

4

“primoTiWIM” - Uma instanciação do Modelo

Neste capítulo apresentamos os resultados de experiências com a concretização de parte do modelo através da ferramenta primoTiWIM. Incluímos um exemplo ilustrativo de uso da ferramenta para discutir sobre modificações em um sistema de controle de processos acadêmicos. O propósito de concretização da ferramenta foi o de viabilizar duas medidas importantes para esta pesquisa: (a) uma prova de conceito, mostrando que e como o modelo pode ser instanciado; e (b) uma avaliação empírica das partes centrais de nossa contribuição para avançar a prática de EUD.

4.1

Concretização do Modelo

A mensagem do modelo conceitual possui três partes (como foi apresentado na Figura 3.3 do Capítulo 3): (i) cabeçalho, (ii) Conteúdo Geral e (iii) Conteúdo Detalhado. A seguir apresentamos como elas foram concretizadas.

Conteúdo Geral: Esta é a parte da mensagem que contém as informações: tipo de fala, estou no papel de, falando para [nome do papel] e descrição geral (textual) da mensagem. Decidimos utilizar recursos de correio eletrônico, simples, para realizar a comunicação assíncrona do modelo. Estas informações do modelo foram concretizadas em mensagem “estruturada” no corpo do email.

Cabeçalho: Parte da mensagem que contém informações de quem está enviando, quem está recebendo, assunto e data. Estas informações foram mapeadas para campos da ferramenta de email: assunto \Leftrightarrow assunto; data \Leftrightarrow data; emissor \Leftrightarrow remetente; receptor \Leftrightarrow destinatário.

Descrição Detalhada: Esta é a parte da mensagem que descreve a representação concreta da proposta de modificação, utilizando a Linguagem de Modificação (LM) do modelo. Ela foi concretizada em uma ferramenta WEB denominada primoTiWIM. O campo “anexo” da

ferramenta de email foi utilizado para vincular a descrição detalhada ao restante da mensagem (concretizada na ferramenta de email).

Vemos na Figura 4.1 que a implementação deste modelo para efeitos dos experimentos realizados envolveu dois ambientes, uma ferramenta de discussão na Web (email) e um ambiente para construção do detalhamento da proposta (primoTiWIM).

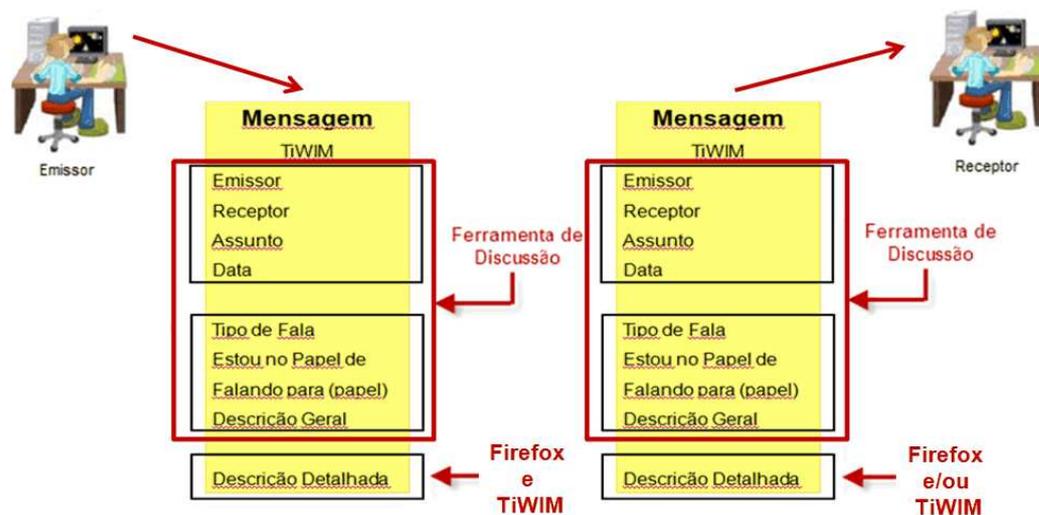


Figura 4.1: Emissão e Recepção de Mensagem

Nosso modelo foi parcialmente implementado como um *plug in* que funciona no navegador Firefox (Mozilla, 2010) e permite a alteração de elementos em páginas da Web (Figura 4.2). A implementação combina três sistemas de representação em uma única “linguagem de interação” para o usuário final. São eles: (i) a linguagem de interface (UIL), (ii) a linguagem natural, utilizada em anotações sobre elementos selecionados da UIL; e (iii) *scripts* construídos no CoScripter, um gravador de macros que permite usuários finais criarem e compartilharem *scripts* para executar interações na Web. A primoTiWIM permite ao usuário apontar, modificar, anotar, remover e copiar elementos da interface. Além disso, é possível expressar ‘ações’ do sistema (signos dinâmicos) através da associação com *scripts* do CoScripter.

Na Figura 4.2 vemos uma cópia da página do Google editada pela primoTiWIM. Do lado esquerdo da tela está a *toolbar* onde são mostradas todas as ações executadas nos elementos da interface. Este registro no momento é apenas uma representação passiva (i.e. destinada a ser interpretada pelo usuário apenas, e não pelo sistema). Porém, no futuro, devemos construir um interpretador para essas ações e apresentá-las aos usuários finais de forma mais legível, como por exemplo uma narrativa simples em linguagem natural. Estas ações servem para registrar todo o *rationale* da elaboração da proposta.

A *toolbar* da ferramenta tem duas abas, uma delas é uma interface para o CoScripter. Ela permite que, no CoScripter, o usuário grave navegações na Web as associe a algum elemento da interface que está sendo modificada. Esta é uma interessante maneira de representar o comportamento de elementos da interface (signos dinâmicos) na primoTiWIM.



Figura 4.2: Ferramenta primoTiWIM

Para esclarecer melhor esta ideia vamos dar um exemplo de como o CoScripter poderia ser usado para representar o comportamento de um novo elemento de interface, por exemplo, nesta página do Google. Vamos imaginar o seguinte cenário:

Suponhamos que Lina tem um canal de comunicação com a equipe de designers do Google e quer fazer uma sugestão de melhoria na página de busca. Lina envia um email à equipe dizendo: “Gostaria de ter uma opção para traduzir textos de inglês para português, na página principal do Google, pois utilizo essa funcionalidade frequentemente.”

Lina conhece o CoScripter e decide fazer um script para representar sua ideia de maneira mais concreta, a fim de mostrar à equipe do Google quantos passos ela economizaria caso existisse a funcionalidade sugerida. Então Lina grava o seguinte script:

Script do CoScripter

```

go to "http://www.google.com.br"
enter "house" into the textbox
copy the textbox
go to "http://www.google.com.br/"
click the "Ferramentas de idiomas" link
paste into the "Traduzir o texto" textbox
select "português" from the first "listbox"
click the first "Traduzir" button

```

Em seguida, Lina envia à equipe de design do Google. Ao recebê-la, os destinatários podem executar o script de Lina. O resultado é um “playback” do script, onde cada ação é realizada automaticamente pelo CoScripter sem que o usuário (no caso os membros da equipe do Google, que recebem a mensagem de Lina) tenha de fazer qualquer coisa exceto acionar este playback, clicando sobre o controle “run” da interface do CoScripter. Lina construiu este script através de passos de interação e ele representa de forma concreta o significado do novo elemento que Lina quer acrescentar à página de busca do Google. Observe que o script comunica uma informação a mais que não foi passada no email de Lina. Ele mostra que ela quer poder digitar o texto a ser traduzido diretamente no campo de texto da página de busca que atualmente serve para digitar o conteúdo a ser buscado.

Voltando à Figura 4.2 que ilustra a ferramenta primoTiWIM, podemos ver as operações para construção e alteração de significados, que são:

Selecionar: seleciona um objeto da interface.

Alterar: modifica o *label* de um objeto da interface.

Anotar: permite que seja associado um comentário a um objeto da interface.

Associar: relaciona um *script* do CoScripter a um objeto da interface.

Remover: apaga um objeto da interface.

Copiar: copia um objeto da interface.

Colar: insere na interface o objeto copiado. Seguindo um padrão ainda incômodo mas comum nas ferramentas que permitem customização e extensão de páginas na Web (e.g. Chickenfoot, Ubiquity e outros), o

posicionamento do objeto copiado tem de ser indicado “em referência a outros objetos da página”. Ou seja, eles podem ser colocados “antes”, “depois”, “dentro”, “fora” de outros objetos.

Para alterar um objeto, primeiramente o selecionamos passando o mouse sobre ele, então este ficará destacado com uma linha azul. Em seguida clicamos o *mouse* sobre o objeto para abrir o menu com os tipos de modificações que podem ser feitas. Essas funcionalidades foram implementadas utilizando o o *CustomizeYourWeb* (*CustomizeYouWeb*, 2010), um *plugin* para customizar páginas Web. Depois é só selecionar a opção desejada. Para finalizar e salvar a página alterada, o usuário deve utilizar a opção “salvar” do navegador. Feito isso, será gravado um arquivo *html* com as alterações. Esse arquivo compõe a descrição detalhada da mensagem. Dizemos que o arquivo “compõe” a descrição detalhada porque esta pode ser composta por mais de um arquivo, caso a modificação envolva mais de uma página do sistema.

É importante ressaltar que as tecnologias selecionadas para concretizar o modelo são voltadas para Web. Portanto, esta implementação do modelo restringe que a discussão aconteça em sistemas Web.

Para melhor entendimento do cenário de uso desta concretização do modelo, a seguir apresentamos um exemplo. O sistema discutido é o Sistema de Requerimento (SR) de pós-graduação de uma universidade brasileira. Ele é um sistema típico de *workflow*, com papéis bem definidos (aluno, professor, coordenador de pós-graduação, coordenador de centro, coordenador geral e secretária). Em um fluxo típico o aluno faz um requerimento a ser avaliado pelo orientador e diversos coordenadores da universidade. As secretárias dos departamentos acompanham o andamento do requerimento e utilizam a informação em outros contextos de seu trabalho.

Neste exemplo a discussão acontece entre coordenadores dos departamentos de Direito, Psicologia, Engenharia e a equipe de designers deste sistema.

4.2

Exemplo Ilustrativo

João é coordenador do curso de pós-graduação em Direito e frequentemente precisa acessar o Sistema de Requerimentos para analisar os pedidos de seus alunos. João procurou a equipe de desenvolvimento desse sistema para dizer que gostaria de vir a ser informado quando um requerimento é concluído e qual a decisão final. Como proposta, ele sugeriu que o sistema envie um email

avisando do término e do parecer final para todas as pessoas envolvidas no processo.

A equipe de design, antes de implementar tal sugestão, resolveu consultar outros coordenadores e ficou surpresa com a divergência de opiniões. Coordenadores de departamentos com muitos alunos não gostaram da ideia porque alegaram que, devido ao volume de requerimentos, os avisos iriam encher suas caixas de email, já coordenadores com poucos requerimentos não alegaram esse problema. Diante disso, o desenvolvedor Mário, resolveu criar um grupo para discutir a solução do problema levantado por João. Mário convidou duas secretárias, três coordenadores de pós-graduação e um coordenador do centro técnico-científico (que avalia todos os processos da área de exatas). Após a definição do grupo, Mário enviou a primeira mensagem:

Primeira mensagem: Mário discute alternativas de reprojeto com usuários

Cabeçalho do e-mail

Assunto: Envio de email no Sistema de Requerimento

Data: 24 de abril de 2010

De: Mario@universidade.com

Para: Joao@universidade.com, Paulo@universidade.com, Lurdes@universidade.com, Ana@universidade.com, Marcelo@universidade.com

Corpo do email

Tipo de fala: Sugiro

Estou falando como: designer do Sistema de Requerimentos

Falando para: coordenadores de pós-graduação, coordenadores de centro e secretárias.

Descrição Geral: Diante do problema apresentado por Mário, sugiro que os coordenadores tenham uma opção no sistema para dizer se quer ou não receber aviso de conclusão do requerimento.

Descrição Detalhada: Segue em anexo a página do sistema alterada, contendo a proposta de modificação.

A Figura 4.3 mostra a página gerada na primoTiWIM, que foi anexada ao email, contendo a expressão que Mário tem em mente. João, Paulo, Lurdes, Ana e Marcelo estão aptos a abrir o anexo e ver que a proposta de Mário foi inserir um radio button. O

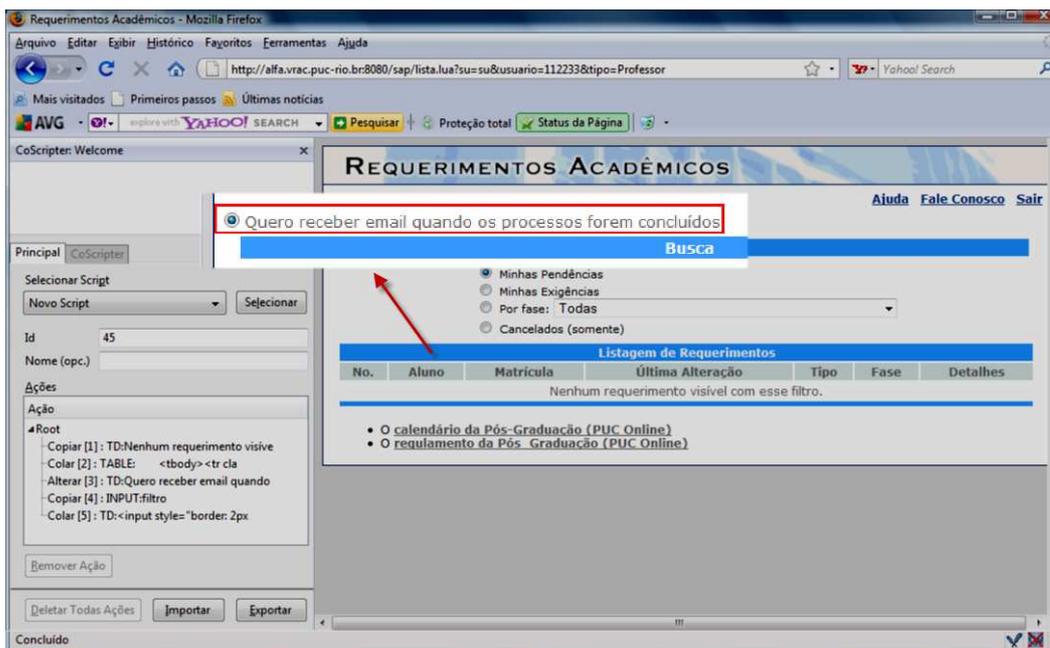


Figura 4.3: Página alterada do Sistema de Requerimentos

rótulo associado a este objeto diz: “Quero receber email quando os processos forem concluídos”, que ficou marcado com uma linha vermelha, indicando para os receptores da mensagem que ali tem um novo elemento.

Quando João lê a proposta de Mário, percebe que não era exatamente aquilo que ele tinha em mente. Receber email para cada requerimento irá encher sua caixa de emails. Então, ele decide comunicar ao grupo o que pensa sobre esta proposta: associar a opção de email para cada requerimento e não para todos. Segue sua mensagem:

Segunda mensagem: resposta de João (coordenador de pós-graduação) à proposta de Mário.

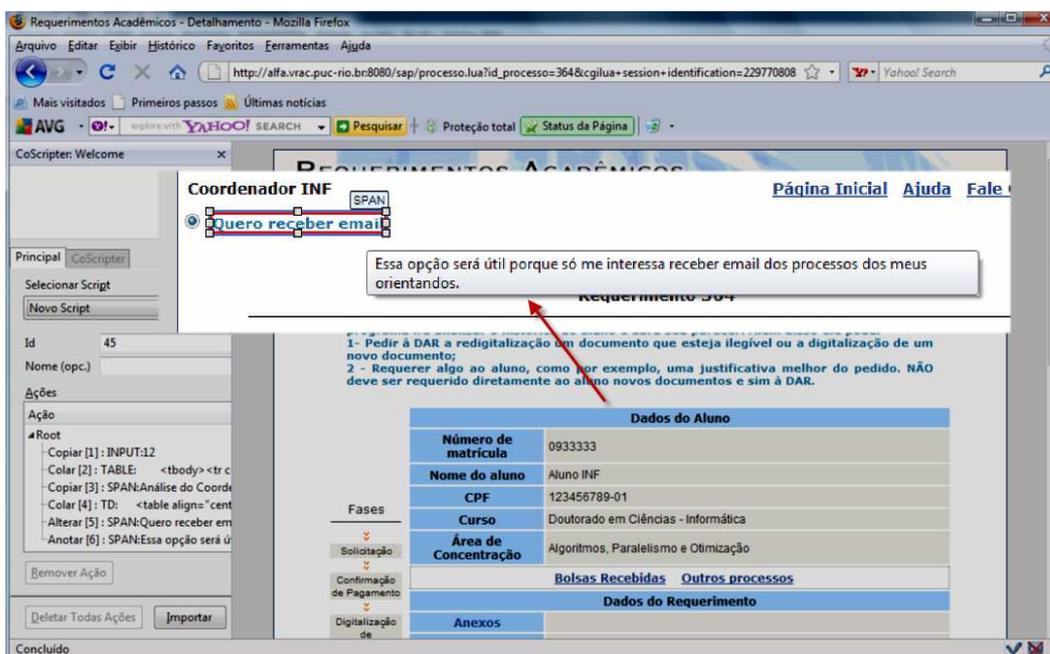
*Cabeçalho do e-mail***Assunto:** Envio de email no Sistema de Requerimento**Data:** 24 de abril de 2010**De:** Joao@universidade.com**Para:** Mario@universidade.com, Paulo@universidade.com,
Lurdes@universidade.com, Ana@universidade.com,
Marcelo@universidade.com*Corpo do email***Tipo de fala:** discordo**Estou falando como:** coordenador de pós-graduação**Falando para:** coordenadores de pós-graduação, coordenadores de centro, secretárias e designer.**Descrição Geral:** Ter uma opção de escolha para cada pessoa é uma boa sugestão, porém acho que a escolha deve ser dentro de cada requerimento.**Descrição Detalhada:** Descrição Detalhada: Segue em anexo a página do sistema alterada, contendo a minha nova proposta.

Figura 4.4: Resposta de João

Na Figura 4.4 vemos a proposta de João. Para implementar sua idéia João alterou a página do sistema onde é mostrado um requerimento específico, inserindo um radio button “Quero receber email”. Além disso, João registrou a ‘intenção’ de sua proposta em

uma anotação sobre esse elemento: “Essa opção será útil porque só me interessa receber email dos processos dos meus orientandos”.

Terceira mensagem: resposta de Marcelo (coordenador de centro)

Quando Marcelo, que está em um congresso em Londres, acessou seu email e viu a proposta de João, ele acrescentou uma informação importante: se a opção já vem marcada, ele provavelmente iria esquecer de desmarcar, e a caixa de email iria ficar cheia. Se não esquecesse, de todo modo estaria sempre tendo que fazer uma interação extra, pois na maioria dos casos ele não quer receber email. Segue sua mensagem:

Cabeçalho do e-mail

Assunto: Envio de email no Sistema de Requerimento

Data: 24 de abril de 2010

De: Marcelo@universidade.com

Para: Mario@universidade.com, Paulo@universidade.com, Lurdes@universidade.com, Ana@universidade.com, Joao@universidade.com

Corpo do email

Tipo de fala: acrescento

Estou falando como: coordenador de centro

Falando para: coordenadores de pós-graduação e designer.

Descrição Geral: Acho que a opção não pode vir marcada, é melhor que a opção padrão seja “desmarcado”.

Descrição Detalhada: Não preciso detalhar minha proposta.

Embora Marcelo não tenha direcionado sua proposta às secretárias, pois ele estava pensando em seu papel de coordenador de centro e como seu trabalho é afetado pela proposta do coordenador de programa, Lurdes, uma das secretárias, viu sua mensagem porque ela está no grupo de discussão. O conteúdo a fez lembrar que talvez ela pudesse ser beneficiada se existisse a opção de filtrar os tipos de requerimento que quer ser notificada. Então, ela enviou a seguinte mensagem ao grupo:

Quarta mensagem: resposta de Lurdes (secretária)

Cabeçalho do e-mail

Assunto: Envio de email no Sistema de Requerimento

Data: 25 de abril de 2010

De: Lurdes@universidade.com

Para: Mario@universidade.com, Paulo@universidade.com,
Marcelo@universidade.com, Ana@universidade.com, Joao@universidade.com

Corpo do email

Tipo de fala: comento

Estou falando como: secretária

Falando para: coordenadores de pós-graduação, coordenador de centro e designer.

Descrição Geral: Como nós, secretárias, só atuamos nos processos de aproveitamento de créditos, acho interessante só receber email desses tipos de processo.

Descrição Detalhada: Segue em anexo a página do sistema alterada, contendo a minha nova proposta.

Na figura 4.5 podemos ver a proposta de Lurdes que modificou uma página diferente das páginas alteradas anteriormente. Ela propôs adicionar o botão “Quero receber email” associado ao tipo de requerimento e utilizou a anotação (veja 1) para explicar que a opção é para enviar email às secretárias sobre os tipos de processos selecionados. A primoTiWIM registrou todo o processo de edição feito por Lurdes (veja 2).

Mário percebeu que existia uma diversidade de interesses quanto ao envio de email, e entendeu que seria necessário construir uma página de configuração de avisos para atender todos os casos. Ele precisa de tempo para pensar a respeito mas ficou satisfeito porque em poucos dias ele conseguiu capturar a grande diversidade de percepção sobre o sistema por diferentes usuários. Então ele enviou uma última mensagem, informando que vai criar uma página de configuração e que em breve enviará mais detalhes sobre tal página. Diante disso, a discussão foi suspensa.

Refletindo sobre o exemplo apresentado, chamamos atenção para os seguintes pontos:

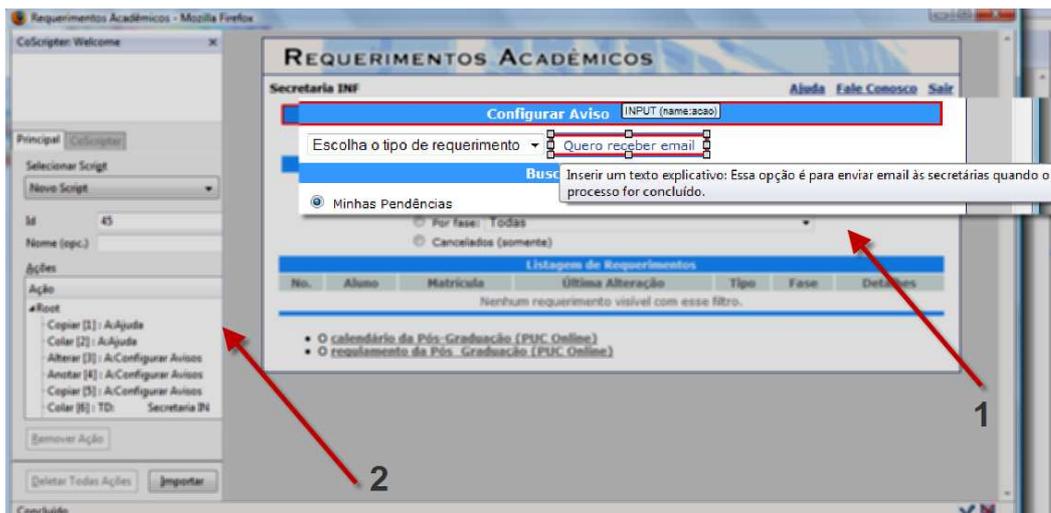


Figura 4.5: Proposta de Lurdes

O fato do *radio button* vir marcado significou algo importante para o coordenador do centro. Será que isto seria observado em uma simples troca de email (usando apenas narrativas textuais)?

Se Mario não tivesse este modelo, como ele comunicaria e discutiria o problema apresentado pelo coordenador de Direito? Provavelmente ele passaria um email e tentaria narrar o que escreveu na descrição geral, mas será que ele escreveria a ideia com a mesma riqueza de detalhes? Qual o esforço para fazer isso?

Uma discussão como esta, ajudou o designer a negociar soluções mais facilmente e rapidamente, porque um dos benefícios deste modelo é levar o usuário a refletir sobre as alternativas de projeto e envolvê-lo nas decisões.

Este exemplo apontou para um ponto importante abordado nesta pesquisa: por que estamos focando em sistemas de grupo? A proposta que era boa para os coordenadores não era adequada para as secretárias, isto porque as secretárias só atuam em alguns tipos de requerimento, portanto, para o papel de secretária seria necessário criar um filtro pelo tipo de requerimento.

Além disso, podemos observar que a solução dada para uma pessoa nem sempre é boa para a outra. Vimos isso acontecer tanto entre os coordenadores e secretárias como também entre coordenadores de departamentos distintos. Este fato reforça a necessidade de discutir sobre modificações em sistemas onde existem grupos de pessoas exercendo diferentes funções e papéis no sistema.

4.3

“primoTiWIM”: um Signo da Engenharia Semiótica

Podemos dizer que a primoTiWIM é um signo da Engenharia Semiótica, ou seja: ela representa inúmeras questões motivadoras e diferenciais desta

teoria. Em si, esta já constitui uma contribuição desta pesquisa para o avanço da Engenharia Semiótica, que é subsidiária em relação às contribuições centrais pretendidas, mas é importante. A ferramenta “significa” a teoria sob três aspectos principais:

1. A Engenharia Semiótica é uma teoria cujo foco é na “comunicação” de ideias e intenções. Portanto, a ideia (por si só) de trocar mensagens sobre modificações de sistemas e de comunicar através de signos estáticos, dinâmicos e metalinguísticos sobre evolução de sistemas, é uma forma de fazer “Engenharia Semiótica”.
2. Para esta teoria, IHC é um caso particular de comunicação humana mediada por sistemas computacionais. O conteúdo da mensagem do projetista para os usuários é veiculado pela interface dos sistemas, e pode ser parafraseado no *template* de metacomunicação. Podemos ver na Figura 4.6 que parte da mensagem sobre modificação de sistema está relacionada ao *template* de metacomunicação. O campo descrição geral na mensagem de um proponente carrega a intenção geral de sua comunicação, que é qualificada pelo tipo de fala. No campo tipo de fala, proponentes podem explicitamente marcar como eles querem que suas mensagens sejam recebidas pelos outros participantes da discussão (por exemplo, eles podem estar sugerindo, discordando, comentando, etc.). Detalhes adicionais relacionados à intenção do proponente podem ser incluídos em anotações para elementos selecionados ou modificados da interface do sistema, opcionalmente anexadas à mensagem para discussão.

Além disso, representações detalhadas das propostas de (re)design incluem signos estáticos, dinâmicos e metalinguísticos, embora usuários possam ignorar completamente que estas classificações existem e o que querem dizer. Copiar ou modificar elementos de interface em uma página existente, por exemplo, tipicamente gera novos signos estáticos. Associar *scripts* à ativação de novos elementos de interface é uma forma de incluir novos sinais dinâmicos. Finalmente, anotações sobre elementos de interface novos ou modificados carregam uma série de signos metalinguísticos que podem ser explorados pelos designers do sistema quando estes decidem sobre como deveria ser a aparência de uma nova interface e como ela deveria se comportar quando o sistema é modificado. Com isso, podemos dizer que a *primoTiWIM* é um signo da Engenharia Semiótica, pois através de seu uso estamos fazendo Engenharia Semiótica.

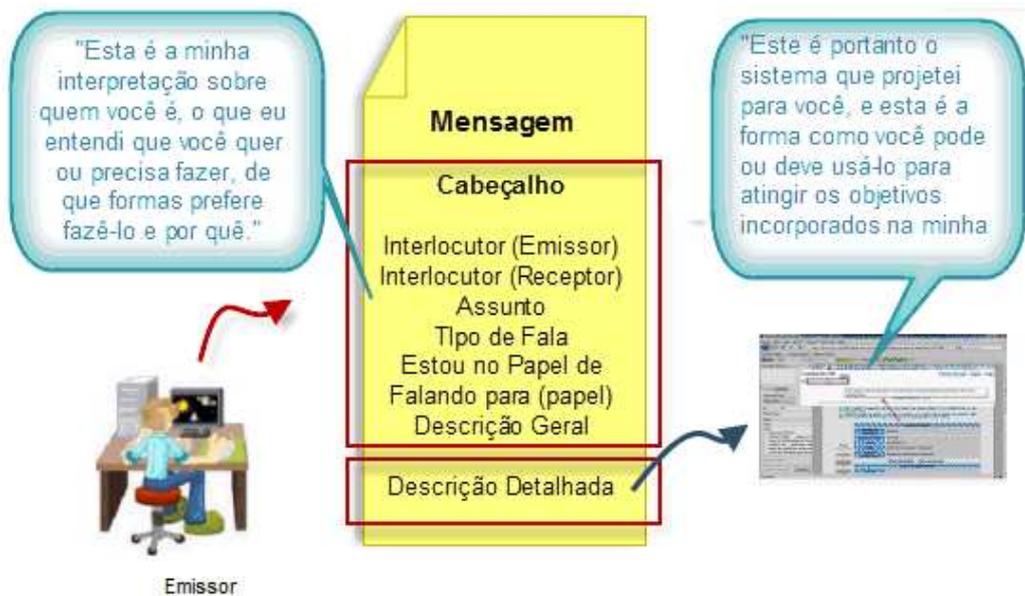


Figura 4.6: Metamensagem da Proposta

3. O ambiente de discussão ao todo (ferramenta de discussão e primoTiWIM) ajuda designers e usuários a focar na melhoria da interação sem perder a qualidade. Como o exemplo ilustrativo mostrou, discussões semioticamente enriquecidas como esta podem ter um efeito educacional sobre os designers, desenvolvedores e usuários, porque a estrutura da negociação e o esforço de representar alternativas de projeto concretamente, fazem os interlocutores pensarem sobre aspectos de projeto que poderiam passar despercebido ou subestimado em formas tradicionais de comunicação. Este efeito reflexivo do uso da primoTiWIM é uma característica da Engenharia Semiótica.

4.4

Estudos Empíricos

Iniciamos os estudos empíricos desta pesquisa em 2008 quando analisamos como dois grupos distintos de usuários finais potencialmente envolvidos em situação de EUD reagiram ao uso da linguagem de macros do CoScripter (Little et al., 2007) (Leshed et al., 2008), para fins de representar extensões ou adaptações de sistemas existentes na Web (Sampaio & de Souza, 2008). A linguagem do CoScripter tem baixo custo de aprendizagem, pois ela possui um conjunto pequeno de ações (e.g. go to, click, enter, etc.) e sintaxe bem definida e simples. Portanto, decidimos observar como os usuários reagem ao uso dela para fins de representar extensões ou adaptações de sistemas existentes. Para isso, utilizamos um método de pesquisa qualitativa (Denzin & Lincoln, 2006), objetivando explorar as

percepções dos usuários sobre esta linguagem na tentativa de realizar um conjunto pré-estabelecido de atividades. Os resultados encontrados serviram para identificar dois pontos importantes: (i) a viabilidade de usar a linguagem do CoScripter como uma linguagem de especificação; e (ii) a necessidade de enriquecer esta linguagem com tipos de construção gramatical específicos para atender a requisitos de especificação em processos de EUD.

Este estudo foi importante para esta pesquisa primeiro porque foi o início de nossas investigações com usuários finais, e logo nas primeiras entrevistas identificamos várias características típicas deste perfil. Segundo, porque os resultados mostraram a viabilidade de utilizarmos a linguagem do CoScripter para expressão do signos dinâmicos da proposta de evolução do sistema. Apesar da viabilidade de utilização do CoScripter, vimos a necessidade de ela ser incrementada, pois não representa ações relacionadas ao Golfo de Avaliação (Norman, D.A., 1986). Este fato é uma indicação de que propostas baseadas na Engenharia Semiótica, as quais favorecem a representação explícita de falas do sistema e do usuário, podem ecoar a intuição das pessoas, visto que elas sentiram falta, ao especificar o que tinham em mente, de representar os dois lados da conversa.

Após a elaboração desta investigação com o CoScripter, partimos para a elaboração do modelo aqui proposto. Com finalidade de avaliá-lo, realizamos dois estudos empíricos com potenciais usuários deste modelo. Falamos sobre “potenciais” usuários porque a resposta para este tipo de tecnologia depende da adoção de uma nova cultura entre os usuários - uma cultura de participação, justamente, que ainda é tímida no cenário de desenvolvimento de sistemas hoje. O objetivo dos estudos foi averiguar: (i) que tipo de conteúdo e intenção pode ser expresso; (ii) qual a qualidade da metacomunicação do modelo; (iii) quais questões de usabilidade devem ser tratadas para eliminar barreiras desnecessárias de uso para esta e para futuras implementações da *primoTiWIM*. No decorrer do texto, estes estudos são chamados de Estudo Empírico I e Estudo Empírico II. O Estudo Empírico I investiga as questões (i) e (iii) e o Estudo Empírico II investiga as três questões.

Usamos uma metodologia de pesquisa qualitativa em ambos estudos, pois neste estágio da pesquisa, como se trata de um proposta que muda uma cultura de participação dos usuários, é mais importante explorar os significados associados à experiência de uso do que fazer previsões precipitadas sobre o sucesso ou insucesso de uma tecnologia ainda não está totalmente delineada. Estamos interessados em explorar os significados e interpretações associados à situação-alvo de nossa pesquisa, e por isto a metodologia qualitativa é a adequada. O interesse na exploração de significados e interpretações é

totalmente alinhado com os objetivos da engenharia semiótica das mensagens de metacomunicação dos sistemas interativos. Estamos trabalhando com grupos pequenos, observando suas interações com o protótipo de nossa tecnologia e entrevistando os participantes para captar sua percepção, sentimentos e atitudes em relação à tecnologia proposta.

O sistema discutido nos estudos é o Sistema de Requerimento (SR) já apresentado no cenário ilustrativo deste capítulo. Escolhemos este sistema porque ele possui todas as características de sistemas de grupo presumidas nesta pesquisa e também porque conhecemos bem algumas das dependências reais de modificação deste sistema. Isto permitiu que os cenários dos estudos fossem bastante realistas, aumentando as chances de encontrarmos significados e interpretações igualmente realistas.

4.5

Estudo Empírico I

Nosso primeiro experimento foi realizado com três alunos de pós-graduação, que são chamados de P1, P2 e P3. P1 e P2 são alunos dos cursos Informática e P3 é aluno do curso de Letras. Todos são usuários do SR e já tinham utilizado este sistema pelo menos uma vez quando realizaram o experimento associado ao estudo.

Os objetivos deste estudo foram: avaliar qual a qualidade da metacomunicação da primoTiWIM e investigar questões de usabilidade da ferramenta. Mostramos um conjunto de problemas reais de IHC no SR para os participantes e pedimos para eles gerarem sugestões e explicações para os usuários do SR. Tais sugestões poderiam ser descritas utilizando a ferramenta primoTiWIM ou não. É importante notar que neste estudo não promovemos discussão, o objetivo primordial foi apenas avaliar como os usuários finais utilizam a UIL para expressar ideias. Ou seja, o foco estava nas representações que a implementação da Linguagem de Modificação (LM) de nosso modelo permite construir.

Apresentamos aos participantes a seguinte problemática:

Os alunos de pós-graduação acessam o Sistema de Requerimentos para solicitar pedidos como afastamento, trancamento, atestados, prorrogações, etc. Ao acessar este sistema o aluno encontra dúvidas tais como: Qual tipo de processo devo escolher? Como faço para imprimir o boleto? Quanto tempo vai levar para o processo ser concluído? Quem avalia este processo? Posso fazer esse requerimento até quando? Que informação devo inserir neste campo? Quantas etapas têm esse processo?

“Você é gestor deste sistema e vai analisar os problemas mencionados objetivando dar soluções para melhorar o entendimento e o uso do mesmo. Para tanto, pedimos para você elaborar explicações que serão enviadas aos designers como sugestões. Tais explicações podem ser desde textos explicativos na página, dicas em botões, até mensagens de erro. Lembre-se que esses textos serão lidos por alunos, coordenadores e secretárias dos diversos departamentos. Portanto, é necessário que você use termos que todos entendam. Além disso, você deve considerar as seguintes restrições:

- 1) Priorizar as informações mais relevantes.*
- 2) Você tem um limite de espaço e não deve produzir textos muito longos.*
- 3) Considerar a fase e o estado em que o processo se encontra. Você dispõe de uma ferramenta para alterar a página e facilitar a comunicação da sua proposta de melhoria do sistema para os designers.”*

Resumindo, mostramos aos participantes problemas de interação do SR. Em seguida, apresentamos a primoTiWIM como uma ferramenta de expressão de modificação de sistemas. Por último, pedimos para eles elaborarem sugestões que endereçassem os problemas mencionados, considerando as restrições de 1 a 3. Frisamos que as sugestões poderiam ser descritas utilizando ou não a primoTiWIM. Ao final de cada experimento entrevistamos os participantes para colher opiniões sobre a atividade.

Todos usaram a primoTiWIM, uns mais, outros menos. A seguir apresentamos os principais resultados deste estudo.

4.5.1 Resultados do Estudo Empírico I

A ferramenta apresentou problemas de Usabilidade e Comunicabilidade. Como exemplo de falha de usabilidade tivemos casos como o de P1 que tentou inserir um elemento depois de outro e não conseguiu colar no local desejado, e disse: “Seria interessante poder arrastar objetos”. Outro problema de usabilidade e comunicabilidade da linguagem da ferramenta foi ausência de *feedback* na alteração de objetos na página. Por exemplo, se um botão ganhava uma anotação, isso não era visto facilmente, pois era preciso passar o mouse sobre ele para aparecer em forma de *tooltip*. Esse problema foi corrigido no Estudo Empírico II, de maneira que todo objeto alterado ficasse destacado com uma linha vermelha. Já o problema de “colar” o objeto em um local

específico da página, não pôde ser atacado devido à restrições da arquitetura da ferramenta.

Deixamos os participantes à vontade para usar ou não a primoTiWIM e observamos que eles utilizaram diferentes estratégias. O primeiro participante fez uso intenso da ferramenta com intuito de explorá-la. Ele inseriu anotações, alterou textos da página e nome de botões. Já o segundo participante usou pouco a ferramenta, mas não por problemas de interação com ela. Ele simplesmente explicou verbalmente a maioria das melhorias sugeridas para a página. Esse fato nos chamou atenção para a necessidade de motivar o usuário a usar a ferramenta. O terceiro participante, aluno do curso de Letras, sentiu-se à vontade e inseriu várias anotações. Ele escreveu textos longos e bem explicados. Inclusive, usou a anotação para dizer:

“Poderia haver um campo extra no qual, caso seja necessário, o aluno inclui uma redação do texto que deveria constar no atestado (quando necessárias informações adicionais e específicas).” (Anotação de P3)

P3 poderia ter criado um novo elemento, utilizando a primoTiWIM, para descrever esta ideia, mas preferiu expressar-se de outra maneira. Nos chamou atenção o fato de o participante não ter criado um novo objeto. Ele nem tentou. Para entendermos o motivo, na entrevista final, perguntamos a P3 como ele imaginava que seria esse novo signo que ele propôs criar. P3 respondeu dizendo que não sabia ao certo. Ele não definiu em que local da página iria inserir o novo elemento, qual seria o espaço adequado para ser digitado o texto do atestado ou até mesmo se poderia ficar em branco ou se o preenchimento seria obrigatório. Em seguida, ele passou a refletir sobre novas possibilidades. Ficou em dúvida se ao invés de criar novo elemento ele substituiria o campo ‘observação’, já presente na interface, para o aluno inserir a redação, dando assim um novo significado a este campo. A outra possibilidade seria não criar novo elemento na interface, mantendo o nome do campo “observação”, visto que este serviria para inserir a redação do atestado. Portanto, observamos que P3 não sabia ao certo detalhar a ideia. Pode ser que não tenha usado este recurso por não ter em mente suficientes informações e conhecimentos para decidir o que fazer. Por exemplo, o novo elemento ficaria onde? Qual seu tamanho? Que cor?

Por outro lado, este fato é uma evidência da flexibilidade de expressão de um ambiente semioticamente rico (com vários tipos de linguagens). Apesar de P3 não sentir-se à vontade para usar o recurso de criação explícita de signos estáticos da primoTiWIM (copiar e colar um novo elemento), ele conseguiu passar a ideia de outra maneira, mesmo que incompleta e ambígua. Esta é uma

questão interessante para nossa pesquisa, porque está diretamente relacionada ao limite adequado de expressividade que a linguagem para descrever as modificações em sistemas deve ter.

De maneira geral, os participantes usaram a *primoTiWIM* sem grandes problemas. Porém, observamos que eles tiveram dificuldade para elaborar soluções de design de sistemas. Por exemplo, eles não tinham em mente boas ideias para inserir textos explicativos em objetos da interface. Dado o problema apresentado, eles não souberam se era preciso dar mensagens de erro, ou alterar o nome de objetos ou inserir *tooltips* explicativos. Em outras palavras, observamos que é difícil para pessoas que não são treinadas desenvolver atividades de design e imaginar que modificações concretas de interface poderiam solucionar problemas percebidos. Comentários mais gerais sobre o problema são mais fáceis de fazer.

Logo, se usuários finais têm dificuldade de imaginar resultados futuros, então como podemos testar o poder de expressão de nosso modelo? Diante desta questão, optamos por realizar um experimento empírico com avaliadores de IHC, visto que eles são capacitados para elaborar atividade de *design*, e conseqüentemente, elaborar propostas interessantes para solucionar problemas de interação.

4.6

Estudo Empírico II

O objetivo geral deste experimento foi saber se o modelo aqui proposto é suficientemente bom para expressar ideias mais elaboradas (ou menos ingênuas do que usuários finais típicos - sem uma cultura de participação em evolução de sistemas - conseguem gerar) sobre modificações de sistemas. Mais precisamente quisemos avaliar o poder de expressão do modelo tal como instanciado na ferramenta *primoTiWIM*.

A primeira etapa do fluxo de uso do modelo, “Criar um grupo de discussão” foi realizada por nós, previamente, e não pelos participantes. Selecionamos a parte do Sistema de Requerimentos (SR) a ser discutida, definimos os papéis (aluno, coordenador de pós-graduação, coordenador de centro, coordenador central de pós-graduação e secretárias de departamento) e os problemas em questão. Escolhemos problemas reais, polêmicos e que envolvem os papéis de coordenador de pós-graduação e aluno. Decidimos que a ferramenta de apoio à discussão utilizada neste experimento seria o aplicativo de email dos participantes. Impusemos ao email uma estrutura na forma de *template*, onde a estrutura da discussão era replicada e imposta. Os participantes do estudo, avaliadores de IHC, estavam na realidade fazendo uma

espécie de avaliação por inspeção, “advogando” pelos usuários dos papéis que lhes foram atribuídos, o que é uma extensão de uso bastante legítima para a ferramenta primoTiWIM.

Este estudo foi realizado em dupla (cada experimento continha uma dupla de participantes), onde um participante (emissor), no papel de avaliador de IHC, elabora uma proposta para solucionar um dado problema e a envia ao outro participante (receptor). Esta forma, nos permitiu observar a elaboração da proposta e captar a percepção do receptor sobre a proposta do colega. Ao todo, foram 4 duplas compostas por pessoas com conhecimentos em IHC, embora algumas sejam mais experientes do que outras. As duplas são: (i) Pedro e Paulo, (ii) Vitor e Leo, (iii) Rute e Raquel e (iv) João e Maria. Estes nomes são fictícios e foram criados com finalidade de preservar o anonimato dos participantes.

Para cada dupla foi realizado um experimento que durou em torno de duas horas, seguindo em três etapas: explicação do experimento aos participantes, elaboração das propostas pelos participantes e discussão final. O Roteiro completo e detalhado do experimento pode ser visto no anexo deste documento.

4.6.1

Etapa 1: Explicação dos Experimentos aos Participantes

Nesta etapa falamos sobre o objetivo da pesquisa e sobre o SR e seus papéis. Em seguida, apresentamos nosso modelo através de um exemplo de uso do mesmo, incluindo a demonstração da primoTiWIM. Por último, mostramos dois problemas reais de interação do SR e pedimos que eles endereçassem possíveis soluções de reprojeto e enviassem ao parceiro para a ser avaliada e comentada por ele. Fica claro que no experimento II estávamos explorando o uso da ferramenta em um contexto de comunicação semelhante ao visado originalmente para esta tecnologia. As etapas do experimento claramente envolviam EXPRESSÃO e INTERPRETAÇÃO de representações nas linguagens de comunicação e modificação do modelo.

4.6.2

Etapa 2: Elaboração de propostas

Neste momento os participantes foram separados fisicamente e orientados a utilizar o modelo para elaborar a proposta de reprojeto do Sistema de Requerimentos para os problemas apresentados na etapa anterior. Cada participante foi acompanhado por um pesquisador, de maneira que pudéssemos observar o uso da ferramenta durante a elaboração de cada proposta. Além

disso, gravamos vídeos das interações dos participantes para análise posterior.

Alguns emissores iniciaram a elaboração da mensagem preenchendo os dados do cabeçalho da mensagem, na ferramenta de email. Outros começaram usando a primoTiWIM para descrever a ideia de reprojeto. Posteriormente, veremos que esse fato influenciou no conteúdo da “descrição geral” e “descrição detalhada” da mensagem. Discutimos a seguir aspectos importantes observados na estrutura de comunicação das propostas.

Comunicando o Assunto

O “assunto” foi utilizado na maioria dos casos para informar o teor da proposta de modificação. Por exemplo, nos chamou a atenção que alguns emissores refletiram bastante sobre como comunicar o assunto de maneira mais adequada. Vimos nisto uma evidência de que o “assunto” da mensagem tem um significado muito importante para a comunicação da proposta. Por outro lado, o fato de discutir propostas de reprojeto e de ter de comunicá-las através deste meio, enfrentando este momento de reflexão, põe em prática um dos preceitos mais fundamentais da Engenharia Semiótica, que é a reflexão sobre os processos de metacomunicação em que usuários, desenvolvedores e sistema estão reunidos quando se realiza a interação humano-computador.

Descrição Geral

João, Pedro e Vitor tiveram dificuldade em saber o que escrever na descrição geral. Pedro e Vitor ficaram na dúvida quanto ao nível de detalhe. João ficou na dúvida se relataria o problema ou a solução, ou ambos. Algumas descrições foram longas e outras sucintas (ver anexo).

Observamos que a ordem com que o emissor descreve os dados da mensagem pode influenciar no que vai ser dito na descrição na geral. Se ele iniciar a proposta escrevendo primeiramente a descrição geral, este campo tende a conter a intenção geral do que está sendo proposto. Mas se a descrição geral é escrita após a descrição na primoTiWIM, a descrição geral pode funcionar como um comentário do que foi proposto na primoTiWIM. Ou seja, na segunda situação, a narrativa é um pouco mais detalhada que na primeira. Por exemplo, Raquel escreveu a descrição geral detalhando o que ela havia feito na primoTiWIM:

“[...] Sugiro deslocar o texto mais pra cima na página, a fim de que chame mais a atenção, já que não sei se é possível mudar a tela de confirmação para uma tela de alerta”. (Descrição geral da mensagem de Raquel)

Já Paulo, ao contrário de Raquel, escreveu primeiramente a descrição geral antes de usar a primoTiWIM. Ele expressou seu entendimento sobre o problema e depois falou como o endereçou:

“A página para confirmar não deixa claro o seu objetivo. Inclui no início da página a mensagem FAVOR CONFIRMAR O ENCAMINHAMENTO e o botão Confirmar”. (Descrição geral da mensagem de Paulo)

A descrição geral é um espaço onde o interlocutor pode registrar a intenção de design. Podemos observar na fala da Raquel, citada anteriormente, que ela explicitou sua intenção quando disse: “a fim de que chame mais atenção”.

A descrição geral pode também nos permitir capturar nomes de objetos rotulados pelo emissor. Por exemplo, Leo alterou o significado do botão de “Confirmar” dando um novo rótulo, na ferramenta, para essa funcionalidade:

*“[...] Sugiro a mudança do nome do botão **Confirmar** para **Pré-Visualizar** na tela de edição [...]”. (Descrição geral da mensagem de Leo)*

Tipo de Fala

Todos os participantes colocaram o tipo de fala “proponho”. Vitor, Leo, Rute, Raquel e João falaram que o “tipo de fala” seria importante se a conversa tivesse uma contraproposta. Seria importante se o experimento permitisse contrapropostas. Porém, observamos que Maria, Pedro e João refletiram sobre o objetivo de sua proposta ao ter que preencher essa informação. Raquel gostou bastante dos tipos de falas disponíveis e lamentou não ter tido oportunidade de usá-las: *“Fiquei doida para usar isso aqui”*.

Estou no Papel de/Falando Para (outro papel)

Todos disseram estar no papel de “Designer” falando para “Designer”, exceto um participante que falou estar no papel de “Consultor/Avaliador” para ambos. Este papel não existe e não seria possível de ser assumido, pois no modelo, os únicos papéis disponíveis são os da aplicação (se estiverem marcados no protocolo tecnológico) e o de “designer”. Acreditamos que isto foi consequência de como esta parte do modelo foi concretizada, pois a metacomunicação para os participantes teria sido totalmente diferente se a primoTiWIM tivesse implementado que o valor deste “campo” devia ser escolhido entre uma lista fechada de valores possíveis.

Adicionalmente, ficou confuso se o papel era o suposto usuário por quem estavam advogando (cujos interesses estavam defendendo ao tentar melhorar o design da interface), como por exemplo, “estou falando como aluno”, ou se era o papel que ele próprio estava exercendo na discussão sobre o uso e os usuários (que apareceriam em 3ª pessoa no discurso), como por exemplo, “Eu sou designer desse sistema”. É bem verdade que pode acontecer de uma pessoa ser designer da aplicação e querer falar como designer. Talvez o cenário tenha contribuído para isso. Mas eles poderiam ter falado com voz de “coordenador” por exemplo. Isto é um exercício interessante porque pode haver mudança no discurso de quem fala em nome de outro.

Uma ruptura na comunicação entre João e Maria foi identificada em decorrência deste problema. Uma das soluções de João envolveu mais de um papel. Ele falou em nome de “aluno” e de “coordenador”, isto porque a solução que ele deu para o problema alterou a página do aluno e do coordenador (ver anexo). Porém, isso não foi dito em nenhum lugar. Quando Maria recebeu a mensagem ela não reconheceu a página do aluno (pois ela esperava receber somente a página do coordenador) e ficou sem entender o motivo dele ter enviado aquela página. Consequentemente, Maria não conseguiu entender a mensagem de João.

Descrição Detalhada

A descrição detalhada é um espaço para associar a parte da mensagem elaborada na *primoTiWIM* com o restante da mensagem escrita na ferramenta de discussão. Então, a “descrição detalhada” serviu para informar que foi anexado ao email um arquivo HTML contendo a proposta narrada detalhadamente na *primoTiWIM*. Leo, Rute, João e Maria elaboraram alternativas de solução para o mesmo problema. Nesses casos a descrição detalhada fez referência à vários HTMLs, e foi um espaço para comunicar que houve alternativas de reprojeto.

Uso da Ferramenta

Apresentamos a seguir questões de usabilidade observadas em relação à *primoTiWIM*.

- Nossa estratégia de posicionar objetos copiados (ex. colar antes ou depois do botão OK) adotada por outras ferramentas como Chickenfoot (Bolin et al., 2005), não foi bem recebida pelos usuários. Assim como no Estudo Empírico I, alguns participantes tentaram redimensionar e/ou arrastar objetos utilizando um estilo de manipulação direta ao invés das diretrizes linguísticas do tipo ‘colar antes’, ‘colar depois’, etc.

- A anotação feita em um objeto é vista através de uma *tooltip* no próprio objeto. Quando a anotação é grande, alguns participantes reclamaram que o tempo que a *tooltip* fica aberta é pequeno, dificultando a leitura. Ainda em relação à anotação, Maria e Pedro sugeriram que a anotação não fosse em forma de *tooltip*, pois causa confusão quando o objeto já tem *tooltip*. Maria sugeriu trocar para um signo que remete um o significado de um “postit”.
- Alguns participantes precisaram criar tipos de elementos que não podiam ser copiados, visto que não existiam na página. Por exemplo, Raquel gostaria de criar uma janela de alerta, mas a *primoTiWIM* não implementou essa funcionalidade, que no entanto era prevista no modelo (ver Capítulo 3). Diante disso, Raquel inseriu uma anotação explicando sua ideia:

“... o ideal seria o uso de uma tela de alerta. Como não consegui fazer isso, sugiro pelo menos deslocar o texto, para assim chamar mais atenção”. (Fala de Raquel)

Leo também relatou sua dificuldade:

“Eu pensei num menu diferente e não consegui... [se referindo que queria criar um elemento que não existia na página] inseri a anotação” (Fala de Leo)

- Alguns participantes sugeriram integrar a parte do modelo que foi executada via email com a ferramenta *primoTiWIM*. Isso aconteceu porque eles acharam trabalhoso e pouco prático a operação de salvar todos os arquivos HTML, compactar e anexar no e-mail. Por exemplo, Leo disse:

“Achei confuso ter que salvar o HTML e Zipar. Se tivesse alguma coisa que já fizesse direto na própria ferramenta... ela poderia já enviar o e-mail”. (Fala de Leo)

- Alguns participantes tentaram usar o modelo como uma ferramenta de design, ou seja, tentaram usar como ferramenta de prototipação, apesar de termos frisado que o propósito era comunicar a ideia. (João, Maria, Vitor, Leo, Pedro e Paulo). Por esse motivo, Vitor e Leo concluíram que esse modelo é interessante para modificações simples. Eles disseram isso se referindo às modificações mais complexas, as quais exigem reflexão antes da elaboração.

- A função da ferramenta que permite associar um *script* do CoScripter a um elemento da interface não foi usada, o que pode estar associado à dificuldade de fazer o *script* no CoScripter. Porém, olhando para as sugestões feitas pelos participantes, acreditamos que o cenário do experimento pode não ter motivado suficientemente o uso deste recurso. Esta é portanto uma das limitações deste estudo empírico.

4.6.3

Etapa 3: Entrevista

Iniciamos esta etapa pedindo separadamente para cada participante interpretar a proposta do outro (e.g. Vitor explicou aos participantes o que entendeu das propostas de Leo e vice-versa). Feito isto, realizamos uma discussão em grupo (a dupla e os dois pesquisadores). Apesar dos dois problemas apresentados para os participantes serem relativamente simples, os avaliadores propuseram soluções bem diferentes um do outro. Por esse motivo, a interpretação das propostas e a discussão nessa etapa ficou ainda mais interessante. Discutimos a seguir, a interpretação das mensagens e outros assuntos abordados nesta discussão em grupo.

Interpretação das Mensagens

No momento da interpretação, Rute não leu o email de Raquel, ou seja, não viu a narrativa inicial do modelo (tipo de fala, descrição geral, etc) e partiu imediatamente para ver o HTML (descrição detalhada). Quando lembrou de ler o email, ela concluiu que se tivesse visto antes teria entendido mais rápido a proposta de Raquel. Já Raquel, ao interpretar as mensagens de Rute, deixou claro que o conteúdo da mensagem estava todo na descrição detalhada:

“O email serve mais para me antecipar o que você está falando... não acho que serve para esclarecer... o esclarecimento ficou todo no HTML”. (Fala de Raquel)

Na solução de um dos problemas, João considerou interfaces diferentes para alunos e coordenadores. Portanto sua proposta envolveu a alteração de duas páginas. Em seguida, ele as enviou a Maria mas não avisou isto em lugar nenhum. Isto causou confusão no entendimento, pois Maria viu a página do aluno e não a reconheceu:

“Eu nunca tinha visto essa tela na minha vida, porque a tela que eu fiz o teste era outra, então eu fiquei completamente perdida”. (Fala de Maria)

Além disso, houve uma desconexão dos nomes dos arquivos HTML e o que eles representam. João nomeou os arquivos (um arquivo para cada página): Proposta1, Proposta do Problema1, Problema 2 - parte1, Problema 2 - parte2 e Problema 2 - parte3. Isso ficou bastante confuso. Em um dos problemas João fez duas alternativas, e no outro ele precisou alterar três páginas no sistema para endereçar o problema.

Outro problema de interpretação é causado quando um objeto sofre mais de uma alteração, neste caso a primoTiWIM não diz que houveram várias alterações, ela só diz: houve alteração. Por exemplo, Maria incluiu um novo objeto e em seguida inseriu uma anotação sobre o mesmo. João, ao ler a mensagem de Maria, não percebeu que o objeto era novo na página: “Achei que era uma anotação... aquilo foi uma inclusão?” Eles sugeriram que a ferramenta deixasse uma janela aberta com a página original. Reforçando essa ideia, Maria complementou: “Mesmo tendo sido eu que fiz eu não consigo lembrar...”, ou seja, o acesso rápido à página original iria minimizar esse problema. Outra situação que justifica mostrar a página original é quando há remoção de objetos, pois a ferramenta não apresenta *feedback* neste caso. A única forma de ver se um elemento foi removido da interface é olhando a lista de ações registradas na *toolbar* da primoTiWIM.

Diferenças de Vocabulário

Um ponto importante no experimento foi a observação do uso de um vocabulário específico de designers e programadores de sistemas computacionais. Por exemplo, Vitor e Raquel, respectivamente, escreveram:

*“Apesar de não ter sido representado, o **radio button** indeferido deverá ficar abaixo do **radio button** Aguardando”.* (Fala de Vitor)

*“Sugiro mudar o **radio button** para um **checkbox**”.* (Fala de Raquel)

Ou seja, as palavras **radio button** e **checkbox** são palavras que fazem parte do vocabulário de programadores. Inclusive, no experimento com usuários finais (Estudo Empírico I), esse fato foi pouco evidenciado. Isto aponta para a possibilidade de que podemos ter problemas na troca de mensagens entre designers e usuários finais.

Comparação com Outras Tecnologias

Um assunto discutido na entrevista final foi a comparação da primoTiWIM com outras tecnologias. Rute e Raquel refletiram sobre a

diferença entre usar a primoTiWIM para comunicar a ideia ou imprimir a tela e rabiscá-la.

“Se for pensar como é que esse tipo de coisa é feita hoje... eu dou um print na tela, pego o papel e vou lá no lado mostrar para a pessoa”. (Fala de Raquel)

Mas ela lembrou que a comunicação via papel não permite a representação de signos dinâmicos como, por exemplo, associar um *script* a algum elemento da interface: *“usar a coisa de associar... você não tem como fazer”*. Ao final elas concluíram que:

“Para um trabalho rápido e remoto eu imagino que é melhor copiar e colar aqui (se referindo a primoTiWIM) do que copiar e colar no Paint”. (Fala de Raquel)

Outra comparação foi realizada por Vitor. Ele levantou a possibilidade de usar o Balsamic (Balsamic, 2010) (um software para construir protótipos de interfaces) para expressar a ideia, ao invés da primoTiWIM, ele alegou que com o Balsamiq poderia ser criado qualquer tipo de elemento. Leo discordou, dizendo que fazer a alteração partindo da página real é mais interessante.

Nível de Descrição

Ao final da entrevista, pesquisadores e participantes discutiram de maneira geral sobre o modelo proposto. Perguntamos explicitamente se eles descreveriam a proposta de modificação com o mesmo nível de abstração caso não seguissem este modelo. A maioria respondeu que não, por exemplo:

“De forma alguma”. (Raquel)

“Com certeza não”. (Rute)

“Provavelmente eu iria só fazer uma descrição textual... Eu ia levar mais tempo se eu tivesse que detalhar muito... Eu ia ter que pensar mais para escrever”. (Vitor)

“É mais fácil fazer na tela do que escrever isso tudo”. (Maria)

Porém, Rute e Raquel acrescentaram que existem problemas simples que não precisariam alterar páginas para comunicar a solução, e nesses casos não seria necessário o uso deste modelo. É importante lembrar que a representação concreta e detalhada da intenção do design não é obrigatória, exatamente para atender a estes casos citados por Rute e Raquel. No exemplo ilustrativo apresentado anteriormente, a última mensagem não incluiu anexo com representação concreta do seu discurso.

Outro assunto discutido foi se o modelo ajudou a comunicar. A maioria dos participantes disseram que o preenchimento dos elementos do modelo, em alguns casos, guiou a construção da mensagem. Por exemplo, ao perguntar a Maria se o modelo facilitou a comunicação, ela respondeu:

“O fato de eu ter que ter feito isso (referindo ao preenchimento dos dados da mensagem) fez eu me preocupar com alguns fatos... me fez dar um direcionamento para minha resposta”. (Fala de Maria)

Assim concluímos este capítulo onde apresentamos a concretização do modelo proposto e os estudos empíricos realizados. Os significados do que verificamos nos estudos empíricos serão discutidos no capítulo seguinte.