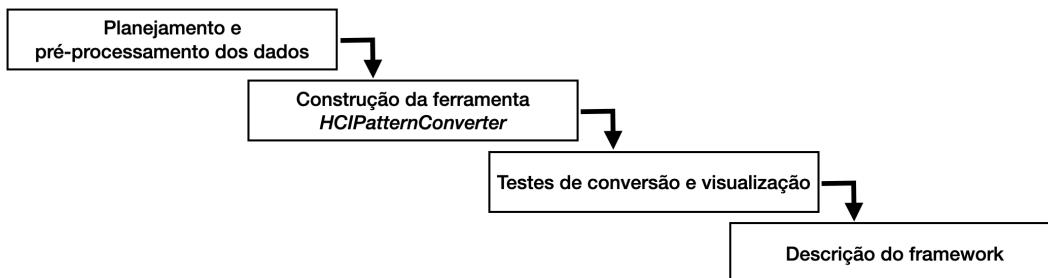


Fig. 2. Metodologia do estudo.

Inicialmente foi feito um planejamento da visualização, com a escolha da biblioteca Welie.com como conjunto de dados a ser utilizado. Também foi identificada a necessidade de desenvolver a ferramenta para conversão dos arquivos de padrões (PLML) para um arquivo de descrição de grafo (GEXF). Uma pesquisa exploratória revelou as duas aplicações para visualização do grafo através de diagramas de rede: Gephi e InfraNodus. Após o download dos arquivos PLML da biblioteca, foi realizado um pré-processamento manual para ajustar arquivos que estavam com problemas de formatação.

Em paralelo ao pré-processamento dos arquivos, foi desenvolvida uma ferramenta na linguagem de programação Java que realiza a leitura dos arquivos PLML e produz um arquivo GEXF contendo a descrição do grafo correspondente à linguagem de padrões. A linguagem de programação Java foi escolhida por ser de conhecimento dos autores e por possuir amplo suporte à manipulação de documentos XML.

A validação da ferramenta desenvolvida ocorreu através do processamento dos 132 arquivos PLML da biblioteca Welie.com já devidamente pré-processados. O arquivo GEXF produzido como resultado foi utilizado como entrada nas aplicações de visualização Gephi e InfraNodus para a geração dos diagramas de rede.

Tendo a ferramenta sido validada com sucesso, as etapas do processo foram detalhadas e organizadas de forma a originar o *framework* descrito na Seção 4. O *framework* descreve de forma estruturada os passos necessários para a geração de visualizações de uma linguagem de padrões e também os artefatos intermediários envolvidos no processo.

4. FRAMEWORK E VISUALIZAÇÕES DA LINGUAGEM

A organização dos passos descritos na metodologia deste estudo deu origem ao *framework* apresentado na Fig. 3. Nesta seção, são descritas as etapas do *framework* e os resultados da execução de cada etapa.

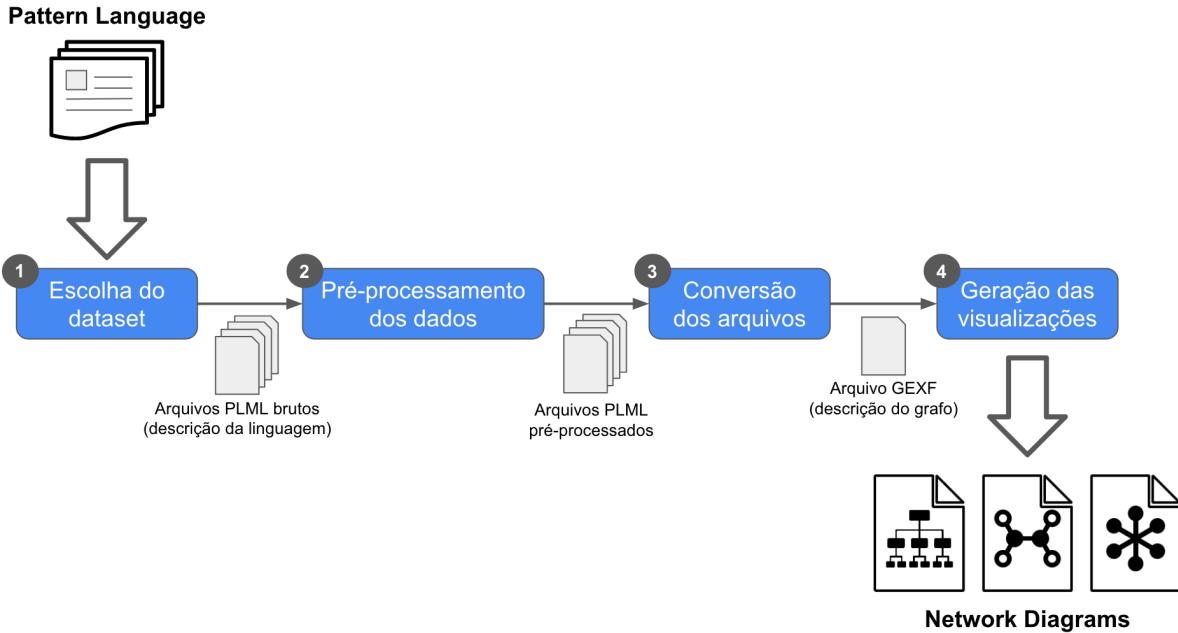


Fig. 3. Framework para visualização de linguagens de padrões de IHC através de diagramas de rede.

4.1 Escolha do *dataset*

A primeira etapa consiste na escolha da linguagem de padrões a ser visualizada. Como, até o momento, a ferramenta de conversão aceita apenas arquivos PLML como entrada, a linguagem deve estar obrigatoriamente descrita neste formato. Caso a biblioteca não forneça as descrições dos padrões em PLML, será necessário primeiro gerar essas descrições. Existem planos de aprimorar a ferramenta para aceitar outros formatos no futuro, como Wiki e HTML.

A biblioteca Welie.com foi escolhida como conjunto de dados para a realização do estudo. Além de ser uma biblioteca com 132 padrões de interação bem conhecida e consolidada na área de IHC, esta coleção de padrões também disponibiliza os arquivos XML contendo a descrição de cada padrão. Os arquivos utilizam a sintaxe PLML, que é uma linguagem para descrição de padrões criada por pesquisadores de IHC [Fincher 2003]. Após a realização do *download* dos arquivos de todos os 132 padrões da linguagem, percebeu-se que os mesmos possuíam problemas de formatação, sendo necessário, portanto, um pré-processamento manual antes da execução da ferramenta.

4.2 Pré-processamento dos dados

Uma etapa de pré-processamento dos dados pode ser necessária caso os arquivos de descrição dos padrões não sigam o padrão da linguagem PLML. Esta linguagem define um elemento `<pattern-link>` para a criação de ligações entre os padrões. Este elemento, por sua vez, pode conter quatro atributos: o atributo `type` determina o tipo de relação entre os padrões; os atributos `patternID` e `collection` servem como identificadores únicos do padrão referenciado; já o atributo `label` oferece um rótulo legível para o padrão. Para que a ferramenta desenvolvida possa reconhecer devidamente as ligações e transformar os vários arquivos PLML em um grafo, é estritamente necessário que elementos `<pattern-link>` estejam bem formatados, com atenção especial para os atributos `type` e `patternID`. Vale ressaltar que, caso os arquivos sigam corretamente o formato PLML, a etapa de pré-processamento pode não ser necessária.

No caso dos arquivos da Welie.com, dois problemas principais foram identificados nas ligações entre os padrões. Primeiramente, os elementos <pattern-link> não incluíam o atributo type. Como estes elementos eram utilizados a todo momento para referenciar outros padrões em meio ao texto, não era possível identificar o tipo de relação que o autor havia idealizado para cada ligação. Inclusive, o elemento <pattern-link> era utilizado em situações nas quais o autor pretendia informar que o padrão referenciado **não** estava relacionado com o padrão do arquivo. Para contornar esta situação, foram definidos sete tipos de relações para o atributo type⁶:

- a. **is-alternative**: Indica que o padrão atual é uma solução alternativa para o mesmo problema abordado pelo padrão referenciado.
- b. **is-contained-by**: Indica que o padrão atual é de um nível mais baixo e é usado (junto com outros) para instanciar o padrão referenciado (de nível mais alto).
- c. **contains**: Significa o oposto de **is-contained-by**, ou seja, o padrão atual é de um nível mais alto e usa o padrão referenciado (junto com outros) para ser instanciado.
- d. **specialization**: Indica que padrão atual é uma solução semelhante porém mais especializada que o padrão referenciado.
- e. **generalization**: Significa o oposto de **specialization**, ou seja, o padrão atual é uma solução mais genérica que o padrão referenciado.
- f. **is-alias**: Indica que o padrão atual é idêntico ao outro padrão (dois padrões iguais apenas com nomes diferentes).
- g. **not-related**: Significa que a ligação não indica nenhuma relação específica entre os dois padrões.

A principal tarefa do pré-processamento consistiu em uma leitura dos 132 arquivos PLML e do preenchimento manual do valor do atributo type para cada um dos elementos <pattern-link> encontrados. Foi necessário deduzir o tipo mais adequado para as ligações a partir do contexto da descrição textual de cada padrão.

Um segundo problema encontrado foi causado por alguns erros de digitação nos atributos patternID dos elementos <pattern-link>. Estes erros impediam a correta identificação do padrão referenciado. Novamente as ligações problemáticas foram corrigidas manualmente, tornando os arquivos prontos para o processo de conversão.

4.3 Conversão dos arquivos

A ferramenta *HCIConverter* foi desenvolvida em linguagem Java para a conversão dos arquivos de descrição dos padrões. O código da ferramenta está disponível na plataforma GitHub⁷. A ferramenta faz a leitura do conjunto de arquivos PLML, realiza uma validação das ligações entre os padrões (e.g., elimina ligações duplicadas) e produz um arquivo GEXF. O formato GEXF é aceito como entrada por diversas ferramentas de visualização de grafos como as descritas na Seção 4.4.

Em um primeiro momento, a ferramenta foi programada para eliminar: ligações do tipo **not-related**; ligações do tipo **is-alias** que referenciavam padrões de outras bibliotecas; e ligações duplicadas (ligações com tipo, padrão origem e padrão destino idênticos). Ainda assim, foi identificado um grande número que ligações (469), o que acabava por produzir um número elevado de arestas no grafo. Para contornar esta situação, a ferramenta foi modificada para transformar todas as ligações **is-contained-by** em ligações **contains** invertidas e todas as ligações **generalization** em ligações **specialization** invertidas. Com estas alterações, chegou-se ao número final de 436 arestas no grafo distribuídas entre três tipos: 385 **contains** (88,3%), 28 **is-alternative** (6,4%) e 23 **specialization** (5,3%).

⁶Os tipos de relações descritos aqui, embora mais específicos e detalhados, foram baseados nos tipos previstos na definição da linguagem PLML [Fincher 2003] e no trabalho de van Welie e van der Veer [Van Welie and Van der Veer 2003] (ver Seção 2.2).

⁷<https://github.com/diegomrosa/HciPatternConverter>

4.4 Geração das visualizações

O arquivo GEXF produzido pela execução da ferramenta pode ser utilizado como entrada em aplicações de visualização de grafos, tais como Gephi e InfraNodus. Normalmente, estas ferramentas permitem uma grande variedade de visualizações de diagramas de rede e incluem a possibilidade de configuração de características dos nós e arestas tais como tamanhos, cores e etiquetas. As visualizações geradas podem ser utilizadas para a realização de análises sobre a linguagem ou podem dar origem a figuras para serem utilizadas em artigos, livros ou [websites](#).

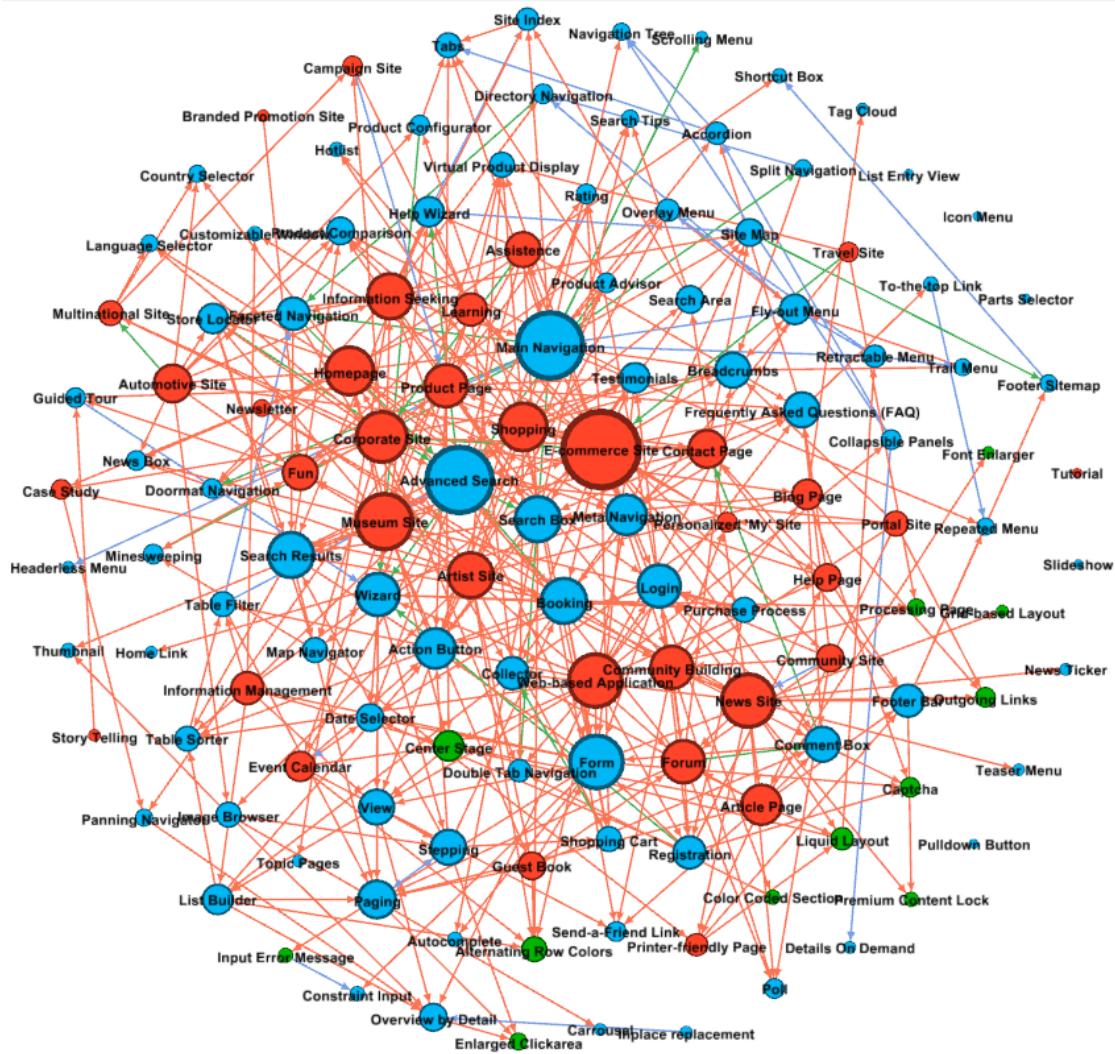


Fig. 4. Visualização gerada utilizando o software Gephi.

Nas visualizações geradas com o Gephi (ver Fig. 4 e Fig. 5), os nós (círculos) representam os padrões, que são devidamente identificados por etiquetas, enquanto as arestas em formato de seta representam as relações entre os padrões. Na visualização da Fig. 4, os nós estão equidistantes, sendo que os nós com maior número de

relações foram posicionados mais ao centro e os nós com menos ligações foram posicionados na periferia do diagrama. Esta visualização é compacta e, ao mesmo tempo, permite uma fácil identificação dos nós. Ainda na Fig. 4, o tamanho dos nós é proporcional à quantidade de relações que o padrão em questão possui. É possível alterar o tamanho dos nós de forma a considerar apenas as relações de saída ou de entrada.

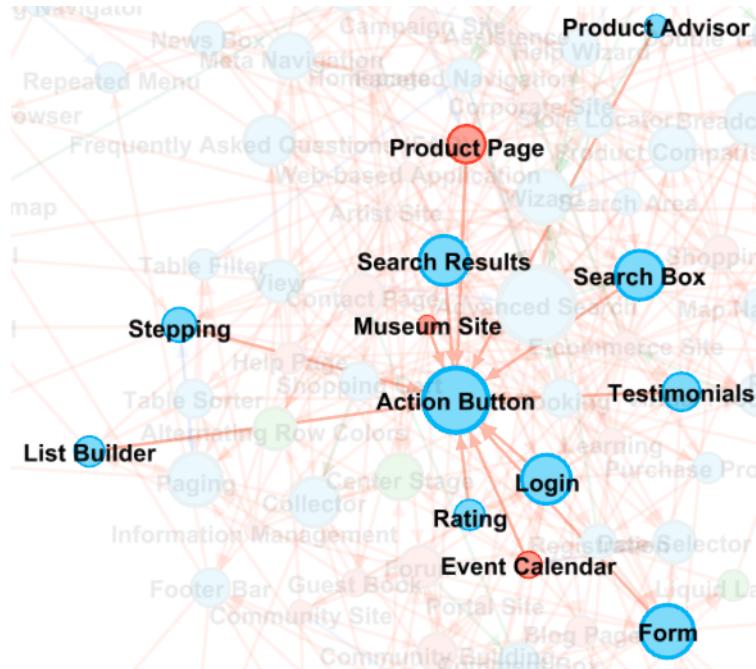


Fig. 5. Visualização no Gephi produzida ao selecionar um nó (representando um padrão).

As cores de cada nó nas visualizações apresentadas estão de acordo com a categoria de cada padrão (atribuídas originalmente pelo autor da linguagem): azul, vermelho e verde representam **necessidades do usuário, contexto do projeto e necessidades da aplicação** respectivamente. Já a cor das setas indica o tipo de relação: laranja, azul e verde indicam contains, is-alternative e specialization respectivamente. A direção das setas indica o sentido da ligação. A Fig. 5 apresenta a visualização quando o usuário seleciona um dos nós no grafo: o padrão selecionado e suas relações são exibidos em destaque.

Utilizando o mesmo arquivo GEXF, foram geradas visualizações na plataforma InfraNodus, como apresentado em Fig. 6 e Fig. 7. Esta plataforma permite um menor controle sobre a visualização das estruturas, mas a sua utilização é bastante intuitiva para o usuário. No InfraNodus, o tamanho dos nós também é proporcional ao número de ligações e as cores dos nós seguem as mesmas regras da visualização no Gephi. As arestas não possuem direção, ocultando o sentido das ligações. Assim como no Gephi, também é possível selecionar um nó para visualização apenas deste e dos nós conectados. Este software também permite a criação de novos nós e a produção de textos explicativos a partir da ferramenta *AI Insight*⁸ que faz uso do modelo GPT-3⁹.

⁸<https://infranodus.com/use-case/brainstorming>

⁹Generative Pre-trained Transformer 3: modelo de linguagem que usa aprendizado profundo para produzir texto semelhante ao humano.

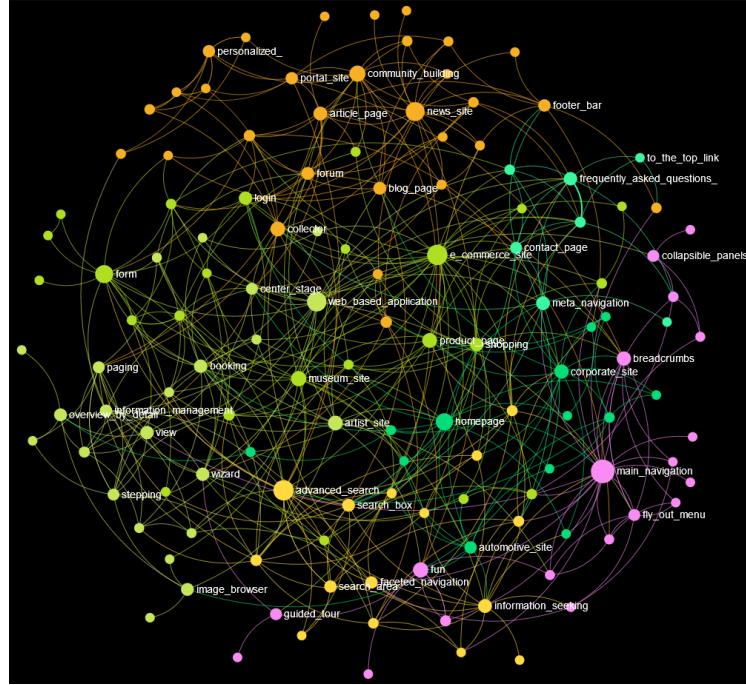


Fig. 6. Diagrama de rede gerado utilizando a plataforma InfraNodus.

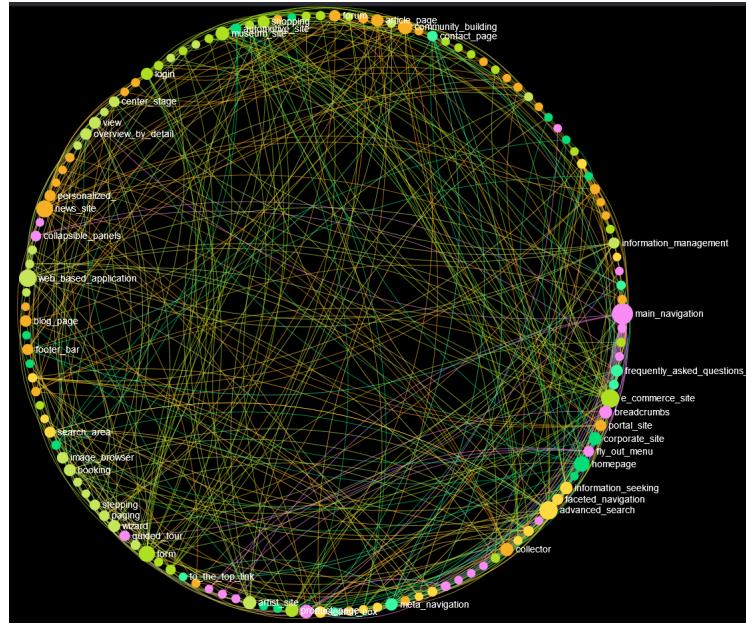


Fig. 7. Diagrama de cordas gerado utilizando a plataforma InfraNodus.

5. CONCLUSÃO

Padrões de projeto são utilizados em várias áreas, incluindo a Arquitetura, Engenharia de Software e IHC. A visualização de linguagens de padrões de IHC através de diagramas de rede facilita a compreensão e a navegação da linguagem. Neste trabalho, foi desenvolvida uma ferramenta para conversão dos arquivos de descrição dos padrões (em formato PLML) para um arquivo de descrição de grafo (GEXF). Os passos para a visualização de linguagens através do uso da ferramenta originaram um *framework*, o qual permite que linguagens de padrões no formato PLML sejam visualizadas em sistemas modernos e flexíveis, como Gephi e InfraNodus. Várias visualizações da biblioteca Welie.com, contendo 132 padrões de interação, foram apresentadas, demonstrando o potencial da solução. A solução apresentada pode ser generalizada para a visualização de qualquer linguagem de padrões descrita no formato PLML.

Uma das limitações do *framework* proposto é o fato de exigir a entrada da linguagem no formato PLML. Ao longo da história das linguagens de padrões, vários tipos de estrutura foram propostos e cada autor costuma ter a sua preferência na hora de descrever os padrões. Um trabalho futuro seria ampliar a capacidade da ferramenta, fazendo-a aceitar mais formatos de entrada. Também propomos, como trabalho futuro, testar o *framework* com outras ferramentas de visualização.

ACKNOWLEDGMENTS

Isabel Harb Manssour gostaria de agradecer o apoio financeiro da bolsa CNPq - Brasil (308456/2020-3).

REFERENCES

- Christopher Alexander. 1977. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, New York, NY.
- Elisabeth Bayle, Rachel Bellamy, George Casaday, Thomas Erickson, Sally Fincher, Beki Grinter, Ben Gross, Diane Lehder, Hans Marmolin, Brian Moore, and others. 1998. Putting it all together: towards a pattern language for interaction design: A CHI 97 workshop. *ACM SIGCHI Bulletin* 30, 1 (1998), 17–23. DOI:<http://dx.doi.org/10.1145/280571.280580>
- Jan O. Borchers. 2000. A Pattern Approach to Interaction Design. In *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS '00)*. Association for Computing Machinery, New York, NY, USA, 369–378. DOI:<http://dx.doi.org/10.1145/347642.347795>
- Sally Fincher. 2003. CHI 2003 Workshop Report - Perspectives on HCI Patterns: Concepts and tools (introducing PLML). *Interfaces* 56, Autumn 2003 (2003), 26–28.
- Sally Fincher, Janet Finlay, Sharon Greene, Lauretta Jones, Paul Matchen, John Thomas, and Pedro J. Molina. 2003. Perspectives on HCI patterns: concepts and tools. In *CHI'03 Extended Abstracts on Human Factors in Computing Systems*. 1044–1045. DOI:<http://dx.doi.org/10.1145/765891.766140>
- Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. 1995. *Design Patterns: elements of reusable object-oriented software*. Addison Wesley, Boston, MA.
- Richard Griffiths, Lyn Pemberton, Jan Borchers, and Adam Stork. 2000. Pattern languages for interaction design: Building momentum. In *CHI'00 Extended Abstracts on Human Factors in Computing Systems*. 363–363. DOI:<http://dx.doi.org/10.1145/633292.633510>
- Yunmi Park. 2015. The network of patterns: creating a design guide using Christopher Alexander's pattern language. *Environment and Planning B: Planning and Design* 42, 4 (2015), 593–614.
- Wolfram Schobert and Till Schümmer. 2006. Supporting Pattern Language Visualization with CoPE. In *11th European Conference on Pattern Languages of Programs (EuroPLoP)*. 343–358.
- Till Schümmer, Jan Borchers, John C Thomas, and Uwe Zdun. 2004. Human-computer-human interaction patterns: workshop on the human role in HCI patterns. In *CHI'04 Extended Abstracts on Human Factors in Computing Systems*. 1721–1722. DOI:<http://dx.doi.org/10.1145/985921.986200>
- Martijn Van Welie, Kevin Mullet, and Paul McInerney. 2002. Patterns in practice: a workshop for UI designers. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems*. 908–909. DOI:<http://dx.doi.org/10.1145/506443.506653>
- Martijn Van Welie and Gerrit C Van der Veer. 2003. Pattern languages in interaction design: Structure and organization. In *Proceedings of Interact*, Vol. 3. 1–5.

Received August 2022; revised October 2022; accepted February 2023