

## 7 Conclusões

O estudo realizado nesse trabalho visa abrir novos caminhos para o apoio à elaboração de modelos de IHC em editores visuais baseados em metamodelos, evidenciando o seu impacto no comportamento do usuário. O objetivo da ferramenta proposta nesse estudo é permitir a modelagem da interação com assessoria da verificação de sintaxe, além de representar um importante passo para a popularização das linguagens de IHC. Este capítulo apresenta as contribuições desse trabalho, algumas lições aprendidas, e aponta para novas questões que devem ser endereçadas por trabalhos futuros.

### 7.1 Contribuições

Este trabalho procurou contribuir com a abordagem de projeto de interação humano-computador baseado em metamodelos, à medida que buscou aprimorar a representação de linguagem de regras. Foi apresentada uma abordagem sobre regras de sintaxe diferente das atuais. Ela se baseia em conceitos mais simples de adição e exclusão de relacionamentos entre elementos que atendem aos requisitos básicos de uma linguagem visual da área de IHC.

Na área dos metamodelos, foi possível definir as três linguagens de IHC abordadas nesse trabalho a partir das definições de nós e arestas, permitindo a transferência de gráficos vetoriais entre a ferramenta Dia e a proposta nesse trabalho.

Comparando a ferramenta proposta nesse trabalho com o programa Microsoft Visio [Microsoft, 2010] conseguimos criar uma linguagem de regras simples o suficiente para não exigir uma experiência do usuário em linguagens de programação. Também criamos uma ferramenta cuja linguagem não necessita de recompilação do editor de diagramas ao modelo requerido comparado do editor DiaMeta [Maier & Minas, 2009]. A linguagem de regras é mais simples e menos abrangente que a RuleML [RuleML, 2011], pois o foco do trabalho é contemplar apenas as linguagens de IHC focando nas necessidades básicas de criação de regras com possibilidade para futuras evoluções.

Comparado à MoLIC Designer [Sangiorgi, 2010], o editor proposto já contém algumas funcionalidades não existentes ou incompletas como: copiar e colar, criação rápida de conectores e detecção de erros de sintaxe. Porém, a representação da MoLIC no editor proposto ainda contém problemas na representação do elemento Cena. Ele não consegue criar mais de uma caixa de texto para um elemento, algo que na MoLIC Designer é representado corretamente.

## **7.2 Trabalhos Futuros**

Nos trabalhos futuros são apontadas as novas questões que devem ser pesquisadas na área de metamodelos e na área de linguagem de regras. Novas abordagens devem ser estudadas em relação ao editor EdMeta e algumas funcionalidades existentes devem ser melhoradas. Estas dizem respeito a:

### **- Linguagens de IHC**

Nessa dissertação foi feita uma avaliação do editor com usuários utilizando a linguagem MoLIC. É interessante realizar novas avaliações focando nas outras linguagens da área de IHC abordadas aqui: CTT e Statecharts.

### **- Metamodelos**

É necessária uma pesquisa mais detalhada sobre a estrutura do metamodelo para abranger linguagens diversas, tanto dentro quanto fora da área de IHC. Para isso, novos estudos devem ser realizados para o metamodelo conseguir representar todos os elementos dessas linguagens.

### **- Linguagem de Regras**

Aprofundar mais o assunto sobre linguagens de regras, procurando aprimorar a linguagem para abranger modelos além da área de IHC. Também é necessário um aprimoramento na linguagem para aceitar diferentes níveis de regras que variam de regras obrigatórias a regras de boas práticas. Essa abordagem leva a novas pesquisas sobre aprendizado da linguagem e ampliação do uso da linguagem.

Uma pesquisa sobre regras semânticas da linguagem também seria muito interessante. O editor poderia aconselhar e alertar o usuário sobre conjunto de

elementos que são válidos sintaticamente mas que não fazem sentido semanticamente.

#### **- Realizar um estudo com outras linguagens de IHC**

Nesse trabalho foi realizada uma avaliação do editor utilizando a linguagem de interação MoLIC. O próximo passo seria realizar uma avaliação semelhante usando as outras duas linguagens apresentadas nessa dissertação: CTT e Statecharts. Uma avaliação do desempenho da ferramenta comparando ao estudo realizado com a MoLIC poderia ser interessante.

#### **- Trabalho colaborativo com programa de prototipação**

Outro trabalho futuro interessante consiste em importar o arquivo gerado por uma ferramenta de prototipação, permitindo associar telas e desenhos de protótipos a um modelo previamente definido em um metamodelo específico. No trabalho de Segura [2011] é apresentada uma ferramenta de prototipação (UISKEI) que permite definição do comportamento da interface com o usuário.

#### **- Melhorar as mensagens de erro**

Adicionar ao metamodelo as mensagens de erro referentes a cada regra criada. Isso permite ao criador da linguagem atribuir mensagens de erro que, na sua concepção, sejam mais expressivas do que as geradas automaticamente pelo sistema.

#### **- Melhorar a visualização dos alertas**

Adicionar uma janela que agregue todos os erros em formato de lista, indicando o erro e a explicação. Também incluir a possibilidade de se desligar o verificador de regras pela interface e ignorar um erro específico sinalizado pelo usuário. Outra adição bem vinda seria ter dois níveis de verificação: um para alerta e outro para erro.

#### **- Criação de múltiplas caixas para um elemento**

Esse aprimoramento é importante para a linguagem MoLIC. O elemento Cena necessita de duas caixas de texto para edição: uma para o tópico e outra para o diálogo. O editor de diagramas apresenta apenas uma caixa de texto para a inserção dessas duas informações.

**- Analisar o conteúdo dos campos de textos dos elementos**

Aumentar a abrangência da análise sintática da linguagem para os textos dos elementos, como por exemplo, alertar o usuário de uma aresta sem texto caso ele seja obrigatório.

**- Sugerir boas práticas para alguns alertas**

Foram percebidas durante o estudo situações em que o usuário não infringe as regras de sintaxe do modelo, porém os participantes não seguiam boas práticas específicas para a elaboração de um determinado modelo. Por isso, seria interessante apresentar um exemplo de boa prática para o usuário ao se detectar uma determinada ação feita por ele. Um exemplo disso seria o login de um site.

**- Aprimoramentos na usabilidade**

Problemas encontrados no decorrer dos testes nos alertaram para alguns pontos da usabilidade do software que devem ser revistos: revisar as funcionalidades do clique esquerdo do mouse durante a interação nos elementos para evitar confusão no arraste dos elementos; escolher os pontos de conexão dos elementos; adaptar o texto ao tamanho da caixa estabelecida; confirmar o texto editado com botão 'Enter' em vez de pular linha.