

# Um framework para visualização de linguagens de padrões de IHC através de diagramas de rede

DIEGO MOREIRA DA ROSA\*, ANDREA GNECCO, MILENE SILVEIRA, CHRISTIAN MATTJIE, RODRIGO C. BARROS, SOFIA APUZZO, MARCIO SARROGLIA PINHO, and ISABEL MANS-SOUR, Pontifical Catholic University of Rio Grande do Sul, Brazil

Definidos no final da década de 1970 pelo arquiteto Christopher Alexander, os conceitos de padrões e linguagem de padrões têm sido adotados por especialistas em diversas áreas, incluindo a Engenharia de Software e a Interação Humano-Computador (IHC). A estrutura hierárquica das linguagens de padrões prevista por Alexander favorece a sua representação através de diagramas de rede, prática esta comum entre pesquisadores de IHC. Apesar das vantagens do uso de grafos para elicitar as inter-relações de padrões dentro de uma linguagem, não foi possível encontrar um método estruturado e documentado para a visualização de linguagens de padrões em ferramentas modernas de diagrama de rede. Neste trabalho, apresentamos um framework que descreve as etapas para a visualização de linguagens de padrões de IHC através de diagramas de rede. Como parte do framework, foi desenvolvida uma ferramenta capaz de converter arquivos XML de descrição de padrões em um arquivo de descrição de grafo pronto para ser interpretado por aplicações de visualização ricas em funcionalidades. O framework foi testado com os dados da Welie.com, uma biblioteca online contendo 132 padrões de interação, e o grafo resultante foi visualizado em duas aplicações conhecidas para o desenho de estruturas de rede (Gephi e InfraNodus). Os resultados indicam que a solução proposta compõe uma estratégia viável para a visualização de linguagens de padrões.

CCS Concepts: • **Human-centered computing** → **Interactive systems and tools**; **Visualization techniques**.

Additional Key Words and Phrases: visualização, linguagem de padrões, IHC, diagrama de rede, grafo

## ACM Reference Format:

Diego Moreira da Rosa, Andrea Gnecco, Milene Silveira, Christian Mattjie, Rodrigo C. Barros, Sofia Apuzzo, Marcio Sarroglia Pinho, and Isabel Manssour. 2018. Um framework para visualização de linguagens de padrões de IHC através de diagramas de rede. In *Proceedings of the SugarLoafPLOP '22, October 18, 2022, online*. ACM, New York, NY, USA, 11 pages. <https://doi.org/XXXXXXX.XXXXXXX>

## 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. O conceito foi definido em 1977 pelo arquiteto e urbanista Christopher Alexander em sua obra seminal *A Pattern Language* [1]. Com o passar dos anos, o conceito foi adotado por especialistas e pesquisadores de outras áreas, incluindo a Engenharia de Software [6] e a Interação Humano-Computador (IHC) [2, 3]. 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.

---

\*Also professor at the Federal Institute of Rio Grande do Sul (IFRS).

---

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. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2018 Association for Computing Machinery.

Manuscript submitted to ACM

53 De acordo com Alexander, «nenhum padrão é uma entidade isolada» e «cada padrão pode existir no mundo, apenas  
54 na medida em que ele é apoiado por outros padrões: os padrões maiores nos quais ele está inserido, os padrões de mesmo  
55 tamanho que o rodeiam, e os padrões menores que estão inseridos nele» [1]. Um conjunto de padrões inter-relacionados  
56 e associados a um determinado contexto forma uma linguagem de padrões. Já a característica hierárquica destas relações  
57 faz com que as linguagens tenham a estrutura de uma rede [1]. Desde o início da adoção das linguagens de padrões  
58 na área de IHC, esta característica foi reconhecida e destacada, sendo comum entre os pesquisadores a representação  
59 visual das linguagens através de diagramas de rede (ver Seção 2).  
60

61  
62 O uso de grafos para representar visualmente as linguagens de padrões permite que sejam facilmente identificadas  
63 as inter-relações entre os padrões e a estrutura hierárquica existente entre eles, facilitando assim a compreensão e o  
64 uso da linguagem na prática. Apesar desta vantagem, não foram encontrados métodos estruturados e devidamente  
65 documentados para a visualização de linguagens de padrões através de aplicações modernas para desenho de diagrama  
66 de rede, como por exemplo Gephi<sup>1</sup> e InfraNodus<sup>2</sup>. O objetivo deste trabalho é propor um *framework*, ou uma sequência  
67 de passos estruturados, para a visualização de linguagens de padrões de IHC em ferramentas de diagrama de rede.  
68

69 Como parte do *framework* proposto, foi desenvolvida uma ferramenta que converte automaticamente arquivos de  
70 descrição de padrões em um arquivo de descrição de grafo pronto para ser utilizado como entrada nas aplicações de  
71 visualização. Como fonte de dados para os testes, foi utilizada a biblioteca *online Welie.com*<sup>3</sup>, uma coleção de 132  
72 padrões de interação, a qual inclui as descrições dos padrões em formato PLML (*Pattern Language Markup Language*)<sup>4</sup>.  
73 Posteriormente, o arquivo GEXF (*Graph Exchange XML Format*)<sup>5</sup> resultante da execução da ferramenta e contendo a  
74 descrição do grafo foi utilizado para produzir visualizações nas aplicações Gephi e InfraNodus, amplamente utilizadas  
75 para o desenho de grafos. Os resultados obtidos indicam que a sequência de passos utilizada, incluindo o processamento  
76 dos arquivos através da ferramenta, compõem um *framework* viável para a visualização de linguagens de padrões de  
77 IHC através de diagramas de rede.  
78

79 Este artigo é composto de cinco seções incluindo esta introdução. A Seção 2 apresenta trabalhos relacionados e outras  
80 definições a respeito de padrões e visualização de linguagens de padrões. A metodologia do trabalho é apresentada na  
81 Seção 3. A Seção 4 descreve o *framework* e os resultados obtidos, enquanto a Seção 5 apresenta considerações finais e  
82 trabalhos futuros.  
83  
84  
85

## 86 2 TRABALHOS RELACIONADOS

87 Nesta seção, são apresentados quatro trabalhos relacionados ao tema da visualização de linguagens de padrões. O  
88 primeiro estudo apresenta uma revisão histórica e uma proposta de definição formal para linguagens de padrões no  
89 contexto do *design* de interação [3]. Já o segundo trabalho detalha, no mesmo contexto, a organização das linguagens de  
90 padrões em estruturas de rede [12]. Uma ferramenta básica para a visualização de linguagens de padrões é apresentada  
91 no terceiro estudo [9], enquanto o quarto artigo apresenta uma proposta de aplicação de análise de redes sociais a uma  
92 rede de padrões [8]. Por fim, esta seção encerra-se com uma discussão das relações entre os trabalhos mencionados  
93 anteriormente e o presente estudo.  
94  
95  
96

97  
98  
99 <sup>1</sup><https://gephi.org/>

100 <sup>2</sup><https://infranodus.com/>

101 <sup>3</sup><http://www.welie.com/>

102 <sup>4</sup>Ambas PLML e GEXF são linguagens de marcação baseadas na linguagem XML (*eXtensible Markup Language*).

103 <sup>5</sup>Veja a nota de rodapé anterior.  
104

## 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 [3]. 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 [2, 5, 7, 10, 11].

No mesmo trabalho, Borchers define quais seriam os atributos básicos para a descrição de um padrão: **nome**, **ranking** (ou nível de confiança), **ilustração**, **problema**, **forças**, **exemplos**, **solução** e **diagrama** [3]. 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.

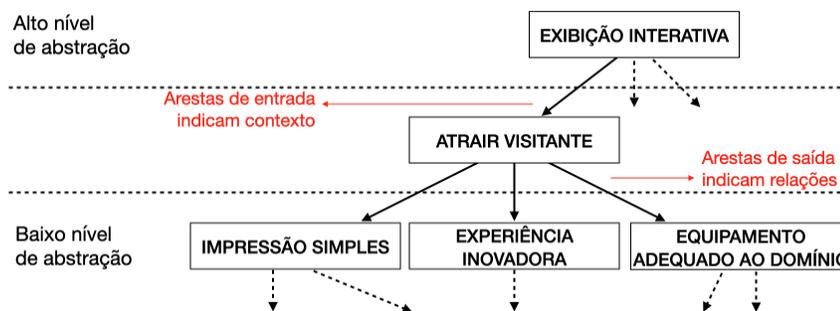


Fig. 1. A estrutura de rede proposta por Borchers para linguagens de padrões (adaptada do artigo original [3])

## 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 [12]. Embasados pelo texto original de Alexander [1], 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 nível aos padrões de mais baixo nível, ou ainda, «do *design* de cidades detalhando até o *design* do assento da janela, uma hierarquia de escala» [12]. 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:

- 157 • **Agregação:** utilizada para descrever relações todo-parte, ou seja, quando um padrão maior contém ou agrega  
158 outros padrões menores (relação “tem-um”).
- 159 • **Especialização:** utilizada nos casos em que um padrão é uma versão especializada de outro padrão, ou seja,  
160 herda características de um padrão mais genérico e adiciona opções estendidas (relação “é-um”).
- 161 • **Associação:** utilizada para descrever casos em que padrões são utilizados em contextos semelhantes ou fornecem  
162 soluções alternativas para um mesmo tipo de problemas (relação “está-relacionado-a”).  
163  
164

165 Ainda no mesmo trabalho, van Welie e van der Veer apresentam a estrutura e organização da biblioteca de padrões  
166 para *web design* Welie.com (recém criada na época e a mesma selecionada para a realização dos testes no presente  
167 estudo) [12]. No diagrama de rede que apresenta a estrutura da linguagem, os autores destacam as diferentes camadas  
168 que podem ser identificadas quando os padrões são organizados do mais alto para o mais baixo nível dentro da hierarquia.  
169

### 170 2.3 Visualização de linguagens de padrões

171 Embora descrevam como formalizar e organizar a estrutura de rede das linguagens de padrões de IHC, os trabalhos  
172 anteriores não abordam a questão do desenho do diagrama de rede. Uma das primeiras ferramentas de software para a  
173 visualização de linguagens de padrões foi proposta por Schobert e Schümmer em 2006 [9]. Os autores destacam a ausência,  
174 na época, de uma ferramenta que auxiliasse a criação das descrições textuais e do mapa de padrões simultaneamente. A  
175 partir disso, os autores propõem a ferramenta *Collaborative Pattern Editor* (CoPE) para preencher esta lacuna.  
176

177 Nesta ferramenta, os diagramas são gerados automaticamente a partir das descrições dos padrões, as quais podem  
178 ser escritas na sintaxe Wiki em um editor de texto ou importadas de arquivos XML no formato PLML. A ferramenta  
179 permite ainda que os autores trabalhem colaborativamente nas linguagens de padrões. Cores podem ser atribuídas  
180 manualmente a regiões do diagrama de forma a explicitar melhor as relações entre os padrões.  
181  
182

### 183 2.4 Análise de rede aplicada a linguagens de padrões

184 Uma proposta de visualização para a linguagem de padrões de Alexander [1] através do método de Social Network  
185 Analysis (SNA) é o foco do estudo de Park [8]. A partir da construção de um mapa de rede (*Network Map*) para os  
186 padrões, o método SNA foi executado com três diferentes critérios de importância: grau, proximidade, e intermediação.  
187 Park implementou uma pontuação para cada um destes critérios dentro da linguagem utilizada, obtendo, ao final da  
188 análise, conjuntos de padrões com distintas pontuações.  
189

190 O resultado desta classificação pode ser visualizado em forma de uma rede de padrões interconectados, na qual:  
191 padrões de maior pontuação para o critério grau possuem mais conexões; padrões de maior pontuação para o critério  
192 proximidade estão mais próximos uns dos outros; e, por último, padrões com maiores pontuações para o critério de  
193 intermediação, estão em posições mais centralizadas do grafo. Ao mesmo tempo, padrões com maior pontuação geral  
194 (considerando os três critérios), foram categorizados como padrões-chave.  
195  
196

### 197 2.5 Discussão sobre os trabalhos relacionados

198 A análise dos três primeiros artigos apresentados indica uma ênfase dos desenvolvedores de padrões de IHC na  
199 representação visual das linguagens [3, 9, 12]. De fato, diagramas de rede são comuns em estudos de IHC que propõem  
200 linguagens de padrões. Embora Alexander já tivesse previsto as inter-relações entre os padrões e uma organização  
201 hierárquica das linguagens, a mesma ênfase na representação visual através de grafos não é observada, por exemplo,  
202  
203  
204  
205  
206  
207  
208

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 [9]. 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 [8].

### 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.

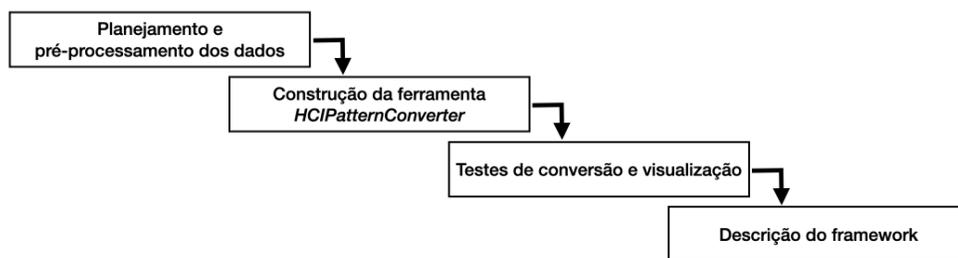


Fig. 2. Metodologia do estudo.

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 para um arquivo de descrição de grafo. Uma pesquisa exploratória revelou as duas aplicações para visualização do grafo através de diagramas de rede. 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. A linguagem Java foi escolhida por ser de conhecimento dos autores e por possuir amplo suporte à manipulação de documentos XML.

A validação da ferramenta ocorreu através do processamento dos 132 arquivos PLML 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.

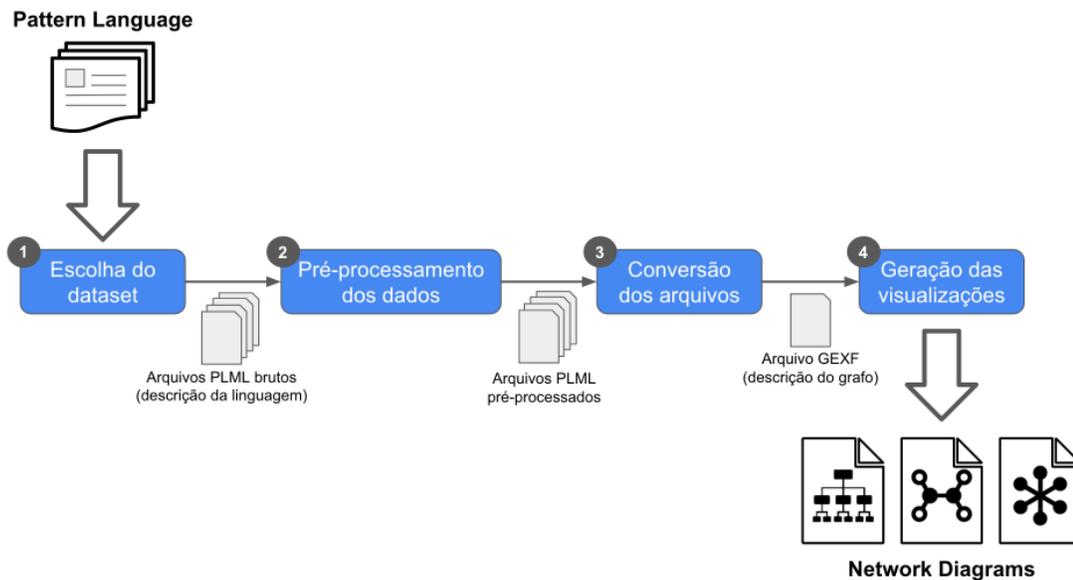


Fig. 3. Framework para visualização de linguagens de padrões de IHC através de diagramas de rede.

### 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 [4]. 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;

313 já o atributo `label` oferece um rótulo legível para o padrão. Para que a ferramenta desenvolvida possa reconhecer  
314 devidamente as ligações e transformar os vários arquivos PLML em um grafo, é estritamente necessário que elementos  
315 `<pattern-link>` estejam bem formatados, com atenção especial para os atributos `type` e `patternID`. Vale ressaltar  
316 que, caso os arquivos sigam corretamente o formato PLML, a etapa de pré-processamento pode não ser necessária.  
317

318 No caso dos arquivos da Welie.com, dois problemas principais foram identificados nas ligações entre os padrões.  
319 Primeiramente, os elementos `<pattern-link>` não incluíam o atributo `type`. Como estes elementos eram utilizados a  
320 todo momento para referenciar outros padrões em meio ao texto, não era possível identificar o tipo de relação que o  
321 autor havia idealizado para cada ligação. Inclusive, o elemento `<pattern-link>` era utilizado em situações nas quais o  
322 autor pretendia informar que o padrão referenciado **não** estava relacionado com o padrão do arquivo. Para contornar  
323 esta situação, foram definidos sete tipos de relações para o atributo `type`<sup>6</sup>:

- 325 a. **is-alternative**: Indica que o padrão atual é uma solução alternativa para o mesmo problema abordado pelo  
326 padrão referenciado.
- 327
- 328 b. **is-contained-by**: Indica que o padrão atual é de um nível mais baixo e é usado (junto com outros) para  
329 instanciar o padrão referenciado (de nível mais alto).
- 330
- 331 c. **contains**: Significa o oposto de `is-contained-by`, ou seja, o padrão atual é de um nível mais alto e usa o  
332 padrão referenciado (junto com outros) para ser instanciado.
- 333
- 334 d. **specialization**: Indica que padrão atual é uma solução semelhante porém mais especializada que o padrão  
335 referenciado.
- 336 e. **generalization**: Significa o oposto de `specialization`, ou seja, o padrão atual é uma solução mais genérica  
337 que o padrão referenciado.
- 338 f. **is-alias**: Indica que o padrão atual é idêntico ao outro padrão (dois padrões iguais apenas com nomes  
339 diferentes).
- 340 g. **not-related**: Significa que a ligação não indica nenhuma relação específica entre os dois padrões.

342 A principal tarefa do pré-processamento consistiu em uma leitura dos 132 arquivos PLML e do preenchimento manual  
343 do valor do atributo `type` para cada um dos elementos `<pattern-link>` encontrados. Foi necessário deduzir o tipo  
344 mais adequado para as ligações a partir do contexto da descrição textual de cada padrão.  
345

346 Um segundo problema encontrado foi causado por alguns erros de digitação nos atributos `patternID` dos elemen-  
347 tos `<pattern-link>`. Estes erros impediam a correta identificação do padrão referenciado. Novamente as ligações  
348 problemáticas foram corrigidas manualmente, tornando os arquivos prontos para o processo de conversão.  
349

### 350 4.3 Conversão dos arquivos

352 A ferramenta *HCIPatternConverter* foi desenvolvida em linguagem Java para a conversão dos arquivos de descrição dos  
353 padrões. O código da ferramenta está disponível na plataforma GitHub<sup>7</sup>. A ferramenta faz a leitura do conjunto de  
354 arquivos PLML, realiza uma validação das ligações entre os padrões (e.g., elimina ligações duplicadas) e produz um  
355 arquivo GEXF. O formato GEXF é aceito como entrada por diversas ferramentas de visualização de grafos como as  
356 descritas na Seção 4.4.  
357

358 Em um primeiro momento, a ferramenta foi programada para eliminar: ligações do tipo `not-related`; ligações  
359 do tipo `is-alias` que referenciavam padrões de outras bibliotecas; e ligações duplicadas (ligações com tipo, padrão  
360

361 <sup>6</sup>Os tipos de relações descritos aqui, embora mais específicos e detalhados, foram baseados nos tipos previstos na definição da linguagem PLML [4] e no  
362 trabalho de van Welie e van der Veer [12] (ver Seção 2.2).

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

origem e padrão destino idênticos). Ainda assim, foi identificado um grande número de ligações (469), o que acabou 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%).

#### 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*.

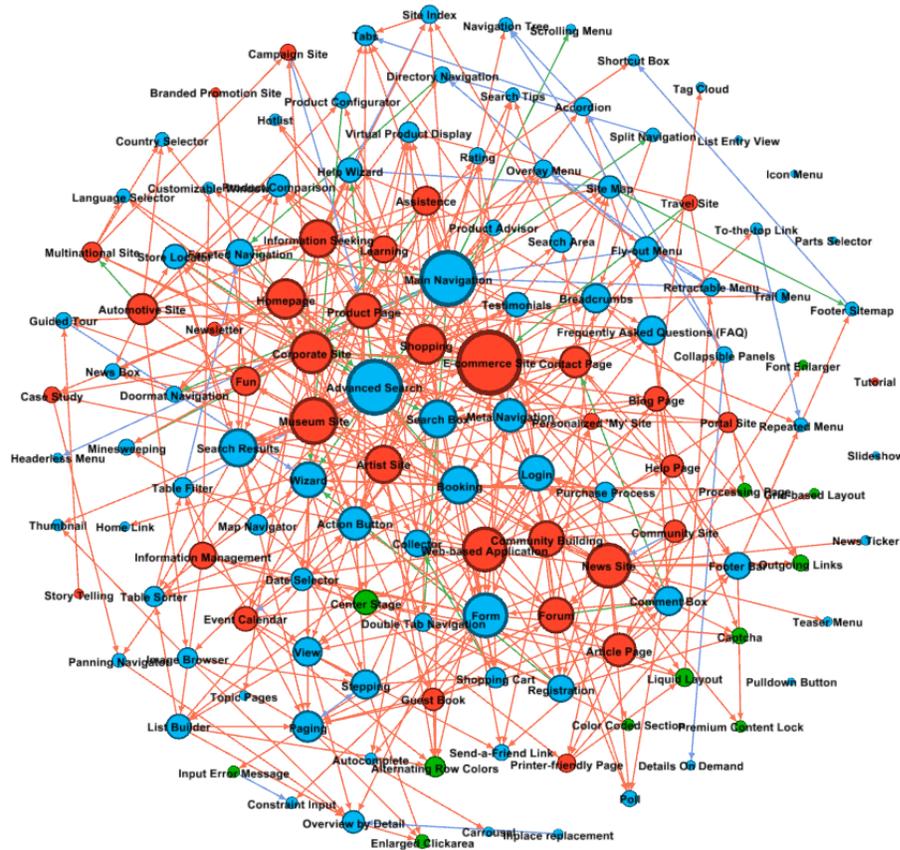
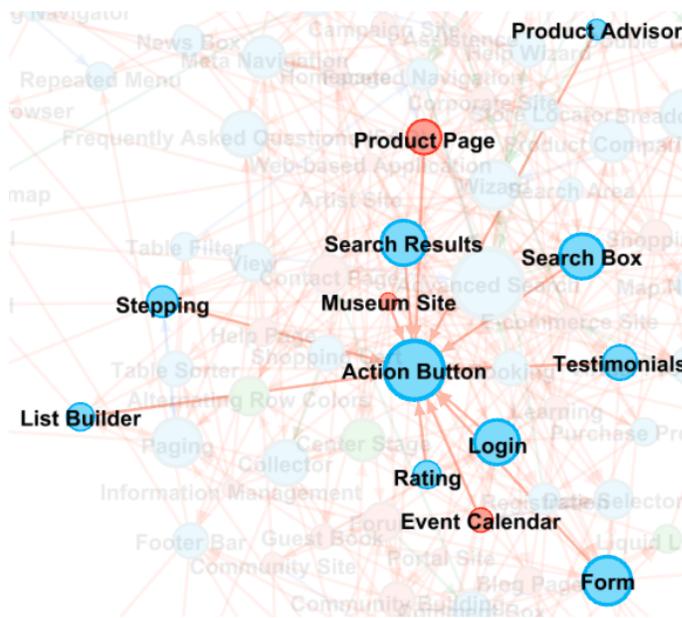


Fig. 4. Visualização gerada utilizando o software Gephi.

Nas visualizações da biblioteca Welie.com geradas com o Gephi (ver Figuras 4 e 5), os nós (círculos) representam os padrões, que são devidamente identificados por etiquetas, enquanto as arestas em formato de seta representam as

417 relações entre os padrões. Na visualização da Fig. 4, os nós estão equidistantes, sendo que os nós com maior número de  
418 relações foram posicionados mais ao centro e os nós com menos ligações foram posicionados na periferia do diagrama.  
419 Esta visualização é compacta e, ao mesmo tempo, permite uma fácil identificação dos nós.

420 Ainda na Fig. 4, o tamanho dos nós é proporcional à quantidade de relações que o padrão em questão possui. É  
421 possível alterar o tamanho dos nós de forma a considerar apenas as relações de saída ou de entrada. As cores de cada nó  
422 nas visualizações apresentadas estão de acordo com a categoria de cada padrão: azul, vermelho e verde representam  
423 **necessidades do usuário**, **contexto do projeto** e **necessidades da aplicação** respectivamente. Já a cor das setas  
424 indica o tipo de relação: laranja, azul e verde indicam *contains*, *is-alternative* e *specialization* respectivamente.  
425 A direção das setas indica o sentido da ligação. A Fig. 5 apresenta a visualização quando o usuário seleciona um dos nós  
426 no grafo: o padrão selecionado e suas relações são exibidos em destaque.



452 Fig. 5. Visualização no Gephi produzida ao selecionar um nó/padrão.

453  
454  
455 Utilizando o mesmo arquivo GEXF, foram geradas visualizações na plataforma InfraNodus, como apresentado nas  
456 Figuras 6 e 7. Esta plataforma permite um menor controle sobre visualização das estruturas, mas a utilização é bastante  
457 simplificada.

458 No InfraNodus, o tamanho dos nós também é proporcional ao número de ligações e as cores dos nós seguem as  
459 mesmas regras da visualização no Gephi. As arestas não tem direção, ocultando o sentido das ligações. Assim como no  
460 Gephi, também é possível selecionar um nó para visualização apenas deste e dos nós conectados. Este software também  
461 permite a criação de novos nós e a produção de textos explicativos a partir da ferramenta *AI Insight*<sup>8</sup> que faz uso do  
462 modelo Generative Pre-trained Transformer 3 (GPT-3)<sup>9</sup>.

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

467 <sup>9</sup><https://openai.com/api/>



## 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).

## REFERÊNCIAS

- [1] Christopher Alexander. 1977. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, New York, NY.
- [2] Elisabeth Bayle, Rachel Bellamy, George Casaday, Thomas Erickson, Sally Fincher, Beki Grinter, Ben Gross, Diane Lehder, Hans Marmolin, Brian Moore, et al. 1998. Putting it all together: towards a pattern language for interaction design: A CHI 97 workshop. *ACM SIGCHI Bulletin* 30, 1 (1998), 17–23. <https://doi.org/10.1145/280571.280580>
- [3] 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* (New York City, New York, USA) (*DIS '00*). Association for Computing Machinery, New York, NY, USA, 369–378. <https://doi.org/10.1145/347642.347795>
- [4] Sally Fincher. 2003. CHI 2003 Workshop Report - Perspectives on HCI Patterns: Concepts and tools (introducing PLML). *Interfaces* 56, Autumn 2003 (2003), 26–28.
- [5] 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. <https://doi.org/10.1145/765891.766140>
- [6] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. 1995. *Design Patterns: elements of reusable object-oriented software*. Addison Wesley, Boston, MA.
- [7] 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. <https://doi.org/10.1145/633292.633510>
- [8] 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.
- [9] Wolfram Schobert and Till Schümmer. 2006. Supporting Pattern Language Visualization with CoPE. In *11th European Conference on Pattern Languages of Programs (EuroPLoP)*. 343–358.
- [10] 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. <https://doi.org/10.1145/985921.986200>
- [11] 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. <https://doi.org/10.1145/506443.506653>
- [12] 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.