

## 6

### Conclusão

Pesquisamos formas de usuários finais de aplicações Web para grupos representarem e negociarem entre si, e com designers, alterações destas mesmas aplicações. Trabalhamos no contexto de especificação realizada por usuários finais (“End User Specification”) com um traço importante de que esta especificação resulta de um processo de negociação “online”, mediada pela própria tecnologia, entre os interessados (ie. o grupo que utiliza a aplicação).

#### 6.1

##### Sumário das Contribuições desta Pesquisa

Nossa principal contribuição é um modelo, teoricamente fundamentado na Engenharia Semiótica, para usuários finais descreverem e comunicarem ideias e intenções sobre modificações de sistemas.

Suas principais características são: (i) estruturar a comunicação entre os participantes e (ii) descrever a proposta de evolução de sistema a partir da linguagem de interface (UIL), uma linguagem conhecida pelo usuário final. O usuário pode com ela ‘apontar’ modificações, incluir e remover objetos de maneira simples e rápida. Além disso, pode inserir anotações associadas a elementos da interface. Outro diferencial desta linguagem de descrição é a possibilidade de representar signos dinâmicos (processos), através da associação de *scripts* de interações em páginas Web. Tudo isso é feito com intuito de comunicar uma proposta de melhoria no sistema.

Além disso, estamos contribuindo para a Engenharia Semiótica, uma vez que produzimos uma ferramenta para apoiar a evolução de sistemas através desta engenharia, ou seja, estamos oferecendo uma tecnologia para ajudar designers e co-designers a fazerem Engenharia Semiótica. Isso é inovador pelo desafio de concretizar os conceitos desta teoria e porque os estados atuais das ferramentas epistêmicas da Engenharia Semiótica ainda não conseguem introduzir seus conceitos, modelos e ontologia, na prática cotidiana.

Os resultados aqui apresentados serviram para identificar três pontos importantes: (i) a viabilidade de usar a linguagem de interface (UIL) para comunicar ideias sobre evolução de sistemas; (ii) a necessidade de ampliar o

vocabulário dessa linguagem, respeitando os limites de maneira que preserve a identidade da aplicação; e (iii) o caráter epistêmico do modelo proposto, visto que, encontramos nos estudos empíricos várias evidências de atos reflexivos dos emissores durante a elaboração da mensagem.

Estamos propondo duas linguagens (a Linguagem de Modificação e a Linguagem de Comunicação) que estão fechando o espectro de ambiguidade que uma mensagem expressa em linguagem natural tem. Consequentemente, estamos ajudando as pessoas entenderem melhor o que está sendo proposto. Com isso, esperamos contribuir para a comunidade de EUD, melhorando a comunicação entre designers e usuários.

Para avaliar o uso deste modelo implementamos parte dele na ferramenta primoTiWIM. Realizamos também dois estudos empíricos com ela. No primeiro observamos usuários finais, o que nos permitiu estudar a comunicabilidade e a facilidade de uso da primoTiWIM. Somente com apenas alguns minutos de treinamento, as pessoas conseguiram usá-la e falar de alterações no sistema sem precisar usar terminologias de engenharia de software. Porém, um resultado importante foi a constatação da dificuldade de pessoas que não são treinadas para desenvolver atividades de “design” em exercer a função de *designer* e a imaginar resultados futuros. Encontramos ainda, alguns problemas de comunicabilidade da ferramenta.

O segundo experimento nos permitiu avaliar o poder de expressão da parte do modelo que foi concretizada e as dificuldades/facilidades de uso do mesmo. Vimos que a diversidade de linguagens oferecidas aos usuários finais aumentou consideravelmente o poder de expressão do mesmo. Isto porque construímos um ambiente rico semioticamente falando, pois combinamos: (i) uma linguagem de especificação forte para *layout* de página, que cria uma semântica que não é ambígua (cada elemento dela tem um identificador); (ii) uma linguagem um pouco mais frouxa para definir comportamento dinâmico do sistema, que é a do CoScripter; e (iii) uma linguagem totalmente livre que é a linguagem natural. Ou seja, fizemos um misto de linguagens forte, intermediária e livre e estamos trabalhando toda esta diversidade em um mesmo ambiente.

Adicionalmente, identificamos que a metacomunicação do nosso modelo ainda não está tão boa e que é bastante complexa. Esta pesquisa apontou vários aspectos desta complexidade, como por exemplo, o problema da inter-referência entre as partes da mensagem, a dificuldade de representar múltiplas referências sobre um mesmo objeto e a diferenciação de papéis. Esses problemas sugerem algumas pesquisas futuras.

## 6.2

### Trabalhos Futuros

Os estudos empíricos do nosso modelo e o estudo de pesquisas correlatas mostraram que ainda há uma série de trabalhos futuros a serem realizados. Uma das evidências do Estudo Empírico II foi a necessidade de desenvolver um ambiente de comunicação assíncrona apropriado para este tipo de atividade online. Em particular, este ambiente tem de oferecer mecanismos sofisticados de referências cruzadas entre elementos que são comunicados em diferentes linguagens ou sistemas de representação, em diferentes momentos do tempo, com diferentes ênfases ou perspectivas. Portanto um trabalho futuro imediato, ainda usando a implementação que fizemos, é investigar a melhor forma de integrar a *primoTiWIM* a um ambiente de discussão. Outro trabalho futuro de curto prazo e que certamente contribuirá para a integração mencionada, é avaliar a reação de usuários finais representando signos dinâmicos através de *scripts* do *CoScripter*. Isto endereça o fato de que esta foi uma limitação do Estudo Empírico II e de que outra pesquisa (Bim, 2009) já relatou que signos dinâmicos são difíceis de conceituar. Na realidade, não é necessário que se use o *CoScripter* - outra ferramenta de gravação/programação de *scripts* pode e deve ser explorada para ver se é mais adequada.

Como a *UIL* original é um código compartilhado por todos os envolvidos na discussão, e que é através dela que a mensagem original de metacomunicação está codificada, seria ideal manter uma versão da interface original que seja acessível para fins de consulta e referência pelos diferentes agentes da discussão. Isto foi evidenciado no segundo Estudo Empírico II. Portanto, um trabalho a ser feito a mais longo prazo é estender o ambiente de elaboração da mensagem detalhada para manter a versão original e deixá-la acessível aos interlocutores. Adicionalmente, é interessante investigar a possibilidade disponibilizar as versões “intermediárias” e o rastro de evolução.

Outra pesquisa de médio prazo sugerida é expandir o poder de expressão da linguagem de Modificação. Os tipos de modificação de um sistema de significação classificados na Engenharia Semiótica nos fizeram refletir sobre o limite de expressão que usuários finais devem ter para expressar suas ideias de modificação de sistemas. Refletimos sobre a remoção de objetos e nos chamou atenção o fato de que se o usuário puder excluir livremente elementos da página então isto pode acarretar na remoção de signos “Essenciais”. Por outro lado se permitirmos a remoção de objetos livremente, estamos dando a possibilidade de ser dito: “Não quero falar sobre isso agora”. Ou seja, é semioticamente bastante estimulante pesquisar como significar a diferença entre “o que não

interessa” e “o que não existe”. Além disso, outra vantagem de não limitarmos a remoção de elementos é quando a ideia ainda está abstrata, neste casos pode acontecer uma “reflexão coletiva” na qual os participantes conjecturam possibilidades de reprojeto.

Diante destas questões a sugestão é estender a linguagem de modificação para:

- Definir quais signos da aplicação são impermeáveis;
- Definir signos ocasionais, ou seja, criar “blocos” de signos relacionados às funcionalidades que podem ser podadas do sistema;
- Habilitar e desabilitar a inclusão e remoção liberada de “tipos” de signos, inclusive impermeáveis.

As possibilidades de marcar signos essenciais e acidentais sugeridas aqui, podem ser bastante úteis no contexto de Design participativo em sistemas de grupo, visto que, as pessoas do grupo teriam oportunidade de discutir agrupamentos de significados, ou seja, eles poderiam definir coletivamente, com o apoio desta tecnologia, os limites de acesso de cada grupo. Esse é um ponto interessante a ser investigado em futuras pesquisas.

Um trabalho futuro dentro do contexto de realizar pesquisas teóricas para a Engenharia Semiótica é usar a TiWIM como ferramenta de treinamento da Engenharia Semiótica. Projetistas a usariam, por exemplo, para exercitar como expressar-se através de signos estáticos, dinâmicos e metalinguísticos. Podemos elaborar cenários onde potenciais usuários (receptores) tentariam interpretar as intenções de design expressas por projetistas (emissores).

Acreditamos que o modelo proposto pode ser útil para designers minimizarem problemas de engenharia de software vivenciados por usuários finais durante atividades de EUD, visto que estas pessoas não possuem formação nesta área mas precisam lidar com conceitos dela. Acreditamos que questões como tratamento de erros, sistema de ajuda e problemas de segurança podem ser identificadas e alertadas durante a conversa. Uma pesquisa interessante é avaliar a possibilidade de uso do modelo em conjunto com metodologias que promovem o Design Participativo.

No experimento com usuários finais, o uso de expressões típicas de programação de sistemas praticamente não foi evidenciado, já no experimento com designers vários jargões apareceram. Esse é um resultado importante porque já sabemos que precisamos analisar a fronteira terminológica entre designers e usuários. Como trabalho futuro sugerimos fazer um estudo empírico para captar a terminologia dos usuários finais. Uma ideia é pedir para ele narrar (via e-mail, por exemplo) para os designers como fazer uma dada tarefa

no sistema, em seguida pedir para os designers fazerem o mesmo e comparar os termos/expressões utilizados por ambos. O passo seguinte é realizar um experimento com ambos possibilitando uma conversa mais longa, com contra-propostas. O ideal de longo prazo seria implementar todo o modelo em um mesmo ambiente e expandir o poder de expressão da linguagem de modificação.