

6 Considerações Finais

Embora os sistemas de ajuda tenham se tornado um componente importante de software, filosofias de design por trás destes sistemas têm sido limitadas (Moily et al., 1987). Esta afirmação, do final da década de 80, continua atual nos dias de hoje. Ao mesmo tempo que pesquisadores clamam por padrões para construção de sistemas de ajuda efetivos, a pesquisa nesta área é quase inexistente atualmente.

Este trabalho visa fornecer aos designers de software ferramentas epistêmicas para a construção de sistemas de ajuda, baseadas em uma teoria sólida de IHC, a teoria da Engenharia Semiótica. Estas ferramentas possibilitam aos designers entender melhor o seu problema – a construção da ajuda – e refletir sobre ele. Estas ferramentas são aqui caracterizadas por princípios teóricos, modelos e arquiteturas focadas no apoio à construção do discurso embutido na ajuda e nas possíveis formas de expressão do usuário frente a este discurso.

As questões de eliciação do ponto de vista do designer (sua lógica de design) permitem que ele não apenas forneça este tipo de informação quando questionado, mas, principalmente, que ele reflita sobre o seu teor, o que impacta na construção da aplicação em si, e não somente no sistema de ajuda.

Já o uso de modelos de design prevendo este conteúdo (informações obtidas da eliciação) possibilita não só que ao final do processo de design as informações necessárias à construção da ajuda estejam disponíveis, mas que sua construção (modelos), durante o processo de design, leve os designers a refletir sobre o design em questão. A análise das informações contidas nos modelos e na ajuda previamente gerada, durante o processo de design, dá aos designers uma visão do design como um todo, podendo permitir, também, um rastreamento da memória da equipe de design durante este processo (decisões tomadas e descartadas, decisões rejeitadas, e assim por diante).

Cabe ressaltar aqui que o enfoque desta tese está no design do sistema de ajuda e como o mesmo deve ser feito durante o processo de design da interação. O enfoque não está no design de software como um todo, embora destaque-se a importância de existir a construção da ajuda como uma das etapas do desenvolvimento, bem como seu teste/avaliação. Tanto é importante esta integração que – no lugar de usar-se um modelo específico de ajuda – adotou-se a ideia de obter-se as informações necessárias a construção da ajuda de modelos (modelo de domínio, de aplicação e de tarefas, dentre outros) já utilizados para outros fins dentro deste processo.

Além do uso das questões de eliciação e sua reflexão nos modelos de design, pensar o sistema de ajuda como uma conversa entre o preposto e seus usuários (discurso do preposto e acesso a este discurso via expressões), conversa esta baseada em conceitos e princípios teóricos, apresentando formas (arquiteturas) alternativas de apresentar esta mensagem e mantendo-se seu teor (lógica de design), possibilita ao designer construir uma comunicação mais coesa, elaborada e extensa sobre a lógica de seu produto, sobre sua utilidade e sobre as formas e situações para utilizá-lo. Desta forma, permite-se que os usuários, compreendendo a aplicação, possam fazer melhor uso de suas facilidades e funcionalidades e, não compreendendo determinadas características da mesma, possam aprofundar seu conhecimento a respeito.

O desenvolvimento destas ferramentas epistêmicas e o detalhamento do discurso do preposto trouxe a necessidade de avançar a teorização de Engenharia Semiótica neste sentido, o que causou a extensão do seu modelo atual, tornando explícita esta metacomunicação designer→preposto↔usuário.

Em relação ao uso de sistemas de ajuda por parte de seus usuários, o que pode verificar-se, tanto nas análises com usuários quanto nas entrevistas com profissionais com diferentes papéis no processo de design, é que as pessoas, definitivamente, não têm o costume de usar o sistema de ajuda das aplicações que utilizam. E a análise das interações dos participantes com a ajuda criada para o estudo de caso mostrou que, mesmo quando induzidos a procurar informações na ajuda, se os signos utilizados para o acesso à mesma não são ostensivos, eles não são nem percebidos.

Os signos utilizados para acesso à ajuda local nesta aplicação eram discretos, presumindo-se uma cultura estabelecida de que o sistema de ajuda é algo “comum” no dia-a-dia das pessoas, e que as mesmas, se necessário, saberiam onde procurá-lo. O que se verificou é que esta não é ainda uma cultura estabelecida, ou seja, que se necessita de um discurso introdutório para apresentar esta tecnologia (de Souza, 2003), fazendo com que os usuários entendam seu propósito e o quanto podem se beneficiar com ela. Na análise do uso da ajuda nesta aplicação pode-se notar que a maioria das pessoas que usaram a ajuda local (e tiveram benefício com esta ajuda) o fizeram depois de ir à ajuda geral e “ler” a informação que explicava o que a ajuda local significava e como ter acesso a ela. Isto, bem como a dificuldade de muitos usuários principiantes em conseguir utilizar a aplicação como um todo, levou ao processo de redesign desta aplicação (atualmente em elaboração), onde muitas das informações de ajuda aos usuários estão explícitas na própria interface da aplicação, em forma de mensagens diretas.

Por fim, deve-se ressaltar novamente que o que estas ferramentas epistêmicas (conceitos, princípios, modelos e arquiteturas) têm em comum é o objetivo de instigar o designer a refletir sobre o **teor** dos sistemas de ajuda. Teor este fundamentado na teoria que embasa este trabalho, que é passar – através do sistema de ajuda – a lógica de design, os aspectos estratégicos da tecnologia, permitindo que o usuário compreenda melhor o que o designer estava pensando quando introduziu esta tecnologia, pois, entendendo os “porquês” do design, o usuário estará mais apto a entender e melhor fazer uso desta tecnologia em proveito próprio.

No decorrer deste trabalho surgiram diversas questões, as quais norteiam os próximos passos desta pesquisa. Estes serão vistos a seguir, em termos de passos a serem seguidos a curto, médio e longo prazo.

Próximos Passos a Curto Prazo

A extensão do modelo de Engenharia Semiótica, tornando explícita a metacomunicação designer→preposto↔usuário, trouxe, também, implicações aos

princípios da Abstração Interpretativa e do Contínuo Semiótico (de Souza et al., 2001).

O princípio da Abstração Interpretativa diz que uma linguagem é uma abstração interpretativa de outra linguagem se uma pessoa conseguir entender a primeira sem conhecer a segunda, utilizando, para isto, apenas sua experiência de uso; o conteúdo da ajuda, tutoriais e documentação do sistema; e sua experiência de mundo e sua experiência computacional.

O sistema de ajuda, conforme proposto neste trabalho, vem modificar a porção desta definição referente ao “conteúdo da ajuda, tutoriais e documentação do sistema” necessário para compreendê-la, por ele (sistema de ajuda), por si só, abranger todo este conhecimento sobre a aplicação, não sendo mais necessárias documentações extra-aplicação. Com isto é necessário, então, refinar-se a definição deste conceito em relação ao papel do sistema de ajuda.

Apesar do “espírito” do sistema de ajuda ser “ele próprio” a abstração interpretativa da linguagem de interface, ou seja, poder entender-se o sistema de ajuda sem conhecer a linguagem de interface, este princípio não pode ser usado na relação “sistema de ajuda ser a Abstração Interpretativa da linguagem de interface”, por o sistema de ajuda não ser uma linguagem e sim uma meta-linguagem (o princípio se aplica a duas linguagens). E, estudando-se o princípio do Contínuo Semiótico, conclui-se que este também não se aplica ao sistema de ajuda, por este (sistema de ajuda) não especificar semanticamente uma outra linguagem.

Dadas estas conclusões, além do refinamento do conceito da Abstração Interpretativa, um dos sub-produtos desta pesquisa foi a proposta, ainda incipiente mas promissora, de que se formule um novo princípio, o **princípio do Contínuo de Discurso**, o qual se aplica a uma linguagem que explica (fala sobre) outra linguagem, ou seja, se aplica à meta-linguagem de uma linguagem.

Um estudo mais aprofundado destes princípios e seu refinamento e formalização são questões de pesquisa que serão tratadas a curto prazo.

Próximos Passos a Médio Prazo

Outras questões levantadas no decorrer desta pesquisa carecem de uma pesquisa mais aprofundada. Dentre elas podemos citar as que estão descritas a seguir.

Conjunto de Expressões

No sistema de ajuda construído no estudo de caso apresentado, foi utilizado apenas um subconjunto das expressões de acesso à ajuda, por nem todas elas serem adequadas à aplicação que de fato se implementou no Projeto ORÉ.

Sistemas de ajuda para aplicações distintas necessitam ser construídos, a fim de verificar o uso de todo o conjunto de expressões e se algumas destas expressões não fazem sentido em nenhum caso, ou se novas expressões devem ser criadas.

A idéia não é utilizar-se, sempre, todas as expressões em uma aplicação, mas, sim, ter um conjunto bem definido de expressões, de seus propósitos e de como implementá-las (acesso e resposta) a fim de que, para cada aplicação, sejam instanciadas as expressões mais adequadas.

Modelos de design

Conforme novas expressões sejam implementadas e/ou criadas, deve-se verificar se as informações contidas nos modelos são suficientes para respondê-las, ou se novas informações necessitam ser incorporadas aos mesmos.

Além disto necessita-se fazer um estudo mais aprofundado em métodos e modelos já utilizados em larga escala por desenvolvedores de software, para ver o que seria necessário acrescentar aos mesmos para sua utilização como fonte de informação para construção de sistemas de ajuda.

Outro ponto a ser estudado é uma maior coesão entre os modelos aqui apresentados com os modelos definidos por (de Paula, 2003) a fim de utilizar-se – se possível – o mesmo conjunto de modelos ou, então, verificarem-se as mudanças a serem realizadas (em um ou em outro conjunto de modelos) para que

isto possa ser feito. Cabe destacar aqui, novamente, que o modelo de interação utilizado nesta tese não foi o mesmo apresentado em (Barbosa et al., 2002b), por ainda estarem sendo discutidas pelo grupo principalmente questões sobre o nível de abstração do mesmo e como isto se reflete na construção da ajuda, da forma como ela está sendo aqui apresentada.

Ainda em relação aos modelos, para facilitar seu estudo e ser possível fazer uma estimativa do custo real de desenvolver sistemas de ajuda baseados em modelos, precisam ser desenvolvidas ferramentas que auxiliem na sua criação, embutindo as questões de ajuda a eles relacionadas.

Geração

A partir das informações contidas nos modelos de design pode ser feita uma geração prévia do rascunho dos conteúdos da ajuda. Esta geração está sendo feita de maneira bastante rudimentar, necessitando um estudo mais aprofundado a fim de que os textos gerados, base para a construção da ajuda, sejam melhor elaborados.

Além disto, precisam ser estudadas formas de fazer a geração automática (e dinâmica) das expressões que serão disponibilizadas através da ajuda local, com base no *status* corrente da aplicação, o que atualmente é feito previamente, de forma manual.

Outro item previsto anteriormente (Silveira et al., 2001a) e ainda não implementado, é a geração de uma lista de FAQ (*Frequently Asked Questions*), a partir das expressões usadas por diferentes usuários da aplicação.

Arquiteturas e Alternativas de Respostas

A arquitetura apresentada neste trabalho, bem como as respostas às expressões de acesso à ajuda foram definidas em cima de porções textuais de informação, onde o objetivo das mesmas era trabalhar a função comunicativa da ajuda, com ênfase no teor da mensagem do designer.

Novas formas de resposta a estas expressões necessitam ser estudadas, explorando-se bastante a função conativa (uma das funções de (Jakobson, 1960),

detalhadas no capítulo 3) desta mensagem, ou seja, dando voz ao designer permitindo-lhe literalmente “chamar” o usuário a verificar como uma tarefa pode ser realizada na interface da aplicação ou, até mesmo, a realizar algo a partir da ajuda. As possibilidades de incluírem-se demonstrações, cenários e exemplos práticos, também devem ser estudadas.

Além disto, como componentes da arquitetura, pode-se, também, estudar formas de incluir sistemas de anotações, para o usuário fazer a “sua” ajuda particular, e *bookmarks*, para que ele possa guardar *links* a informações já visitadas, por exemplo.

O importante é ressaltar que a arquitetura vista não é a arquitetura para a abordagem aqui apresentada, e, sim, uma possível arquitetura.

Novas arquiteturas podem e devem ser criadas na tentativa de, cada vez mais, explorar o poder comunicativo da ajuda.

Uso de Tutoriais

Outro ponto a ser estudado é o uso de sistemas tutoriais – conforme os mesmos vêm sendo utilizados para ensino e/ou treinamento – e a possível aplicação de seus princípios para melhorar a forma como a ajuda geral está sendo apresentada atualmente, e, até mesmo, para as mensagens que devem ser apresentadas diretamente na interface da aplicação.

Próximos Passos a Longo Prazo

Dois grandes trabalhos de pesquisa a serem realizados são a definição de sistemas de ajuda para aplicações de *groupware* e para aplicações que permitam programação por usuários finais.

Ambas necessitam um estudo bastante aprofundado, pois têm características muito diferentes da abordagem aqui proposta, destinada a um único usuário.

Em aplicações para *groupware*, mesmo as expressões a serem utilizadas em testes de comunicabilidade para estes tipos de ambiente ainda não foram

completamente definidas, existindo diversos fatores que atuam sobre a maneira como um grupo de pessoas se comunica com uma aplicação e/ou através desta aplicação. Algumas considerações sobre avaliação de comunicabilidade em ambientes de grupo podem ser vista em (Prates et al., 2001).

Já aplicações que permitem programação por usuários finais, além da ajuda sobre como utilizar estas facilidades, devem permitir, também, que o próprio usuário estenda (ou customize) o sistema de ajuda quando estender (ou customizar) a aplicação. E isto não pode interferir nem no trabalho das demais pessoas com a aplicação, nem no núcleo a ser preservado da mesma.

Estas possibilidades de extensão (tanto da aplicação em si quanto de seu sistema de ajuda) são muito importantes quando considera-se a questão da re-significação por parte dos usuários. Usuários nem sempre percebem o significado pretendido pelo designer, dando-lhe novos significados, além de estarem, sempre, num processo de criação de novos significados (uma possibilidade alternativa de uso de um conceito já conhecido, por exemplo).

As aplicações (e seus sistemas de ajuda) devem conseguir acompanhar esta evolução de seus usuários. De uma forma simples isto pode ser feito permitindo-lhes fazer anotações sobre suas “descobertas” (para futura recordação das mesmas) e, de uma forma mais elaborada, permitindo-lhes estender suas aplicações de forma que estas reflitam seu pensamento atual. E, neste caso, o sistema de ajuda também deve refletir estas novas significações.

Seguindo estes passos, no aprofundamento destas questões de pesquisa, quem sabe em um futuro não tão distante consigamos responder afirmativamente a uma questão levantada no início da década de 80: *Será que a assistência online será boa para quem não conhece computação?* (Houghton Jr., 1984), a qual, infelizmente, continua sem resposta.