

2 Trabalhos Relacionados

Este capítulo apresenta, na seção 2.1, a Engenharia Semiótica, teoria de IHC na qual a MoLIC se originou (de Souza, 2005) e, na seção 2.2, a perspectiva de reflexão-em-ação proposta por Schön (1983), que desejamos encorajar.

Antes de apresentarmos a proposta deste trabalho, fornecemos na seção 2.3 um panorama sobre os modelos e representações mais comumente utilizados em IHC que podem servir como insumo ou complementar uma modelagem MoLIC. Apesar da proposta deste trabalho não ser um método de avaliação, apresentamos também, na seção 2.4, duas abordagens de avaliação de sistemas computacionais interativos particularmente interessantes de serem discutidas: a avaliação por inspeção (seção 2.4.1), por ser o tipo de avaliação mais propenso a estimular a reflexão do avaliador, e a avaliação baseada em modelos (seção 2.4.2), já que este trabalho consiste em uma forma de inspecionar modelos de interação MoLIC.

2.1. Engenharia Semiótica

Este trabalho é fundamentado na teoria da Engenharia Semiótica, que caracteriza a Interação Humano-Computador como um caso particular de comunicação humana mediada por computador (de Souza, 2005). Esse processo de comunicação envolve designers e usuários de sistemas computacionais, pois eles têm um *status* especial em IHC: os designers devem ser capazes de embutir no artefato que estão produzindo a gama de mensagens que são esperadas e úteis para todos os *stakeholders*² (incluindo eles mesmos), e os usuários são os juízes finais da qualidade desse artefato. Dessa forma, a Engenharia Semiótica designa para designers e usuários o mesmo papel em IHC: o de interlocutores em um processo comunicativo completo.

² Qualquer pessoa que esteja direta ou indiretamente envolvida no desenvolvimento ou uso do sistema, tais como: clientes, gerentes de projeto, usuários, designers, redatores técnicos etc.

Esse processo de comunicação começa quando o designer estuda os usuários, suas atividades e seu ambiente. O designer, então, consolida todo esse conhecimento em sua interpretação de quem são os usuários, do que eles precisam, o que eles gostam e preferem etc, e o articula com o seu conhecimento técnico e criatividade, produzindo um artefato que conta esse conhecimento sobre os usuários na forma de uma mensagem interativa. Ao interagir com o sistema, os usuários recebem essa mensagem do designer, compreendendo gradualmente os diferentes significados codificados no artefato, e respondem a ela.

A partir dessa visão, as aplicações computacionais interativas são consideradas artefatos de metacomunicação (de Souza, 1993; Leite, 1998), pois os designers enviam uma mensagem que conta para os usuários como eles podem ou devem se comunicar com ela para alcançar os efeitos desejados. O conteúdo dessa mensagem pode ser organizado no seguinte modelo:

“Eis a minha interpretação de quem você é, o que aprendi que você quer ou precisa fazer, preferencialmente de que forma, e por quê. Eis, portanto, o sistema que conseqüentemente concebi para você, o qual você pode ou deve usar assim, a fim de realizar uma série de objetivos associados com esta minha visão.”³

Essa conversa entre designer e usuário é realizada em tempo de interação através da interface do artefato produzido, por meio de inúmeras mensagens codificadas em palavras, gráficos, comportamento, ajuda online e explicações. No entanto, como o designer não pode estar presente quando os usuários utilizam a aplicação, ele é representado pelo preposto do designer, responsável por contar suas intenções para os usuários em tempo de interação.

Nesse contexto, é importante ressaltar o conceito de semiose. Quando um indivíduo percebe um signo, que é qualquer coisa que signifique algo para alguém (Peirce, 1992; 1998) e tenta interpretá-lo, é gerado um pensamento ou idéia que

³ Original em inglês: “Here is my understanding of who you are, what I’ve learned you want or need to do, in which preferred ways, and why. This is the system that I have therefore designed for you, and this is the way you can or should use it in order to fulfill a range of purposes that fall within this vision.” (de Souza, 2005: 84).

corresponde ao significado (o interpretante) temporário concebido sobre o signo por esse indivíduo. Esta idéia, por sua vez, dá origem a outras idéias associadas, iniciando um processo de interpretação (atribuição de significado) chamado de semiose (Peirce, 1992; 1998). Daí surge o princípio da semiose ilimitada, definida por Eco (1984) como esse processo interpretativo de signos que, potencialmente, pode nunca ter fim. Na comunicação, esse princípio determina que a semiose continuará enquanto as partes comunicativas estiverem ativamente engajadas em um entendimento mútuo.

No âmbito da Engenharia Semiótica, o processo de codificação da mensagem de metacomunicação do designer como um sistema computacional captura e cristaliza apenas uma parte da semiose do designer (aquela que pode ser codificada em uma linguagem computacional naquele momento da construção do sistema), estabelecendo, dessa forma, o limite entre a semiose humana e computacional.

A metacomunicação designer-usuário só é completamente atingida se, ao interagirem com a mensagem, os usuários compreenderem a metamensagem da forma como projetada pelo designer e se comunicarem com sucesso com ela. Caso o usuário não consiga interpretar adequadamente as intenções que o designer tinha ao construir o artefato computacional, dizemos que ocorrem rupturas na comunicação designer-usuário. O designer deve tentar antecipar essas rupturas e fornecer meios para reestabelecer a comunicação de forma que o usuário possa continuar utilizando o sistema para atingir seus objetivos.

Usando a Engenharia Semiótica para estudar, analisar e tomar decisões sobre os usuários e suas reações esperadas, os designers estão simultaneamente estudando, analisando e tomando decisões sobre o seu próprio comportamento e estratégias comunicativos. Dessa forma, a Engenharia Semiótica é uma teoria eminentemente reflexiva, que traz explicitamente os designers para a etapa de processos de IHC e lhes designa uma posição ontológica tão importante quanto a dos usuários, envolvendo-os no mesmo fenômeno comunicativo (de Souza, 2005).

2.2. Reflexão-em-Ação

A natureza reflexiva da Engenharia Semiótica pode ser facilmente articulada com a perspectiva de design de reflexão-em-ação de Schön (1983), de acordo com a qual o conhecimento prático envolve a capacidade de nomear os elementos presentes em situações problemáticas e enquadrá-los como problemas. Nesta perspectiva, um problema é considerado único, assim como a atividade de design e a solução encontrada. Dessa forma, o designer não está procurando uma solução padrão, mas sim características particulares de sua situação problemática, para, a partir de sua descoberta gradual, projetar sua intervenção.

Caso não encontre uma solução que o satisfaça, o designer critica não apenas as soluções que se apresentam, mas também sua própria formulação do problema, e experimenta reformulá-lo. Essa reformulação pode modificar os elementos envolvidos na situação problemática e seus significados de forma não intencionada pelo designer, exigindo que o designer permaneça refletindo e descobrindo quais são as conseqüências e implicações desta reformulação para a próxima tentativa de resolver o problema.

Enquanto expressa a solução em alguma representação, o designer “conversa” com a representação, ao que Schön denominou “*conversation with materials*” (ou conversa com materiais). Em outras palavras, a reflexão em ação ocorre enquanto o designer conversa com a representação, refletindo, avaliando e aprendendo sobre o que está fazendo enquanto o faz, de tal forma que essas reflexões influenciem ações futuras em direção à concepção da solução.

Segundo de Souza (2005), a maior contribuição teórica que a Engenharia Semiótica pretende dar é endereçar necessidades epistêmicas tanto dos designers quanto dos usuários quando estão se comunicando através de tecnologias baseadas em computador. Para isso, a Engenharia Semiótica propõe a utilização de ferramentas epistêmicas, que, ao contrário das que geram diretamente a resposta para o problema, visam aumentar o conhecimento de quem está lidando com o problema, sobre o problema em si e as implicações que isso traz.

A Engenharia Semiótica tem diferentes tipos de ferramentas epistêmicas que apóiam as atividades centradas no conhecimento: modelos interpretativos, princípios analíticos e métodos analíticos. A teoria enfatiza os modelos e outras

representações de conhecimento produzidos e utilizados em IHC, o que também ocorre neste trabalho. Dessa forma, será apresentada na seção seguinte uma ferramenta epistêmica utilizada para modelar a interação humano-computador desenvolvida com base na Engenharia Semiótica.

2.3. Modelos e representações

Um modelo é uma descrição ou abstração do mundo real. De acordo com Diaper (2004), um modelo representa o mundo como sendo feito de algo, chamado coisas, objetos, entidades, itens, componentes, pedaços, partes, atributos, propriedades etc. Representações e modelos são de grande valia para especificar a interação e interface de sistemas interativos, mas a escolha de uma representação específica depende dos objetivos da análise e do nível de abstração desejado.

No projeto de IHC, diferentes modelos e representações podem ser complementados para que se consiga representar melhor as necessidades dos usuários em relação ao sistema que está sendo desenvolvido. Assim sendo, descrevemos nas seções a seguir as representações e modelos mais comumente utilizados em IHC que podem ser utilizados antes ou em conjunto com a MoLIC, como é o caso de cenários (seção 2.3.1), personas (seção 2.3.2) e modelos de tarefas (seção 2.3.3), bem como os próprios modelos de interação (seção 2.3.4), categoria na qual a linguagem MoLIC se encaixa.

2.3.1. Cenários

Cenários têm sido utilizados como uma ferramenta poderosa de design, sendo empregados em várias disciplinas, incluindo IHC, engenharia de requisitos, design orientado a objeto e planejamento estratégico. Um cenário é uma descrição que contém atores, informação de contexto sobre eles e suposições sobre seu ambiente, suas metas e objetivos, e seqüências de ações e eventos. É compartilhado por vários *stakeholders* no design do sistema e pode ser expresso em várias mídias e formas: narrativas textuais, *storyboards*, demonstrações de vídeos ou protótipos.

Cenários são orientados ao uso, integrados e flexíveis, podendo ser facilmente desenvolvidos, compartilhados e manipulados (Go & Carroll, 2004). Por outro lado, por serem tipicamente narrativas textuais, estão mais suscetíveis a problemas de ambigüidade e incompletude do que outras representações mais rígidas e sistemáticas.

Os autores da MoLIC afirmam que os cenários podem ser utilizados em conjunto com a MoLIC em situações diversas, tais como: antes da concepção e da modelagem da interação, através de cenários de análise; durante a concepção da interação, na elaboração de situações de interação alternativas; durante a modelagem, na contextualização de situações representadas na MoLIC; e após a modelagem (ou após cada etapa de modelagem), em situações a serem avaliadas pelos usuários (Paula, 2003). Isso é possível porque cenários podem ser vistos como mais uma forma de representar a conversa usuário-sistema. Uma diferença importante entre os cenários e a MoLIC é que os cenários são ricos em contextualização e representam neles próprios a motivação dos usuários para formular seus objetivos, ao passo que a MoLIC não representa explicitamente essa motivação.

2.3.2. Personas

Segundo Cooper (2004), uma equipe de desenvolvimento obtém muito mais sucesso projetando para uma ou poucas pessoas importantes no domínio em questão ao invés de desenvolver o maior número de funcionalidades possível para acomodar a maioria das pessoas. De acordo com essa visão, Cooper criou as personas, que são arquétipos, representações de pessoas reais, utilizadas durante todo o processo de design. Personas são baseadas em dados etnográficos dos usuários, pois se acredita que combinar entrevistas com observação direta – preferencialmente no contexto de uso real –, focando o que os usuários querem, o que os deixam frustrados, e o que os dá satisfação, é mais eficiente do que apenas perguntar aos usuários o que eles querem (Goodwin, 2002). De certa forma, o conceito de personas está em linha com a Engenharia Semiótica, pois promove o foco no particular, como prega a Engenharia Semiótica e abordagens de cunho

etnográfico, e não no universal, como prega boa parte das abordagens de base cognitiva.

Em geral, uma persona possui um nome fictício, uma foto e características consistentes a um dos perfis de usuários identificados. Tais características incluem: idade, nível de escolaridade, etnia, profissão, objetivos e tarefas relacionados ao domínio do problema, ambiente físico, social e tecnológico em que se encontram. Além disso, é comum ter uma citação que resume o que é mais importante para a persona em relação ao domínio do problema, bem como uma breve descrição textual e palavras-chave.

As personas devem ser definidas com rigor e precisão, pois, quanto mais específicas, mais eficientes são como ferramenta de design. Cooper também observa uma outra contribuição muito importante das personas: seu valor como ferramenta de comunicação. Segundo ele, as personas se tornam uma taxonomia de design com grande capacidade de explicar as decisões de design.

De acordo com Pruitt & Adlin (2006), a utilização de personas traz diversos benefícios para o desenvolvimento de sistemas. Personas são cognitivamente convincentes, pois apresentam um rosto humano personalizado ao invés de informações abstratas sobre os usuários. Além disso, ao pensar sobre as necessidades de uma persona, os designers podem melhor inferir as necessidades de uma pessoa real. Pruitt e Adlin argumentam, ainda, que as personas são comunicadas facilmente, permitindo, assim, que todos os envolvidos no projeto absorvam as informações dos usuários de forma mais agradável.

Podemos dizer que a utilização de personas foca alguns aspectos da metamsagem, em destaque no trecho abaixo:

“Eis a minha interpretação de quem você é, o que aprendi que você quer ou precisa fazer, preferencialmente de que forma, e por quê.”

Vale dizer que uma persona pode representar a forma preferencial que utiliza para atingir seu objetivo. No entanto, o foco de personas está em responder “o que” a persona faz e “por que” o faz, mas não “como” o faz. Já a MoLIC permite responder “o que” o usuário pode ou deve fazer, e “como”.

Fundamentado nas personas e em seus objetivos, Cooper elabora uma abordagem denominada *goal-directed design* (ou design dirigido por objetivos, DDO), afirmando que, quando os designers analisam objetivos para resolver problemas, eles tipicamente encontram soluções muito diferentes – e muito melhores. Essa abordagem de focar os objetivos e atividades dos usuários (os porquês e comos) está em linha com o pensamento de Donald Norman, que afirma que a atenção exagerada nas características dos usuários em detrimento aos seus objetivos pode levar a uma falta de coesão e a um acréscimo de complexidade no design (Norman, 2005). Segundo Cooper, uma persona existe para atingir seus objetivos, enquanto os objetivos existem para dar significado a uma persona. Os dois são inseparáveis, como os dois lados de uma moeda. Cooper afirma, ainda, que objetivos não são a mesma coisa que tarefas. Um objetivo é uma condição final, enquanto uma tarefa é um processo intermediário necessário para atingir o objetivo. As tarefas mudam conforme a tecnologia, mas os objetivos se mantêm bastante estáveis. Sendo assim, a abordagem de design dirigido por objetivos permitiria redigir a primeira parte da metamensagem (resultante das atividades de análise), mas não se preocupa em transmitir aos usuários as intenções do designer no momento de construção do sistema (e.g., decisões de design).

Assim como os cenários, personas são um recurso que pode ser utilizado durante todo o processo de design para manter a equipe de design informada sobre características relevantes dos usuários-alvo e seus objetivos, e pode dar insumos para os modelos de interação MoLIC, em especial para o diagrama de metas, a partir dos objetivos identificados de cada persona.

2.3.3. Modelos de tarefas

Para se construir um modelo de tarefas, primeiro é necessário fazer uma análise de tarefas. A maioria das formas de análise de tarefas recai na idéia de que a ação humana pode ser decomposta e que esta decomposição pode ser usada para se pensar sobre o que as pessoas deveriam fazer e saber para completar uma tarefa (Barnard & May, 2000). A análise de tarefas é usada para analisar a funcionalidade de sistemas em termos das metas e inerentes sub-metas do usuário ao executar a tarefa (Ormerod, 2000). Para tanto, oferece um método simples para

permitir que as tarefas e sua ordem sejam descobertas, representando-as em modelos de tarefas.

Modelos de tarefas descrevem como executar as tarefas dos usuários, normalmente em uma estrutura hierárquica, para que estes possam alcançar seus objetivos. Desta forma, um modelo de tarefas registra o resultado da discussão entre os diferentes atores envolvidos no projeto (Paternò, 2000). As representações e modelos de tarefas mais conhecidos são: GOMS: *Goals, Operators, Methods and Selection Rules* (Card et al., 1983), TKS: *Task Knowledge Structures* (Johnson et al., 1988), MAD: *Méthode Analytique de Description* (Scapin & Pierret-Golbreich, 1989), GTA: *Groupware Task Analysis* (van der Veer, 1996), TAG: *Task-Action Grammar* (Payne & Green, 1989), e CTT: *ConcurTaskTrees* (Paternò, 2000).

Paternò (2002) defende que cenários e modelos de tarefas são complementares e, portanto, suas metodologias (ou ao menos parte delas) deveriam ser integradas para se obter uma melhor análise das tarefas do usuário. No entanto, se em um projeto de IHC forem desenvolvidos apenas cenários ou modelos de tarefas, o designer terá somente representações dos usuários e seu ambiente, das tarefas que eles realizam ou gostariam de realizar, das relações entre essas tarefas e, de forma bem superficial, da maneira como essas tarefas são realizadas através de uma seqüência de passos. Boa parte dos modelos de tarefas mais utilizados atualmente ainda não representam os possíveis erros que os usuários podem cometer, e muito menos as formas deles se recuperarem de tais erros. Assim sendo, em um modelo de tarefas, o designer se concentra na performance correta de usuários competentes, sem se preocupar com a interação real e propensa a dúvidas ou erros desses usuários com o sistema sendo projetado. Essas últimas questões são endereçadas por modelos de diálogo e de interação, categoria na qual a MoLIC se encontra, a serem apresentados na próxima seção.

Como será visto, a MoLIC fornece mecanismos explícitos para o designer lidar com o desempenho tido como “incorreto” dos usuários. A Engenharia Semiótica reconhece a natureza imprecisa da comunicação com pessoas e a impossibilidade de prever todas as diferentes interpretações que os usuários poderão ter durante sua interação com o sistema. Em linha com a Engenharia Semiótica, a MoLIC foi elaborada para motivar o designer a projetar interfaces que lidem adequadamente com rupturas na comunicação.

2.3.4. Modelos de diálogo e de interação

Há modelos que descrevem a interação do usuário com o sistema, tais como os modelos de diálogo (Puerta, 1997; Clarke & Crum, 1994; Green, 1986), e os modelos de interação (Beaudouin-Lafon, 2000; Hix & Hartson, 1993). No entanto, tais abordagens têm o objetivo de especificar a estrutura e o comportamento da interface em um baixo nível de abstração. Em outras palavras, estes modelos referem-se aos comandos de interface que o usuário pode executar (i.e. cliques de *mouse*, operações de manipulação direta etc.) e às correspondentes respostas do sistema (Silva & Barbosa, 2005). Neste trabalho, porém, é utilizado o conceito de modelo de interação proposto em (Paula & Barbosa, 2003), que não apresenta detalhes de interface ou de plataforma tecnológica, e que será descrito em detalhes no capítulo 3.

Existem ainda outras representações utilizadas em IHC, tais como os modelos de interface – como por exemplo, linguagens de descrição de interface abstrata, tal qual a UsiXML⁴ –, sobre os quais não entraremos em detalhes por serem construídos em etapas posteriores à modelagem da interação e não terem relação direta com este trabalho.

Na seção seguinte, serão apresentadas duas abordagens de avaliação de sistemas interativos. Apesar deste trabalho não apresentar uma proposta de avaliação, as abordagens de avaliação de sistemas interativos também suscitam reflexões sobre a interação no profissional que realiza a avaliação, o que está fortemente relacionado a este trabalho.

⁴ <http://www.usixml.org/>

2.4. Avaliação de Sistemas Computacionais Interativos

A avaliação é uma etapa essencial no processo de desenvolvimento de sistemas computacionais interativos, pois é através dela que obtemos indicações do grau de sucesso do processo ou do produto resultante (Preece et al., 2002). Além disso, é muito importante que os resultados da avaliação sejam utilizados para futuros refinamentos deste processo ou produto. No entanto, não são raras as situações em que esta etapa é negligenciada, pois (a) nem sempre são reservados recursos específicos para sua execução; (b) mesmo quando realizada, geralmente tem caráter remediador e não preventivo; e (c) independente da forma como é realizada, nem sempre tem seus resultados utilizados para guiar futuros refinamentos.

A avaliação de um sistema interativo é classificada por Hix & Hartson (1993) em virtude do momento em que é realizada, podendo ser formativa, quando é realizada ao longo do processo de design, ou somativa, quando é realizada somente nas etapas finais do desenvolvimento. Segundo Nielsen & Mack (1994), a avaliação formativa tem por objetivos prever a usabilidade de alguns aspectos do produto, verificar a compreensão da equipe de projeto sobre os requisitos dos usuários, examinando como um sistema existente está sendo utilizado, e testar idéias de forma rápida e informal. Já a avaliação somativa tem por objetivos identificar problemas de interação e de interface, bem como propor alterações para aumentar a qualidade de uso de um sistema interativo.

Nielsen & Mack (1994) classificam, ainda, os métodos de avaliação existentes em: automático, formal, informal e empírico. Já Preece e co-autoras (2002) concentram-se nos dois últimos, categorizando-os em: entrevistas e questionários para captura da opinião dos usuários, avaliação interpretativa, avaliação preditiva ou por inspeção, observação e monitoramento de uso, experimentos e testes de desempenho.

De modo geral, as avaliações são realizadas visando emitir um julgamento de valor sobre determinado artefato, porém algumas delas permitem também refletir sobre o problema e sobre a própria solução.

Investigaremos a seguir se os métodos de avaliação mais freqüentemente utilizados nas abordagens de avaliação por inspeção (seção 2.4.1) e avaliação

baseada em modelos (seção 2.4.2) apóiam, de forma relevante, alguma reflexão do avaliador ao longo de sua aplicação.

2.4.1.

Avaliação por Inspeção sobre Esboços, Protótipos ou Maquetes

Tendo como base a classificação de Preece e co-autoras (2002), o tipo de avaliação mais propenso a estimular a reflexão do avaliador sobre o problema e a solução concebida é a avaliação por inspeção, pois privilegia a percepção do avaliador, e não do usuário, tal como ocorre em entrevistas, questionários e observações de uso. Em alguns métodos de avaliação por inspeção, tal qual o percurso cognitivo, os possíveis processos mentais dos usuários são levados em consideração, mas, mesmo nesses casos, é o avaliador quem procura antecipá-los. Em outras palavras, em avaliações por inspeção não há evidências *reais* da percepção ou juízo de valor dos usuários que possam influenciar, positiva ou negativamente, a reflexão do avaliador.

Em avaliações por inspeção, os especialistas tentam prever os problemas que os usuários típicos do sistema provavelmente terão ao utilizá-lo. Em IHC, são muito utilizadas as inspeções de usabilidade, que visam encontrar problemas de usabilidade em uma interface existente, a fim de se fazer recomendações para resolvê-los e melhorar a usabilidade do design.

Nielsen & Mack (1994) identificam pelo menos oito métodos de inspeção de usabilidade: percurso cognitivo, avaliação heurística, revisão de *guidelines*, percurso pluralístico, inspeções de consistência, inspeções de *standards*, inspeções formais de usabilidade e inspeções de funcionalidade, sendo os dois primeiros os mais utilizados e, por este motivo, aqueles que serão analisados a seguir.

O percurso cognitivo é um método de avaliação que foca principalmente a facilidade de aprendizado de um sistema, especialmente por exploração. Seu objetivo é antecipar os pensamentos e ações dos usuários quando utilizam uma interface pela primeira vez (Wharton et al., 1994). Ao tentar antever os pensamentos e ações dos usuários através de uma série de perguntas, o avaliador está claramente refletindo sobre o problema em questão, bem como sobre a solução apresentada na interface, maquete ou protótipo. A cada passo de interação

previsto para o atingimento de um determinado objetivo, os avaliadores buscam responder (e justificar sua resposta), tendo em vista um determinado perfil de usuário alvo, as seguintes questões:

1. O usuário tentará atingir a meta correta?
2. O usuário perceberá que a ação correta está disponível na interface?
3. Uma vez encontrado o elemento de interface, o usuário reconhecerá que ele produzirá o efeito desejado?
4. Após a ação correta ser executada, o usuário perceberá que progrediu em direção à conclusão da tarefa?

O resultado de um percurso cognitivo não é apenas o conjunto de respostas a essas perguntas, mas também as explicações sobre os problemas encontrados, que podem ser úteis para reprojeter a aplicação.

Já o método de avaliação heurística foi criado como uma alternativa rápida e de baixo custo para encontrar problemas de usabilidade em um projeto de interface como parte de um ciclo de desenvolvimento iterativo, a partir de conhecimento de base empírica tido como “boas práticas”. Nesse método, poucos avaliadores examinam e julgam se a interface obedece a determinados princípios de usabilidade, as chamadas “heurísticas” (Nielsen & Mack, 1994). Tal método de avaliação não promove tanto a reflexão quanto o percurso cognitivo, pois consiste, basicamente, na comparação de elementos de interface a uma lista de heurísticas de usabilidade. No entanto, o conjunto inicial de heurísticas pode ser expandido pelos avaliadores para incluir princípios relevantes para o estilo de interação em questão. Nesses casos em que ocorre a adaptação ou inclusão de novas heurísticas, o processo de análise do conjunto ideal de princípios poderá levar a uma maior reflexão sobre os problemas a serem endereçados e sobre as possíveis soluções dentro daquele contexto.

Como é possível perceber, existem diversos métodos de avaliação por inspeção, porém a grande maioria deles avalia a *usabilidade* da interface concreta. Na Engenharia Semiótica foi desenvolvido o Método de Inspeção Semiótica (de Souza et al., 2006) cujo objetivo é avaliar a *comunicabilidade* de sistemas interativos. Nesse método, o avaliador examina as diferentes formas da comunicação designer-usuário presentes em uma aplicação, e agrega a essência da metacomunicação oriunda do discurso explícito sobre o sistema, tipicamente

encontrada na documentação e ajuda, com a oriunda do discurso implícito sobre o sistema, tipicamente revelada na interação com o sistema.

Ao contrário do que ocorre no percurso cognitivo e no método de avaliação heurística descritos anteriormente, no Método de Inspeção Semiótica, o avaliador não é guiado por perguntas ou princípios, mas pela metamensagem que procura reconstruir ao longo da inspeção. Ao tentar reconstruir essa metamensagem do designer, o avaliador está refletindo sobre o problema, sobre a solução dada e sobre a forma como o designer escolheu para comunicar essa solução para os usuários. Ao final da inspeção, o avaliador apresenta uma lista de problemas de comunicabilidade do sistema avaliado, com possíveis soluções. Em princípio, o artefato avaliado no método de inspeção semiótica é a interface (semi-) funcional, mas admite-se a possibilidade do método ser aplicado em modelos, o que, no entanto, ainda não foi explorado por seus proponentes.

2.4.2. Avaliação Baseada em Modelos

Além de avaliar interfaces, também é possível avaliar modelos, tanto empiricamente quanto por inspeção. Abordagens baseadas em modelos podem auxiliar projetistas e desenvolvedores de interface, destacando aspectos relevantes e fornecendo descrições lógicas das principais funcionalidades de aplicações interativas. Segundo Paternò e co-autores (2001), tais abordagens têm se tornado parte da prática atual no desenvolvimento de interfaces. No entanto, ainda existe uma carência considerável de métodos estruturados para utilizar as informações contidas em modelos para o projeto e a avaliação de interfaces.

Boa parte das avaliações baseadas em modelos utiliza-se de modelos de tarefas, visando produzir predições quantitativas sobre o quão bem os usuários serão capazes de executar suas tarefas. Modelos de tarefas, por exemplo, têm sido alvo de grande interesse para projetar aplicações interativas, pois permitem aos projetistas endereçarem a especificação de aspectos funcionais e interativos de forma integrada. Em (Paternò & Mancini, 2000), defende-se que a especificação deveria ser usada pelo projetista para remover ambigüidades, avaliar opções de design, e verificar a qualidade do projeto como um todo. Isto é particularmente importante quando novas tarefas são disponibilizadas ou quando passa a existir a

necessidade dos usuários executarem tarefas já existentes de formas diferentes. Nestes casos, o modelo de tarefas indica qual parte da aplicação tem que ser modificada e como ela se relaciona com as demais partes.

Avaliação empírica baseada em modelos

Apesar de ser útil para destacar aspectos relevantes, as avaliações baseadas em modelos não incluem informações empíricas. Por este motivo, Paternò & Ballardín (2000) sugerem que é importante encontrar métodos que permitam que os projetistas combinem modelos significativos e informações empíricas. Uma tentativa nesta direção é o USAGE (Byrne et al., 1994), que fornece uma ferramenta de apoio a um método cujas ações dos usuários são analisadas pela abordagem NGOMSL, Natural GOMS Language (Kieras, 1988, 1997a). No entanto, as informações extraídas a partir deste método ainda são limitadas se comparadas com as contidas nos *logs* de ações executadas por usuários durante suas sessões de trabalho. Há, também, trabalhos de derivação de modelos GOMS a partir de análises de *logs* de usuário (Hudson et al., 1999).

O CTT, ou ConcurTaskTrees, é uma notação gráfica desenvolvida por (Paternò, 2000) para especificar modelos de tarefas, que tem o apoio de uma ferramenta computacional, o CTTE⁵, *CTT Environment*. No CTTE, é possível verificar se o modelo está de acordo com as regras sintáticas de construção de modelos de tarefas seguindo a notação do CTT. A Figura 1 indica os erros sintáticos encontrados em um modelo de tarefas desenvolvido na ferramenta. Além de erros, esta verificação pode retornar avisos que indicam, por exemplo, que uma tarefa não pode ter apenas uma filha, o que é possível de se fazer no modelo, mas que semanticamente não faz sentido.

⁵ <http://giove.cnuce.cnr.it/ctte.html> (último acesso em agosto de 2007)

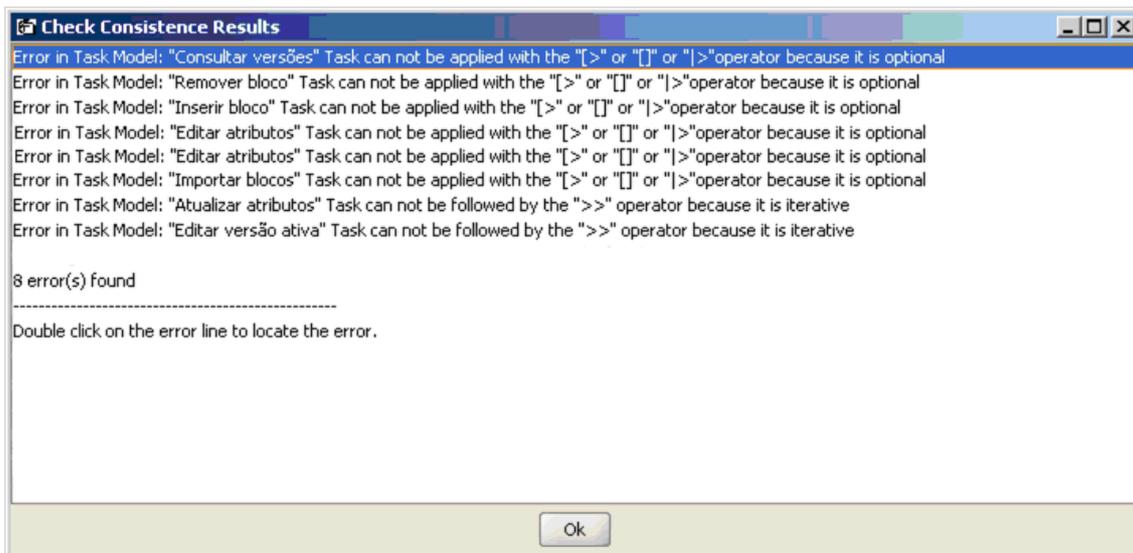


Figura 1: Resultado da verificação de um modelo de tarefas no CTTE.

Há, também, uma ferramenta específica para a avaliação de modelos de tarefas seguindo a notação do CTT: a RemUSINE (Paternò, 2000). Esta ferramenta utiliza-se de modelos de tarefas do CTT não-prescritivos e flexíveis, possibilitando a análise, através de *logs* de sessões de interação dos usuários com o sistema, de possíveis discordâncias entre o comportamento do usuário e o modelo de tarefas da aplicação.

Outra ferramenta automática que ajuda na avaliação de usabilidade através de modelos de tarefas é a ErgoLight (Harel, 1999), que utiliza uma notação simples para especificar os modelos, e, por isso, ainda depende do avaliador para identificar partes problemáticas.

O foco deste trabalho não é a avaliação empírica de modelos, mas sim a inspeção de modelos (em particular, representados em MoLIC) como forma de motivar a reflexão do designer durante a construção da segunda parte da metamensagem. Trata-se de uma avaliação formativa, mas cujo foco não é apenas descobrir problemas, e sim refletir sobre alternativas de design e sobre as consequências que certas decisões de design terão na interação do usuário com o sistema em questão.

Avaliação por inspeção baseada em modelos

Recentemente foi elaborada uma abordagem de avaliação baseada em casos de uso, a UCE, *Use Case Evaluation* (Pedersen et al., 2007), cujo *input* consiste

em uma coleção de casos de uso reais, descrevendo o uso do sistema em desenvolvimento, uma breve descrição do contexto de uso, e uma lista de *guidelines*. Para cada caso de uso, o avaliador tenta prever os problemas de usabilidade que um usuário terá enquanto o percorre. O produto desta avaliação é apresentado como uma lista de problemas de usabilidade.

Outra técnica utilizada para avaliar a qualidade de casos de uso foi proposta em (Tao, 2005), baseada na definição de um modelo de comportamento, apresentado como uma máquina de estados, que foca o fluxo de interação entre o usuário e o sistema enquanto o usuário está executando uma tarefa. Na técnica desenvolvida por Tao, princípios de usabilidade são utilizados como guia para avaliar este modelo de comportamento.

2.4.3. Considerações

Dos métodos de avaliação por inspeção, o percurso cognitivo e o Método de Inspeção Semiótica têm grande potencial de contribuir para este trabalho. O percurso cognitivo ajuda o designer, durante uma avaliação formativa, a antecipar problemas que os usuários poderão enfrentar durante a interação. O que muda, no nosso caso, é principalmente a natureza dos problemas identificados. Nosso trabalho visa apoiar o designer na identificação de possíveis rupturas comunicativas, e não dos problemas de natureza cognitiva. No entanto, aproveitamos a idéia do método de utilizar perguntas para fomentar a reflexão sobre a interação representada em MoLIC.

O Método de Inspeção Semiótica tem mais afinidade com o nosso trabalho, pois visa reconstruir a metamensagem do designer e identificar problemas nessa metamensagem. A diferença aqui é que, em vez de inspecionarmos a documentação para o usuário os signos de interface e de interação de um sistema em funcionamento, propomos inspecionar os signos de modelos de interação representados em MoLIC.

As abordagens baseadas em modelos de tarefas, GOMS e CTT, são fundamentadas em teorias cognitivas e, em virtude disso, visam encontrar problemas de usabilidade. Seu foco principal está em problemas de desempenho do usuário competente, ou seja, que já sabe o que precisa ser feito. Não faz parte

da sua proposta motivar a reflexão do designer sobre as decisões de design sendo tomadas, apenas avaliar o desempenho do produto resultante. Além disso, as abordagens de avaliação empírica sobre modelos possuem apoio de uma ferramenta computacional, que visa gerar resultados quantitativos e possivelmente genéricos. Dessa forma, apesar desses trabalhos relacionados também constituírem abordagens de avaliação sobre modelos, elas não têm muito a contribuir para este trabalho.

Não há hoje uma abordagem de avaliação analítica baseada em modelos no âmbito da Engenharia Semiótica. Acreditamos que nossa proposta de explicitar a reflexão do designer possa ser um primeiro passo na direção de uma abordagem futura de avaliação baseada na MoLIC.