2 Trabalhos Relacionados

Na tentativa de encontrar o apoio necessário para a realização do estudo de avaliação aqui proposto, esta seção apresenta alguns métodos de avaliação e instrumentos relacionados a este trabalho.

2.1 Engenharia Semiótica

Este trabalho é fundamentado na teoria da Engenharia Semiótica, que caracteriza a Interação Humano-Computador (IHC) como um processo de comunicação entre o designer e o usuário, mediada pelo sistema. Durante a etapa de análise e projeto de IHC, o designer deve procurar conhecer o usuário e a forma como ele realiza suas atividades. Isso permite ao designer definir "sua interpretação de quem é o usuário, do que ele precisa, do que ele gosta e o que prefere, e assim por diante" (de Souza, 2005 p.25).

Ao interagir com o sistema, os usuários interpretam as mensagens construídas pelos designers e respondem a elas. Como quase sempre o designer não está presente (física ou remotamente) durante a interação do usuário com o sistema, a sua comunicação com o usuário sobre como o sistema funciona e por que ele foi concebido daquele modo é feita através da interface do sistema, vista pela engenharia semiótica como o representante do designer, o seu porta-voz, chamado de preposto (*designer's deputy*). O preposto do designer é quem na verdade "conversa" com o usuário, possibilitando indiretamente a comunicação designer-usuário, também chamada de *metacomunicação*, por fornecer ao usuário a chave para a interpretação da comunicação usuário-sistema através da própria comunicação usuário-sistema mediada pela interface.

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 (*breakdown*), isto é, uma falha, um mal-entendido na comunicação usuário-sistema. É responsabilidade do designer se esforçar para prever rupturas de comunicabilidade e fornecer ao usuário meios de reestabelecer a comunicação de forma que o possibilite continuar utilizando o sistema para atingir seus objetivos.

A MoLIC é um exemplo de ferramenta epistêmica fundamentada na Engenharia Semiótica que possibilita ao designer refletir sobre sua solução de design para interação, representada como um modelo.

2.1.1 Métodos de Avaliação

A Engenharia Semiótica possui dois métodos utilizados para avaliação de sistemas computacionais, apresentados abaixo.

2.1.1.1 Método de Inspeção Semiótico – MIS

O Método de Inspeção Semiótico (MIS) (de Souza e Leitão, 2009), visa reconstruir a metamensagem do designer e identificar problemas nessa metamensagem, focando os significados expressos pelo conjunto de signos e auxiliando na antecipação dos tipos de consequências que usuários poderiam trazer ao interagirem com a aplicação. Apesar de ser baseados na Engenharia Semiótica, esse método de inspeção não é aplicável a modelos, não tendo sido aprofundado neste trabalho.

2.1.1.2 Método de Avaliação da Comunicabilidade – MAC

O Método de Avaliação da Comunicabilidade (MAC) (de Souza e Leitão, 2009), tem como objetivo principal avaliar a qualidade da comunicação do designer com o usuário (comunicabilidade), através da interface, em tempo de interação. Este é um método exploratório que leva em consideração a previsibilidade da aplicação e do contexto, sendo múltiplas as interpretações possíveis. Esta avaliação permite a identificação de rupturas na comunicação que possam ocorrer durante a interação do usuário com o sistema.

Este método pode ser aplicado em diferentes estágios do design, não apenas na avaliação formativa a qual esta dissertação busca apoiar. O MAC realiza a avaliação de um artefato computacional através da observação e é possível que a ferramenta MoLIC WOz seja usada por pesquisadores para

observar usuários reais usando a simulação da interação representada em MoLIC. Este poderia ser um estudo futuro com a ferramenta.

Neste trabalho, os tipos de falhas na comunicabilidade usadas no MAC foram utilizadas para ajudar na construção dos modelos MoLIC usados nesta dissertação, descritos na seção 4.1.

2.1.2 Modeling Language for Interaction as Conversation (MoLIC)

MoLIC é uma linguagem de modelagem baseada na Engenharia Semiótica (de Souza, 2005) que possui uma família de representações para dar suporte à reflexão do designer durante o design de IHC (Paula, Barbosa *et al.,* 2003). A linguagem é composta por quatro artefatos:

1. **Diagrama de metas** – representa os objetivos dos usuários.
2. **Esquema conceitual de signos** – define e organiza os conceitos relacionados com a interface do usuário;
3. **Diagrama de interação** – representa a interação como uma conversa entre o usuário e o designer.
4. **Descrição textual** – complementa o diagrama de interação.

A interpretação do modelo MoLIC permite ao designer refletir sobre a interação e buscar problemas de comunicabilidade. Ao usar a Engenharia Semiótica para conhecer o usuário e a forma como ele realiza suas atividades, os designers estão simultaneamente estudando, analisando e tomando decisões sobre seu próprio comportamento e estratégias de comunicação. Assim, a Engenharia Semiótica é uma teoria reflexiva, que explicitamente traz os designers para a etapa dos processos de IHC e atribui-lhes uma posição ontológica tão importante quanto a dos usuários, envolvendo-os no mesmo fenômeno comunicativo (de Souza, 2005).

Para este trabalho, o diagrama de interação é o artefato da MoLIC mais importante a ser detalhado, pois nosso objeto de estudo (a MoLIC WOz) emula a interação representada nele. Abaixo é apresentado um índice visual do diagrama de interação. Ele faz parte do material gerado por Araujo (2008) para auxiliar na avaliação, por inspeção, de um diagrama MoLIC:

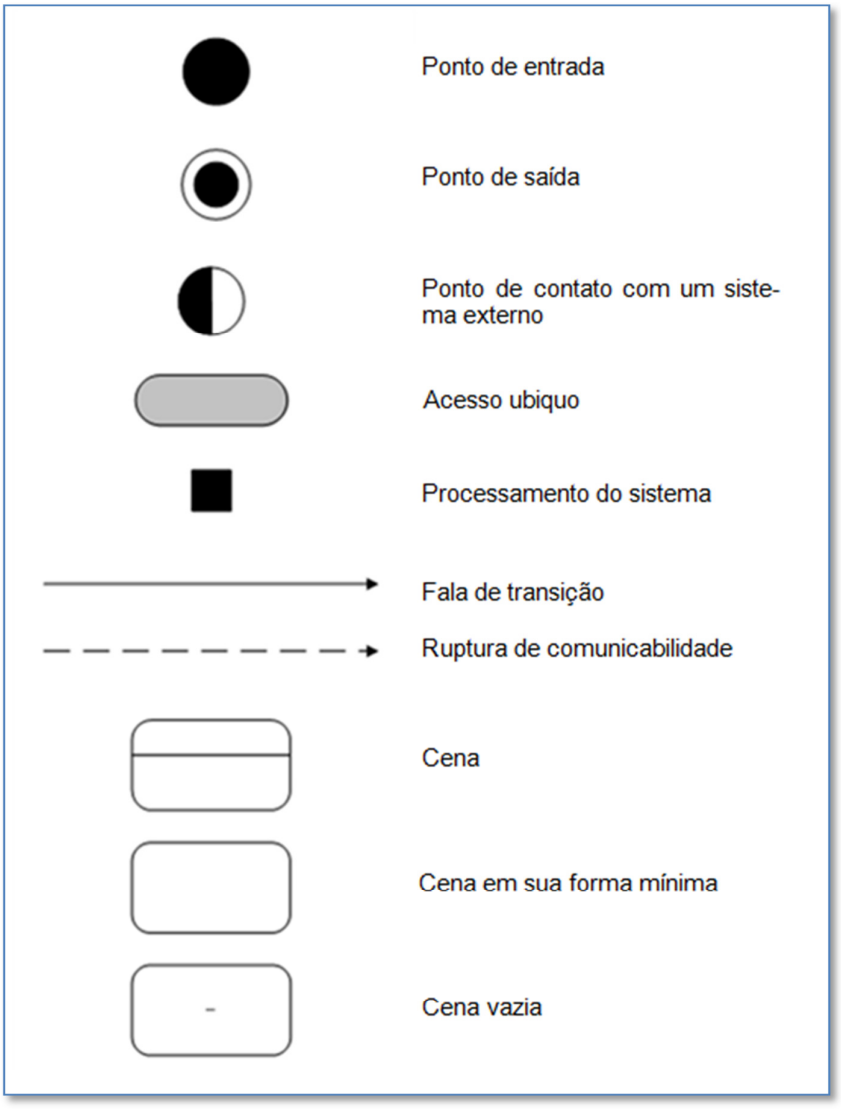

Figura 1 - Elementos do diagrama de interação da MoLIC.
Extraído e adaptado do Glossário MoLIC (Araujo, 2008)

A Figura 2 mostra um exemplo de diagrama de interação MoLIC que representa um sistema de compras.

.

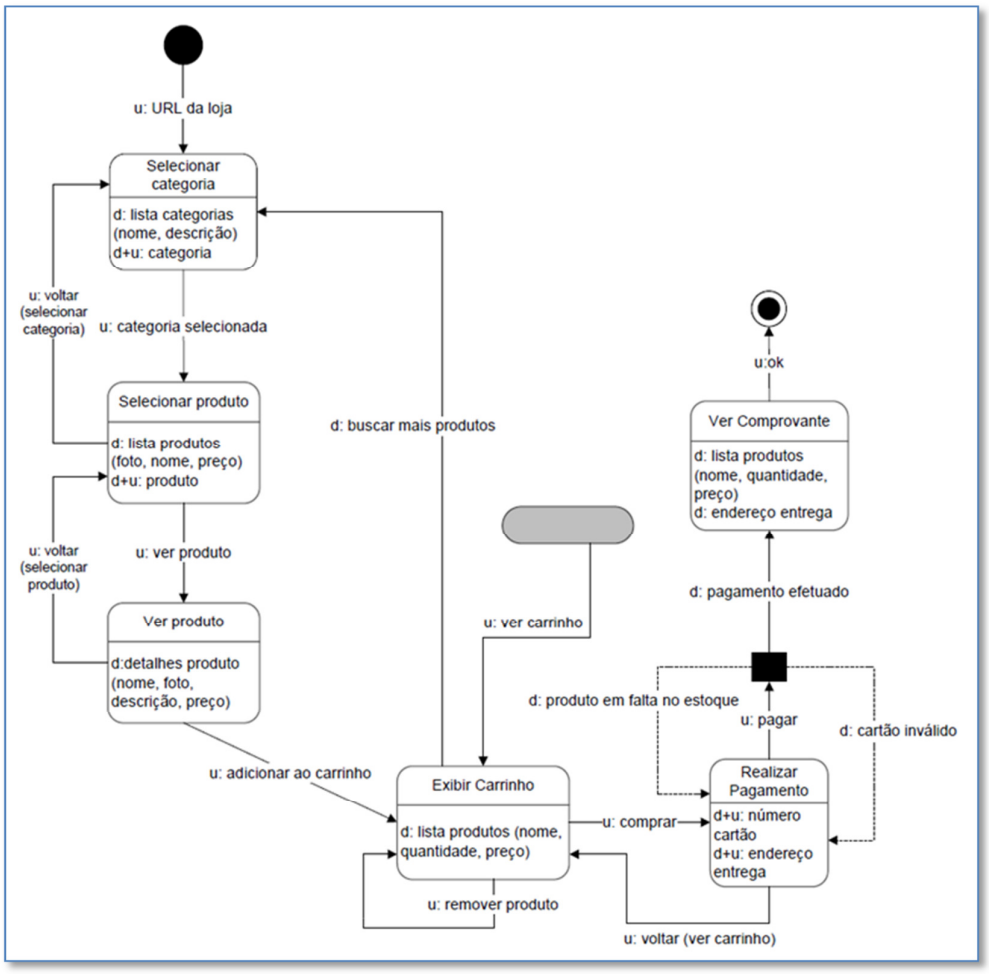

Figura 2 - Exemplo de um diagrama de interação MoLIC.

A Figura 3 apresenta com mais detalhes o trecho final do diagrama acima. Repare que os identificadores “u:” e “d:” representam as falas do usuário e do designer respectivamente.

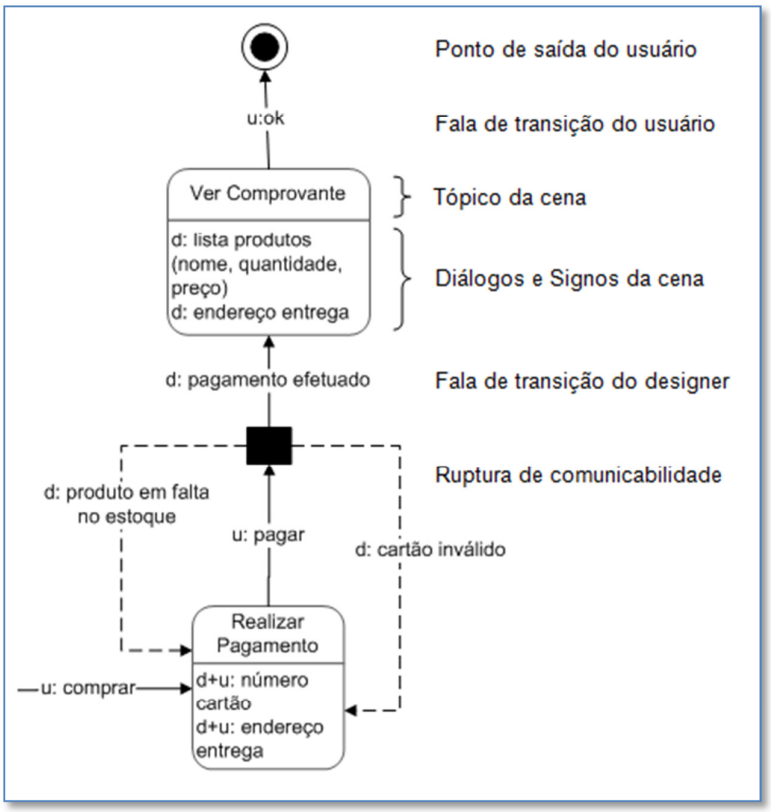

Figura 3 - Elementos do diagrama de interação MoLIC no trecho final do exemplo.

2.1.3 Inspeção de Modelos MoLIC

Visando explorar o valor epistêmico da MoLIC através do apoio a reflexão do designer, Araujo (2008) propôs um conjunto de 25 perguntas que um designer pode se fazer sobre a representação da interação, de forma a atingir dois objetivos. O primeiro é apoiar a atividade de (re)design em si, através da explicitação das consequências das decisões de design representadas na MoLIC. O segundo é apoiar a interpretação da interação humano-computador, a fim de que o próprio designer ou um outro leitor seja capaz de entender e explicar modelos MoLIC seguindo a metáfora de uma conversa entre usuário e designer.

Além do conjunto de questões, um glossário MoLIC foi preparado por Araujo. Esse glossário contém todos os termos usados na linguagem MoLIC. Para cada termo é descrita uma explicação com sua representação e ao menos um exemplo de utilização.

Araujo (2008) deixa claro que o foco de seu 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, cujo foco não é apenas descobrir problemas mas refletir sobre alternativas de design e sobre as consequências que certas decisões de design terão na interação do usuário com o sistema em questão." (p.39)

Como ainda não existe um método de inspeção específico para modelos MoLIC, o trabalho de Araujo foi a opção mais interessante encontrada para contrastar com a ferramenta MoLIC WOz, pois colabora para a interpretação dos modelos. A ideia de contrastar a ferramenta com a abordagem de Araujo foi abandonada na revisão desta dissertação por ser um primeiro estudo com a MoLIC WOz. Permitindo assim, melhorar a análise qualitativa e focar nos tipos de reflexão promovidos pela ferramenta.

Abaixo é apresentada uma das perguntas propostas por Araujo para exemplificar as reflexões e recomendações passadas ao designer.

"Que outras conversas (que não sobre as metas finais) podem ser iniciadas a qualquer momento? Com que freqüência são iniciadas, ou o quanto são importantes?

Reflexão: Acessos ubíquos também podem ser utilizados para cenas que, apesar de não iniciarem metas finais, são freqüentemente acessadas ou são muito importantes para a aplicação.

Recomendação para o designer: Analisar se estas conversas devem possuir acesso ubíquo em função de sua freqüência de acesso ou em função de sua importância para a aplicação.

Exemplo na Figura 4: Dependendo do domínio do problema e da solução dada pelo designer, buscas (Figura 4.a) e acessos para sair do sistema (Figura 4.b) podem não ser categorizados como metas finais, mas, por serem realizados freqüentemente, é comum que tenham acesso ubíquo. De forma análoga, cenas de ajuda (Figura 4.c) em geral também não são consideradas metas finais dos usuários, mas, por

serem cenas de grande importância em uma aplicação, é desejável que se ofereça um acesso rápido e direto a elas, através de acessos ubíquos.

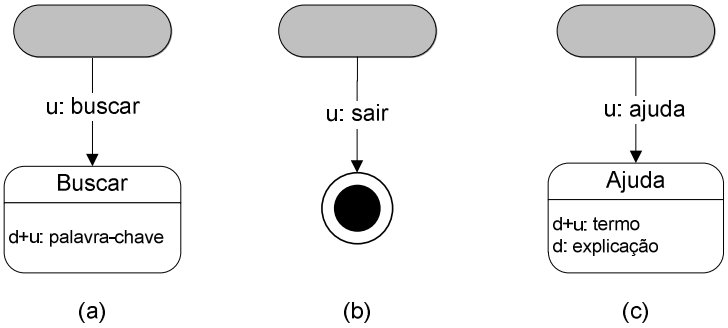

Figura 4 - Conversas que não são sobre metas finais, mas que podem ser acessadas a qualquer momento em virtude de sua freqüência de acesso, (a) busca e (b) saída do sistema, ou de sua importância para a aplicação: (c) ajuda." (Araujo, 2008)

2.2 Ferramentas de Avaliação baseadas em modelos

Nesta seção, serão apresentados alguns modelos de design não baseados na Engenharia Semiótica que possuem ferramentas de avaliação.

2.2.1 CTT – Modelo de Tarefas

O ConcurTaskTrees (CTT) é uma linguagem usada para modelos de tarefas. Ele foca o design e especificações de aplicações seguidas por roteiro que combina estruturas hierárquicas de tarefas simultâneas com um conjunto de operadores temporais. Ele representa atividades interativas através da decomposição das tarefas em uma estrutura de árvore invertida (Paternò *et al.*, 1999). Modelos de tarefas são usados para representar a interação focada nas tarefas que o usuário pode realizar com o sistema em um alto nível de abstração.

Existem duas ferramentas computacionais que apoiam o uso do CTT, o CTTE[1], *CTT Environment* e o RemUSINE (Paternò, 2000). O CTTE permite verificar se o modelo está de acordo com as regras sintáticas de construção de modelos de tarefas seguindo a notação do CTT e o RemUSINE é uma ferramenta específica para a avaliação do modelo de tarefa seguindo a notação CTT. Essa é uma solução que permite aos designers avaliar remotamente a utilidade dos aplicativos através de ferramentas computacionais, dados empíricos e modelos de tarefa. A ferramenta faz uso de modelos de tarefas não prescritivas e flexíveis no CTT. A Figura 5 apresenta um exemplo de log de eventos do REMUsine onde, para cada evento, a ferramenta indica que tarefa está associada a ele, que outras tarefas estavam disponíveis quando ele ocorreu, se a tarefa associada possuía pré-condições verificadas quando o evento ocorreu e, caso as precondições não tenham sido satisfeitas, que tarefas precisavam ser realizadas antes para satisfazer a condição.

1 http://giove.cnuce.cnr.it/ctte.html (último acesso em fevereiro de 2013)

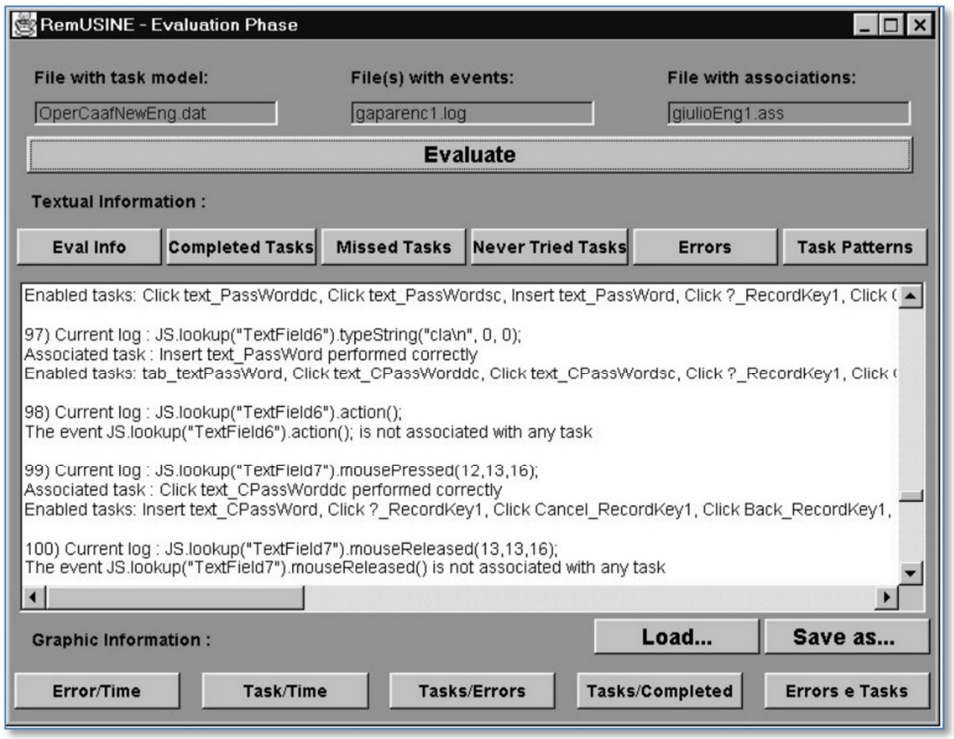

Figura 5 - Exemplo de análise interativa de um log de eventos do REMUsine.

O RemUSINE permite realizar avaliação empírica baseada no modelo CTT por contrastar um modelo de tarefas com um registro (*log*) das ações realizadas pelo usuário. A MoLIC WOz também gera um registro (*log*) das ações (falas de transição) realizadas pelo usuário, mas as falas e rupturas do designer (*Wizard*) também são registradas. Diferente do RemUSINE, não existe uma análise padrão sobre os resultados da interação com a MoLIC WOz, a avaliação pode ser feita pelo usuário da interação simulada, pelo observador, pelo designer ou apenas pelo histórico da interação. O usuário não interage com o sistema avaliado mas sim com uma simulação do sistema representado em MoLIC.

2.2.2 Modelos da Família GOMS

O modelo KLM (Keystroke-Level Model, Card *et al.*, 1980) ou KLM – GOMS (Goals, Operators, Methods, and Selection rules) é um dos modelos da família GOMS (Card *et al.*, 1983) mais simples e fácil de ser usado. Ele utiliza a teoria psicológica da cognição humana e capacidades motoras para transformar os passos necessários para cumprir uma tarefa simples em uma sequência de ações elementares (físicas ou cognitivas) cujos tempos médios são bem definidos. O KLM possui uma ferramenta chamada CogTool (John *et al.*, 2004) capaz de automatizar o modelo sem exigir que o usuário tenha conhecimentos avançados sobre os fundamentos teóricos da cognição humana e capacidades motoras. O objetivo desse software é melhorar a interface de um sistema qualquer diminuindo o tempo que o usuário do sistema leva para completar tarefas simples.

Um exemplo de uso do CogTool pode ser visto na Figura 6, onde existe uma sequência de 11 ações elementares produzidas pela ferramenta (à direita da imagem) ao testarmos a interface de um sistema móvel (à esquerda da imagem) para a tarefa de buscar em um guia de Nova York informações sobre o Museu de Arte Metropolitana (MET). Na imagem também podemos constatar a previsão de que tal tarefa levaria 13,4 segundos para um usuário experiente. Para tanto foi preciso criar 5 quadros percorridos por 4 passos simples:

1. Acessar a lista de museus do guia, pressionando o botão “Museums” na tela principal do guia.
2. Apertar a tecla “M” do teclado virtual, a fim de filtrar os museus que começam por M.
3. Apertar a tecla “E” do teclado virtual, a fim de filtrar os museus que começam por ME.
4. Acessar as informações do MET, pressionando o item “Metropolitan Museum of Art”

Uma vez modelado os 5 quadros e as 4 transições no CogTool, este foi capaz de gerar as ações elementares para concluir a tarefa e calcular o tempo que ela levaria.

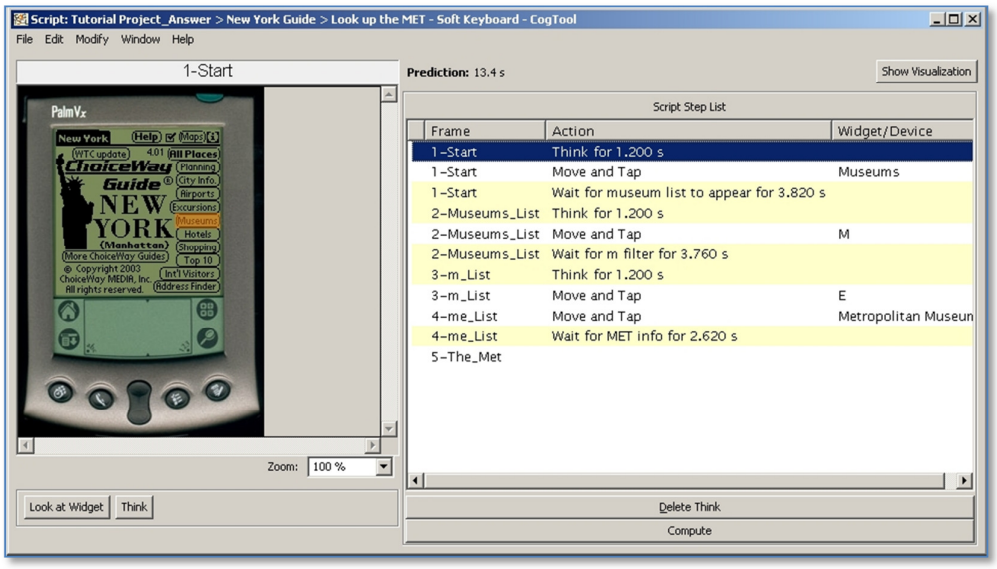

Figura 6 - Ferramenta CogTool fazendo uma avaliação baseada no modelo KLM – GOMS

O modelo KLM-GOMS é capaz de avaliar apenas tarefas simples que durem menos de 5 minutos para serem completadas. Para tarefas mais demoradas é preciso usar modelos mais complexos do que o KLM. O CogTool é mais uma ferramenta de avaliação empírica baseada em modelo, seu objetivo é fundamentalmente diferente da MoLIC WOz, cuja proposta não é emular e avaliar a interface de um sistema, mas avaliar a comunicabilidade.

2.2.3 Dimensões Cognitivas de Notações

O *framework* de *Dimensões Cognitivas de Notações* (CDN, do inglês *Cognitive Dimensions of Notations*) (Blackwell, 2006; Blackwell e Green, 2003; Green, 1989) é uma ferramenta analítica comumente usada para avaliar a usabilidade da interface de um software com seu usuário e linguagens computacionais, ela consiste em um conjunto de critérios ou dimensões que influenciam a facilidade de um ser humano em manipular artefatos descritos em uma notação arbitrária.

A definição de notação do CDN é bastante genérica, pois considera qualquer sistema com que um ser humano pode interagir, fornecendo e obtendo informações relativas a um artefato sendo manipulado. Em geral, as avaliações de software usando o CDN consideram a interface do software com seu usuário como parte da notação sendo avaliada e o resultado da utilização do software como o artefato sendo produzido.

Um conceito fundamental quando se trata do CDN é a noção de *trade-offs*, em outras palavras, situações conflitantes onde as decisões promovem uma mudança de paradigma onde se sacrificam alguns benefícios por outros. No caso, todos os *trade-offs* do CDN envolvem trocar benefícios de certas dimensões cognitivas por outras sempre e quando estas sejam melhores.

Blackwell e Green (2003) definem catorze dimensões cognitivas que analizam constantemente todos as dimensões que possam vir a se tornar novas dimensões cognitivas do framework. São elas:

1. Viscosidade (resistência a mudanças)
2. Visibilidade (habilidade de ver componentes facilmente)
3. Compromisso prematuro (regras na ordem de ações devem ser conhecidas previamente)
4. Dependências ocultas (relações entre entidades não podem ser vistas facilmente)
5. Expressividade do “papel” (o objetivo da entidade é diretamente inferido)
6. Propensão ao erro (notação instiga o erro)
7. Abstração (tipos e disponibilidade de mecanismos de abstração)
8. Notação secundária (informação adicional apresentada de forma além da sintaxe formal)

9. Proximidade de mapeamento (representação mais próxima ao domínio desejado)
10. Consistência (semântica similar expressada ao longo de todo o produto)
11. Difusão (verbosidade da linguagem)
12. Operações mentais difíceis (exigência maior dos recursos cognitivos)
13. Provisoriedade (grau de compromisso para ações ou marcações)
14. Avaliação progressiva (trabalho em dia pode ser verificado a qualquer momento)

O objetivo principal do CDN é definir um vocabulário ou um conjunto básico de questões de usabilidade que possam ser utilizados por designers de software, sem a necessidade de treinamento prolongado ou conhecimento multi-disciplinar (Blackwell *et al.*, 2001). Desde então, o CDN já foi utilizado na avaliação de linguagens de programação (Clarke, 2001), ambientes de programação visual (Green e Petre, 1996), APIs (Clarke, 2006; Afonso *et al.*, 2012), Middleware (Maia *et al.*, 2012) etc.

O framework CDN é uma ferramenta de avaliação da usabilidade muito flexível, trata-se de princípios de design para notações, interface do software com seu usuário e linguagens computacionais. A avaliação realizada com a MoLIC WOz não trabalha com o conceito de usabilidade, mas sim com a comunicabilidade, e se restringe a avaliação de artefatos computacionais representados em um diagrama de interação.

2.3 Técnicas de Observação

Nesta seção, serão descritas duas técnicas usadas neste trabalho: o *Think-aloud* para coleta de dados e a *Wizard of Oz* por viabilizar a ferramenta MoLIC WOz.

2.3.1.1 Think-aloud

Think-aloud é um protocolo que envolve participantes pensando alto à medida que realizam um conjunto específico de tarefas. Os participantes são convidados a dizer o que estão olhando, pensando, fazendo e sentindo enquanto cuidam de sua tarefa. Isso permite aos observadores verem em primeira-mão o processo de conclusão das tarefas (Newell e Simon, 1972). O *think-aloud* é um método que, a princípio, não interfere muito no processo de pensamento. Por esse motivo ele foi usado neste trabalho para captar as reflexões levantadas pelos participantes a partir da análise do modelo de interação e da interação simulada.

2.3.1.2 Wizard of Oz

Wizard of Oz (originalmente OZ Paradigm; Kelley, 1984) é um método rápido de construção de protótipos para sistemas com alto custo de construção ou que requerem novas tecnologias (Wilson e Rosenberg, 1988; Landauer, 1987). Neste método, um ser humano (o *Wizard*) simula o comportamento do sistema, podendo ser usado para avaliar a interação de qualquer sistema em seus estágios iniciais de desenvolvimento (Hil e Miller, 1988).

Um exemplo de ferramenta que utiliza o método *Wizard of Oz* (além da MoLIC WOz) é o SketchWizard (*Davis et al*., 2007), uma ferramenta que permite aos designers criarem protótipos *Wizard of Oz* baseados em caneta (*Ink*) de interface do usuário nos estágios iniciais do design. No SketchWizard, designers e usuários finais compartilham uma tela de desenho entre dois computadores, permitindo que os designers simulem o comportamento do reconhecimento *do que foi desenhado ou escrito pelos usuários*. A Figura 7 demonstra o módulo do *Wizard* simulando o comportamento de um sistema capaz de reconhecer formas geométricas e alinhar o desenho. Já a Figura 8 demonstra o *Wizard* simulando o

reconhecimento de um texto escrito "à mão" e retornando para o usuário o texto escrito em caracteres ASCII.

Esse é um exemplo de ferramenta baseada no *Wizard of Oz*. Essa técnica é usada pela MoLIC WOz, o simulador de interação avaliado no presente trabalho.

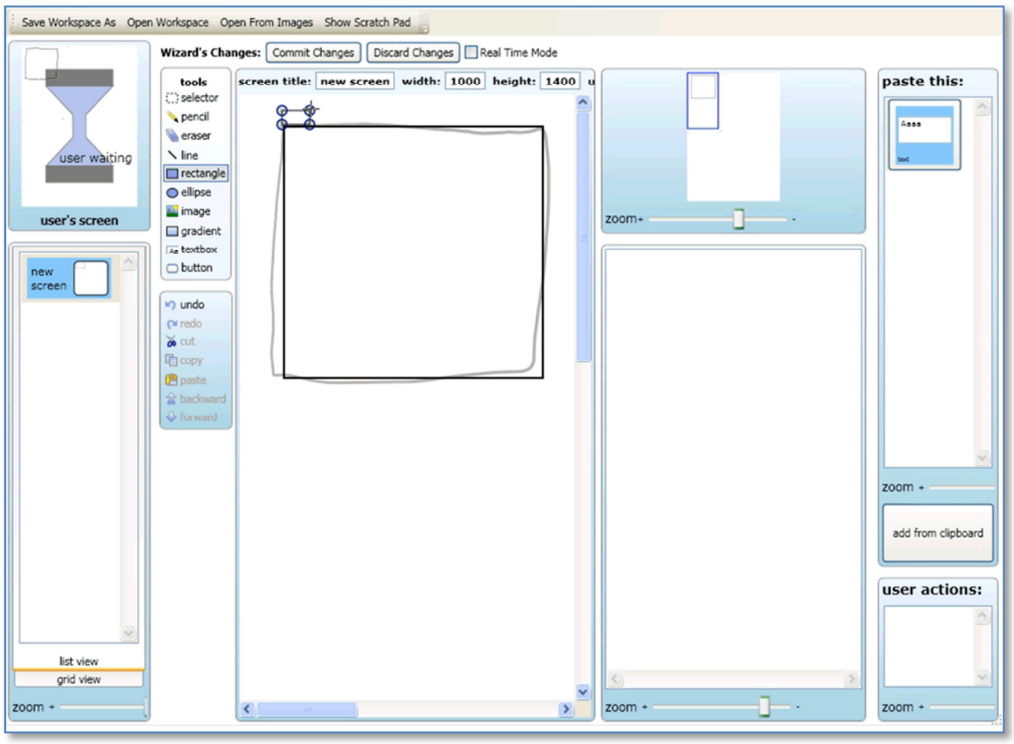

Figura 7 - SketchWizard - *Wizard* simulando a identificação de uma forma geométrica e retornando ela alinhada para o usuário.

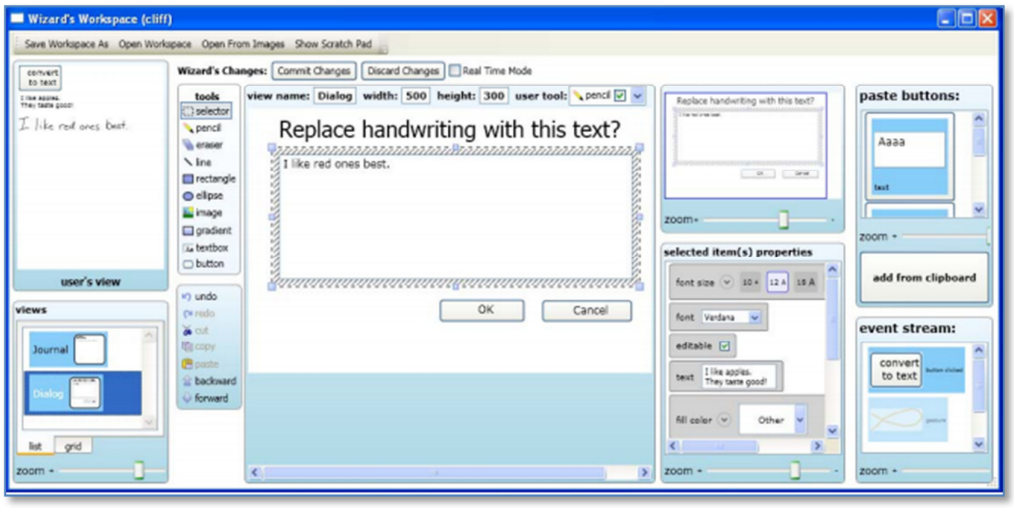

Figura 8 - SketchWizard - *Wizard* simulando o reconhecimento de um texto escrito "à mão" (usando Ink) e retornando o mesmo como texto comum (ASCII).