

(Re)construção de Interfaces de Usuário a partir da Lógica de Construção de Sistemas, Modelos Conceituais e Processo de Reflexão Humana

Elizabeth Sucupira Furtado

David Falcão Barbosa

Kênia G. de Oliveira

Alécio Carvalho

Universidade de Fortaleza – UNIFOR

Mestrado em Informática Aplicada

elizabet@unifor.br

davidfba@yahoo.com.br

kenniago@yahoo.com.br

aleciocarvalho@yahoo.com.br

Resumo

Dentre os fatores considerados durante a avaliação da qualidade de um Sistema Interativo (SI), destaca-se a usabilidade, que diz respeito à capacidade do SI de ser fácil de usar e aprender. Durante esta avaliação, problemas de usabilidade são geralmente encontrados. Este artigo descreve uma estratégia de reconstrução (*redesign*) de interfaces de usuário, que integra três áreas: Educação, Engenharia de Software (ES) e Interação Humano-Computador (IHC). Mais especificamente, são considerados estudos sobre o processo de reflexão humana, a forma lógica de realizar especificações funcionais a partir de objetos de saída (como: campos de relatórios, objetos de interação de interfaces de usuário) e os modelos conceituais usados para expressarem a interatividade do usuário com um SI.

Palavras chaves

Qualidade de Software na WEB, Avaliação da Usabilidade, Processo de (re)construção de Interfaces de Usuário.

Abstract

Among the factors considered during the quality evaluation of an Interactive System (IS), we focus on IS's usability, which is related to its ability of being easy to use and learn. During this evaluation, many usability problems can be found. This article describes a strategy of user interfaces redesign, which integrates three areas: Education, Software Engineering and Human-Computer Interaction. Particularly, we take into account studies about reflection human process, different forms to do functional specifications from output objects (as: data of reports, interaction objects of user interfaces), and the conceptual models used to express the interactivity of a user with the IS, which he/she interacts with.

Keywords

Software Quality, Usability Evaluation; Redesign Process of User Interfaces

1. Introdução

Relevantes trabalhos na área de ES têm sido publicados destacando a importância de se avaliar o processo de desenvolvimento de um software e o produto (próprio software) [Rocha et al, 2001]. Quando se aplicam métricas para medir a qualidade do processo, procura-se, dentre outras coisas, não ultrapassar orçamentos e prazos definidos [Boggs, 1999]. Quando se

aplicam métricas para medir a qualidade do produto, procura-se, principalmente, satisfazer o cliente e/ou usuário do sistema construído, verificando se o produto está desempenhando as funcionalidades exatamente como ele(s) havia(m) solicitado(s) [Belchior et al, 2001]. Existem diversas formas (não exclusivas) de se realizar esta avaliação durante todo o desenvolvimento do produto (tais como: aplicando protótipos, usando cenários, fazendo simulações com base de testes, dentre outros).

Apesar de se realizar este trabalho proativo (a fim de evitar futuros erros), existem situações em que se tem que realizar uma avaliação reativa do sistema já construído. Fatores que comprometem o desempenho da aplicação, como o alto tempo de resposta para realizar uma operação, a dificuldade de acesso às informações ou a execução errada de uma operação são frequentemente encontrados como causa de insatisfação do usuário. De forma geral, a maioria destes fatores está relacionada a problemas de usabilidade do sistema interativo. Um problema de usabilidade pode ser definido como algo que interfere na habilidade do usuário para realizar suas tarefas de forma eficaz e eficiente [Karoulis e Pombortisis, 2003].

Métodos de inspeção da usabilidade consistem em fornecer diretrizes para auxiliarem avaliadores a inspecionarem ou examinarem aspectos de uma interface de usuário. Geralmente, durante uma sessão de avaliação, o avaliador percorre a interface diversas vezes inspecionando os componentes de interface e comparando-os com uma lista de princípios de usabilidade pré-estabelecidos, por exemplo [Silva e Fontenelle, 2003]. O resultado da avaliação é uma lista de problemas de usabilidade da interface. Desta forma, percebe-se que não existe preocupação em corrigir os problemas, propondo um processo de reconstrução da interface. Vale salientar que, este processo deve ser baseado na reconstrução conjunta da interface e da interação. Com relação à interface, soluções abrangeriam aspectos visuais, como tamanho de títulos, identificação, fontes e estilos usados, cores, dentre outros. Com relação à interação, soluções abrangeriam aspectos relativos à estrutura da interface, como: relacionamentos entre os objetos de interação, pertinência de um certo objeto de interação, organização dos objetos de interação na interface, etc.

O objetivo deste artigo será apresentar uma estratégia de reconstrução conjunta da interface de usuário e da interação, baseada num processo de reconstrução da interface. Este processo envolve a reflexão sobre os objetos de interação, identificados usando lógica de construção de sistemas, e características conceituais da interface.

Este artigo está organizado da seguinte maneira: na seção 2, estão descritos os estudos realizados sobre conceitos interdisciplinares envolvendo as áreas citadas acima. Na seção seguinte, está descrita a solução que motivou a integração destas áreas para a reconstrução da interface de usuário e da interação em um caso de estudo de avaliação da qualidade de um software na WEB.

2. Conceituação Interdisciplinar

Inicialmente, serão descritos os principais conceitos relacionados ao pensamento reflexivo. Em seguida, serão apresentados a lógica de construção de sistemas usada e os conceitos de IHC considerados (modelos conceituais e recomendações ergonômicas).

2.1. Processo de reflexão

Quando nos dispomos a avaliar um objeto, geralmente realizamos um processo de reflexão deste com base em conceitos pré-estabelecidos e/ou informações que dispomos, a fim de fazemos um julgamento coerente e sermos capazes de propor melhorias. Segundo Mattos (2001) apud [Dewey, 1959], "julgar é selecionar, pesar e medir as conseqüências dos fatos e

das sugestões como nos apresentam". Para se formar um juízo adequado de uma situação, Dewey diz que é necessário passarmos por uma série de aproximações sucessivas, que constituem um processo reflexivo dialético, composto por dois movimentos: a análise e a síntese. Análise é o processo de desconstrução do real; síntese, o processo de (re)construção. A análise conduz à síntese, que remete a nova análise, em processo permanente de desconstrução/construção.

O exemplo de um processo de desconstrução pode ser o seguinte: uma interface de usuário tem muitas informações desnecessárias, causando descontentamento ao usuário. Para solucionar este problema, é necessária uma análise de cada informação, realizando um processo de desconstrução da interface.

Quanto à construção, opta-se por uma análise do processo segundo Piaget, que descreve que no processo construtivo (síntese) busca-se a adaptação das informações através da assimilação seguida da acomodação [Bordenave & Pereira, 1998; p. 28]. Na assimilação, o indivíduo incorpora os elementos desconstruídos nos esquemas cognitivos que já possui; na acomodação, ele cria um novo esquema mental para incorporar os novos conceitos ou modifica um já existente

Assim, todo juízo constitui-se de análises e sínteses sucessivas. Como formular, então, bons julgamentos? Na verdade, quanto mais o julgador possuir práticas anteriores de operações mentais sugeridas à formação de juízos, maior probabilidade terá de ser bem-sucedido em bons juízos na atualidade [Mattos, 2001]. No contexto deste trabalho, o bom julgamento de uma interface de usuário irá depender da capacidade do julgador para desconstruí-la, usando os conceitos da lógica de construção de sistemas, e de reconstruí-la, usando os conceitos estudados em IHC.

2.2. Lógica de Construção de Sistemas

Foram nos trabalhos de Warnier (1984), que a Engenharia de Software, mais especificamente, o desenvolvimento de sistemas, utilizou conceitos da lógica matemática para facilitar a organização dos dados de um sistema. A idéia básica era descobrir os dados provenientes da empresa e em seguida, elaborar um organograma arborescente, representando uma cascata de dados hierarquizados. Durante esta elaboração, fazia-se uma analogia entre a lógica matemática dos conjuntos e os relacionamentos entre os dados. De forma que, para se chegar à definição de um conjunto era preciso executar uma subdivisão arborescente de dados a tratar. A sub-divisão dos dados em níveis permitia impossibilitar a correspondência entre conjuntos de dados que não obedecessem à estrutura definida.

Esta estratégia ajudou bastante os programadores na construção de programas para emitirem relatórios, por exemplo. A partir dos campos de saída disponíveis num relatório, é possível obter a base lógica de um programa. Para os campos de saída, observa-se o número de vezes em que eles aparecem no relatório, para se determinar os conjuntos que eles pertencem. Considere, por exemplo, um relatório de emissão de pedidos por cliente que conterà, para cada cliente, todos seus pedidos efetuados. A estrutura hierárquica de organização do programa será composta dos seguintes níveis: nível 1 (topo): conterà informações/tratamentos genéricas sobre o relatório, nível 2: conterà informações/tratamentos sobre cada cliente (*c* vezes), nível 3: conterà informações sobre cada pedido (*p* vezes).

Hoje em dia, esta abordagem não é mais usada, porque ela apóia a programação funcional, e muitas modelagens são orientadas a objetos. No entanto, como se quer mostrar neste artigo, sua filosofia de organização dos dados pela frequência dos mesmos é útil para apoiar a engenharia reversa.

2.3. Modelos Conceituais

Durante o processo de construção de interfaces de usuário de um SI, o projetista, apoiado por métodos [Cardoso, 2003] e ferramentas [Vanderdonckt, 1995], é capaz de definir especificações (tais como: necessidades do usuário, suas tarefas e contexto de trabalho) gerando modelos que expressem as funcionalidades do sistema e possam ajudar a deduzir interfaces mais usáveis. Dentre estes modelos, podemos citar: modelos de tarefas e modelos de interação.

O modelo de tarefa serve para descrever as tarefas efetuadas pelo usuário e/ou SI. Ele tem por objetivo facilitar a compreensão de um domínio de aplicação dando uma noção geral das possíveis interações que podem existir, projetar novas aplicações de forma consistente com o modelo conceitual dos usuários e analisar e avaliar a usabilidade de sistemas interativos [Paternò, 2000]. Dentre estes modelos, existem o MAD* (*Modèle Analytique de Description de Tâche*) [Hammouche, 1995] e o TAOS (*Task and Action Oriented System*) [Medeiros, 2003], baseados na planificação hierárquica, onde uma tarefa é composta em sub-tarefas, que por sua vez são compostas por ações, que se situam no nível mais baixo da hierarquia (ver figura 1).

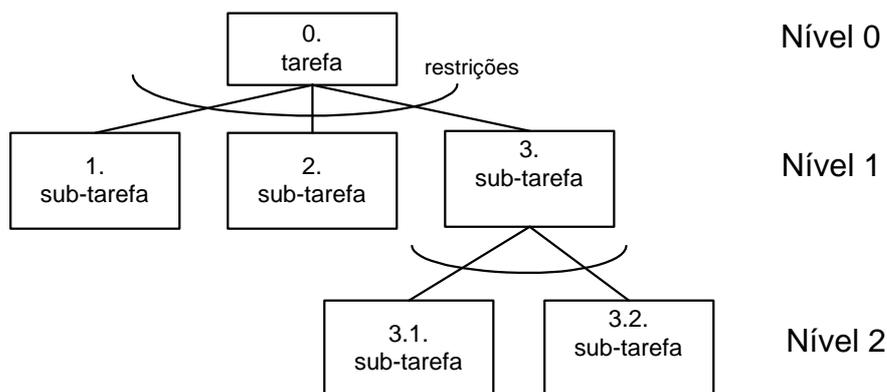


Figura 1 - Planificação Hierárquica de Tarefas

Entre as sub-tarefas de uma tarefa é importante se definir restrições estruturais, para expressarem a ocorrência com que elas são executadas (todas, algumas) e restrições temporais, para definirem a forma de execução das sub-tarefas (seqüencial, paralela, simultânea).

Além disto, pode se definir características das tarefas que serão relevantes no processo de geração da interface de usuário, tais como: se a tarefa é disparada pelo operador ou automática, se a tarefa pede ou mostra informações na tela, entre outras.

Quanto aos modelos de Interação, Furtado (1997) e Barbosa et al. (2002) descrevem modelos para representarem conceitualmente interfaces de usuário: nenhuma representação sobre como a informação será de fato apresentada em uma tela ou outro dispositivo é incluída ainda no modelo. Estes modelos servem de insumo tanto para avaliação quanto para a geração de interfaces de usuário. Do ponto de vista gerativo, deve-se seguir uma arquitetura adequada. Um exemplo de arquitetura é a utilizada na abordagem de desenvolvimento baseada em camadas de componentes, que visa principalmente a uma separação entre informações de cunho interativo (envolvendo interfaces de usuário que formatam os dados requisitados pelo usuário) e informações de cunho não-interativo (envolvendo as camadas de lógica e de

dados). Durante os processos de especificação e geração de interfaces de usuário, é necessário ainda aplicar princípios de projeto encontrados na literatura, como os princípios ergonômicos, descritos a seguir.

2.4. Recomendações Ergonômicas

Recomendações ergonômicas, também conhecidas como heurísticas de usabilidade e princípios ergonômicos, são sugestões de projeto sobre aspectos de interface, que orientam no desenvolvimento de sistemas interativos. Elas permitem determinar a melhor maneira que as informações devem ser mostradas ao usuário durante sua interação com o SI (por exemplo, o fato de mostrar somente as informações mais necessárias ou de permitir ao usuário controlar o diálogo com o sistema) [Furtado, 1997].

Vanderdonck (1999) apresentou a seguinte classificação sobre recomendações:

- Regras para projeto: que compreendem um conjunto de funções e/ou especificações operacionais que determinam o projeto de uma interface de usuário em particular. São normalmente apresentadas como regras físicas, formatos de telas e janelas;
- *Guidelines*: são princípios de avaliação e/ou projeto a serem considerados no projeto de interface de usuário. Eles são normalmente incorporados em grandes documentos chamados de “conjunto de *guidelines*” ou “guias ergonômicos”. Citamos como exemplo os oito critérios ergonômicos, definidos por Christian Bastien e Dominique Scapin (1993). Este tipo de recomendação é definido geralmente por laboratórios e instituições de pesquisa;
- Padrões ou Normas: compreendem um conjunto de especificações funcionais e/ou operacionais com o intuito de padronizar o projeto de interfaces de usuário. São definidos por organizações nacionais ou internacionais que promovem a padronização, como a ABNT (Associação Brasileira de Normas Técnicas) no Brasil, a ISO (International Standards Organization) na Europa (partes 14, 16 e 17), que se referem, dentre outros, a estilos de diálogos (tais como menu, comandos, etc.) e a HFES (Human Factors and Ergonomics Society) nos Estados Unidos;
- Guias de estilo: são documentos que normalmente são produzidos por uma organização e disponibilizados comercialmente. Podemos citar como exemplos de guias de estilo: OSF/Motif; Open Look; VUE da HP; CUA da IBM, entre outros e;
- Algoritmos Ergonômicos: aparecem como componentes de software que objetivam sistematizar a implementação de um aspecto do projeto de interfaces usando geralmente regras de produção para representarem as recomendações ergonômicas.

Devido à facilidade de acesso aos critérios ergonômicos mencionados acima, percebe-se sua utilização em vários métodos de inspeção da usabilidade. Eles são os seguintes: i) Condução, refere-se aos meios disponíveis para aconselhar, orientar, informar e conduzir o usuário na interação com o computador (mensagens, alarmes, rótulos, etc.); ii) Carga de trabalho, refere-se à brevidade das apresentações (concisão), às entradas (ações mínimas) e à densidade informacional visando reduzir a carga do usuário e melhorar a eficiência do diálogo; iii) Controle explícito, define-se no caráter explícito das ações do usuário e no controle que ele tem sobre os processamentos; iv) Adaptabilidade, refere-se às possibilidades de personalização do sistema que são oferecidas ao usuário; v) Gestão de erros, diz respeito a mecanismos que permitem evitar ou reduzir a ocorrência de erros; vi) Consistência, refere-se à forma na qual as escolhas na concepção da interface são conservadas idênticas em contextos idênticos, e diferentes para contextos diferentes; vii) Significado dos códigos e denominações,

avalia se os códigos e denominações são claros e significativos para os usuários do sistema e; viii) Compatibilidade, verifica a compatibilidade do sistema com as expectativas e necessidades do usuário em sua tarefa.

3. Estratégia de (Re)Construção da Interface de Usuário

A estratégia proposta consiste na execução seqüencial das seguintes etapas (ver figura 2): i) desconstrução da interface, através da identificação dos objetos de interação; ii) elaboração do modelo de tarefa a partir da estruturação funcional associada aos objetos identificados; iii) dedução do modelo de interação a partir do modelo de tarefa e; iv) construção da nova interface.

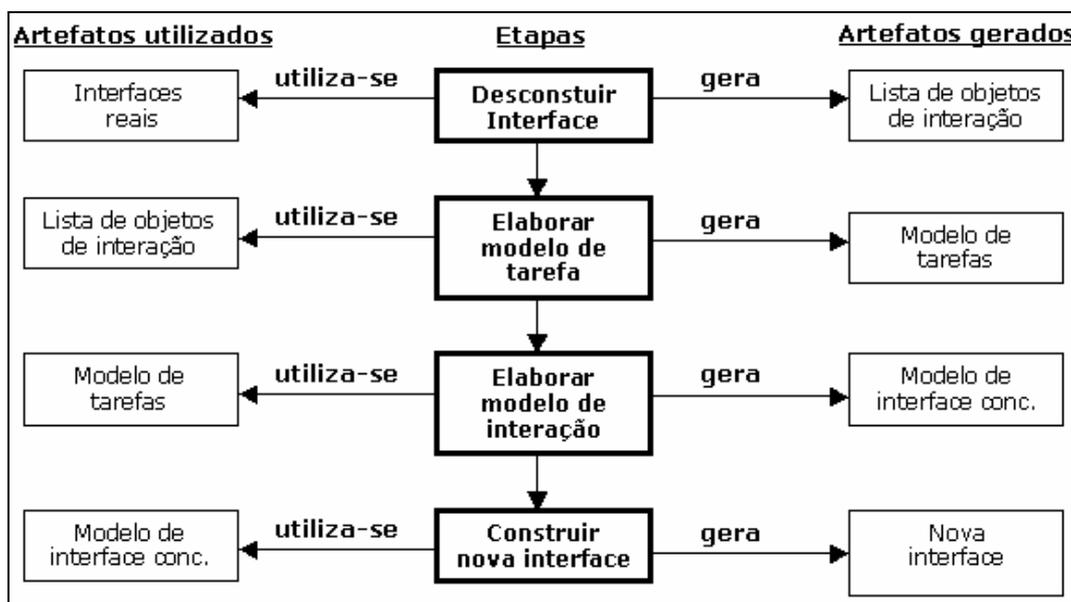


Figura 2 – Método de (re) construção da interface de usuário

Nos próximos parágrafos, segue uma descrição detalhada de cada uma destas etapas, ilustrada pelos artefatos (modelos, listas, protótipos) usados na reconstrução da interface do Yahoo (<http://my.yahoo.com/bookmarks>), mostrada na figura 3. Nesta aplicação, o usuário pode criar uma nova pasta ou mantê-la realizando as operações: adicionar, importar, exportar, apagar e editar favoritos de sua pasta.

The screenshot shows the Yahoo! Favoritos interface. At the top, there is a navigation bar with the Yahoo! logo and 'Favoritos BRASIL'. Below this, the current folder is 'Meus Favoritos' with 6 entries. A menu bar includes options like 'Adicionar Favorito', 'Nova pasta', 'Importar favoritos', and 'Exportar favoritos'. Below the menu, there are buttons for 'Apagar', 'Selecionar tudo', and 'Apagar tudo', along with a 'Mover para...' button. The main content is a table with the following columns: 'Nome', 'Comentários', 'Modificado', and 'Editar'. The table lists several favorite items, each with a checkbox, a name, a comment (e.g., 'treinamento para a certificação java'), a modification date, and an edit icon.

Nome	Comentários	Modificado	Editar
<input type="checkbox"/> Getty Images		30-Sep-02	
<input type="checkbox"/> livro de Automatos Finitos (Teoria da Computacao)		16-Mai-02	
<input type="checkbox"/> Mock Exam	treinamento para a certificação java	4-Sep-02	
<input type="checkbox"/> to Java Certification		7-Jan-03	
<input type="checkbox"/> VIDEOS de F1		20-Sep-02	
<input type="checkbox"/> wallpaper of F1		10-Jan-02	

Figura 3 – Exemplo da interface de usuário

3.1. Desconstrução da Interface

Durante a etapa de desconstrução, o projetista deve listar todos os objetos de interação, que pode ser de diversos tipos: títulos, campos de entrada e/ou saída, acionadores de tarefas, etc. Em seguida, ele deve associar a cada objeto, o elemento que identifica sua frequência de utilização, isto é a frequência com que o objeto será executado e/ou calculado. Por exemplo, na figura 4, verifica-se que o objeto *NomePasta* deve ser diferente a cada pasta, assim este objeto é associado ao elemento (1/*Pasta*), significando que *NomePasta* deve ser diferente e visualizado para cada nova *Pasta*. Neste caso, o objeto *NomePasta* é um atributo do elemento *Pasta* que identifica sua frequência de utilização.

Objeto de Interação	Frequência	Descrição
Nome Pasta	1/pasta	Indica o nome da pasta
Entradas	1/pasta	Número de favoritos da pasta
Adicionar Favoritos	2/pasta	Adicionar um novo link
Nova Pasta	2/pasta	Criar nova pasta
Importar Favoritos	2/pasta	Adicionar novas listas a partir de um arquivo especificado
Exportar Favoritos	2/pasta	Salva <i>links</i> selecionados para um arquivo
Ir para a pasta	2/pasta	Exibe árvore de pastas para selecionar nova pasta corrente
Apagar	2/pasta	Apaga Favorito Selecionado
Selecionar tudo	2/pasta	Seleciona todos os Favoritos
Apagar Tudo	2/pasta	Apaga todos os componentes sem precisar seleção prévia
Mover para	2/pasta	Move favorito selecionado para outra pasta
Nome	1/favorito	Exibe o nome do Favorito
Comentários	1/favorito	Exibe comentários sobre o favorito
Modificado	1/favorito	Exibe última data de Atualização
Editar	1/favorito	Edita todos os Favoritos

Figura 4 – Lista dos objetos de interação e frequência de utilização

Obtidos todos os objetos, o próximo passo é definir as relações que existem entre os elementos identificados. Fazendo uma associação com a orientação a objetos, significa determinar os relacionamentos estáticos, tais como composição, associação, que existem entre as classes (aqui os elementos). Na figura 4, os elementos identificados foram: pasta e favorito, interligados pelo relacionamento de composição. Relacionar estes elementos ajuda a organizar as funções e objetos de interação que a interface disponibiliza, conforme será visto na próxima etapa.

3.2. Elaboração do Modelo de Tarefa

Quando se interage com uma interface de um SI, é importante notar que existe um controle do diálogo que flutua entre o usuário e o SI, significando que o diálogo pode ser realizado ora pelo usuário (quando ele decide clicar sobre um botão) ora pelo SI (quando ele responde a uma solicitação do usuário). A definição deste controle do diálogo pode ser feita de diversas formas, dentre elas através da especificação das diversas possibilidades de interação, modelando que tarefas serão executadas pelo usuário e/ou pelo SI. Esta modelagem constitui no objetivo desta etapa e que será feita a partir do levantamento dos objetos de interação.

Para cada objeto de interação sugere-se identificar a tarefa responsável pela manipulação e/ou visualização pelo usuário. Para um campo do tipo *entrada* (que pode ser um texto editável), deve-se modelar uma tarefa responsável por viabilizar a digitação deste pelo usuário. Para um campo do tipo *saída* (que pode ser uma mensagem de erro), deve-se modelar uma tarefa responsável por viabilizar a visualização deste pelo usuário. Para um campo do tipo *acionador de tarefa* (que pode ser um botão rotulado *Editar*), deve-se modelar a tarefa (no caso *Editar*) que será executada, quando o usuário clicar sobre o botão associado. Com relação aos campos do tipo *label*, não é necessário modelar uma tarefa para mostrá-los na tela, pois no próximo passo serão aplicadas recomendações ergonômicas que não permitem a existência de campos sem *labels* associados. Tarefas relativas às regras de negócio da aplicação (tais como: calcular valor a pagar, criticar campo digitado, etc.) são também especificadas. No entanto, não precisam ser especificados os detalhes da execução da mesma. As tarefas serão especificadas segundo o modelo de tarefa MAD* e observando o relacionamento entre os elementos já identificados. Para ilustrar a modelagem, considere os elementos X, Y e Z e as três seguintes possibilidades de relacionamentos entre eles.

a) X é composto de Y, que é composto de Z. É possível fazer a hierarquia de tarefas da figura 5, onde no nível superior da hierarquia (nível zero) está a tarefa raiz. No nível um, estão as tarefas que tratam os objetos de interação associados a 1/X. No nível subsequente inferior (nível dois), estão as sub-tarefas que tratam os objetos de interação associados à 1/Y e no nível três, estão as sub-tarefas que tratam os objetos de interação associados à 1/Z..

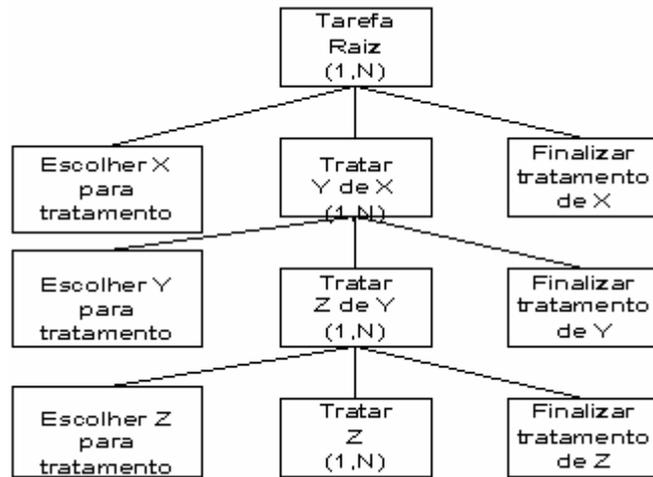


Figura 5 – Modelo de Tarefa

b) X é composto de Y e Z, onde Z não está obrigatoriamente agrupado dentro de Y. Neste caso, deve-se tratar os objetos de interação associados a 1/Y e à 1/Z no mesmo nível hierárquico.

c) Não é possível encontrar uma relação entre os elementos. Então as tarefas que tratam os objetos de interação associados a 1/X, 1/Y e 1/Z devem ser modeladas no mesmo nível hierárquico, definindo as relações estruturais e temporais entre elas.

O objetivo de modelar as tarefas respeitando os relacionamentos entre os elementos é evitar que um objeto de interação associado a um elemento X fique acessível ao usuário, quando ele tem que antes acessar um elemento mais genérico, que agrupa X, por exemplo.

A figura 6 ilustra o modelo de tarefas elaborado a partir dos objetos de interação identificados na figura 4. No nível 0, fica a tarefa raiz. No nível 1, as tarefas são relativas ao tratamento de uma pasta. Para uma pasta selecionada, são executadas as sub-tarefas da tarefa *AtualizarPasta*, que visam realizar os tratamentos sobre favoritos (nível 2). No nível seguinte, são efetuados tratamentos sobre os favoritos. Antes do tratamento, os favoritos são visualizados. Em seguida, de forma paralela, novos favoritos podem ser acrescentados à pasta ou os selecionados podem ser retirados ou editados. Na notação do modelo, a seta sobre uma tarefa simboliza uma tarefa disparada pelo usuário.

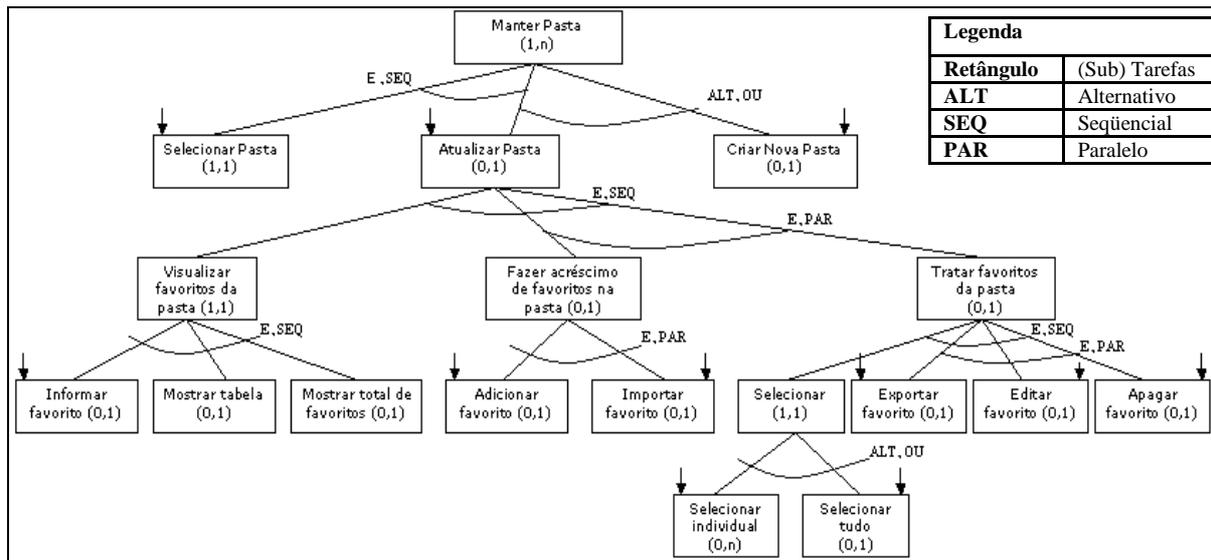


Figura 6 – Modelo de tarefa para o estudo de caso

No final da modelagem, deve-se checar se todos os objetos de interação foram tratados no modelo de tarefa. É importante também que as tarefas modeladas sejam avaliadas com o usuário da aplicação para se certificar que todas as possibilidades de interação estão de acordo com suas necessidades. Pode acontecer da aplicação já possuir limitações (como não mostrar determinadas informações), que podem ser reparadas neste momento.

3.3. Dedução do modelo de interação

Esta etapa de especificação da interface conceitual consiste em gerar o Modelo de Interface Conceitual (MIC) a partir do modelo de tarefa [Furtado, 1997]. A organização do MIC é similar àquela do modelo de tarefa, a fim de que o projetista possa perceber facilmente a maneira como as informações serão mostradas nas interfaces que vão ser construídas. O MIC é composto de dois componentes : espaço de interação e objeto de interação. Um espaço de interação é o lugar de atividade virtual que oferece ao usuário uma visão dos dados e tarefas que ele pode manipular. Um objeto de interação é o componente elementar da interface conceitual. Ele permite o diálogo com o usuário e pode representar um campo ou tarefa manipulada pelo usuário.

O processo de especificação da interface conceitual é realizado pela aplicação de regras ergonômicas, deduzidas de recomendações ergonômicas, sobre as características das tarefas. As recomendações consideradas são aquelas que procuram definir e simplificar as visões de trabalho do usuário, existentes nos seguintes critérios ergonômicos: carga de trabalho, controle explícito, adaptabilidade e gestão de erros. Durante a passagem do modelo de tarefa para o MIC, cada tarefa pode gerar um espaço ou ser associada a um espaço gerado. Esta escolha depende das características das tarefas e de seus dados. Por exemplo, existe uma regra, na qual define que cada tarefa paralela, que pode interromper a execução de outra tarefa, deve dar origem a um espaço diferente. Esta regra objetiva mostrar precisamente ao usuário o momento que uma tarefa paralela interrompe a execução de outra tarefa e/ou o momento que a execução da tarefa deve continuar. Esta regra foi baseada na seguinte recomendação: *uma janela que mostra algum tipo de alarme não deve ser coberta por nenhuma outra janela* [KOLSKI, 1993]. Quando tarefas de um mesmo nível são seqüenciais ou alternativas, elas podem ser colocadas juntas no espaço já criado por uma das tarefas deste

nível. No entanto, restrições podem ser feitas quando o número dos objetos de interação do espaço ficar muito grande ou se tratarem de dados com tipos diferentes.

Para definir os objetos de interação de um espaço, são consideradas as seguintes recomendações [Bastien e Scapin, 1993]: *dar oportunidade para o usuário tanto para realizar uma tarefa quanto para cancelá-la, tornar acessível todo dado manipulado pelo usuário durante a execução de uma tarefa, requerer do usuário disparar somente as tarefas que não são executadas automaticamente*, etc. Especificamente, para a primeira recomendação, a regra deduzida é a seguinte: *Se uma tarefa deve ser somente realizada quando o usuário a valida, então dois objetos de interação devem ser criados. O primeiro objeto serve para ativar a tarefa e o segundo para cancelar sua possível execução.*

A figura 7 ilustra o MIC deduzido do modelo de tarefa da Figura 6.

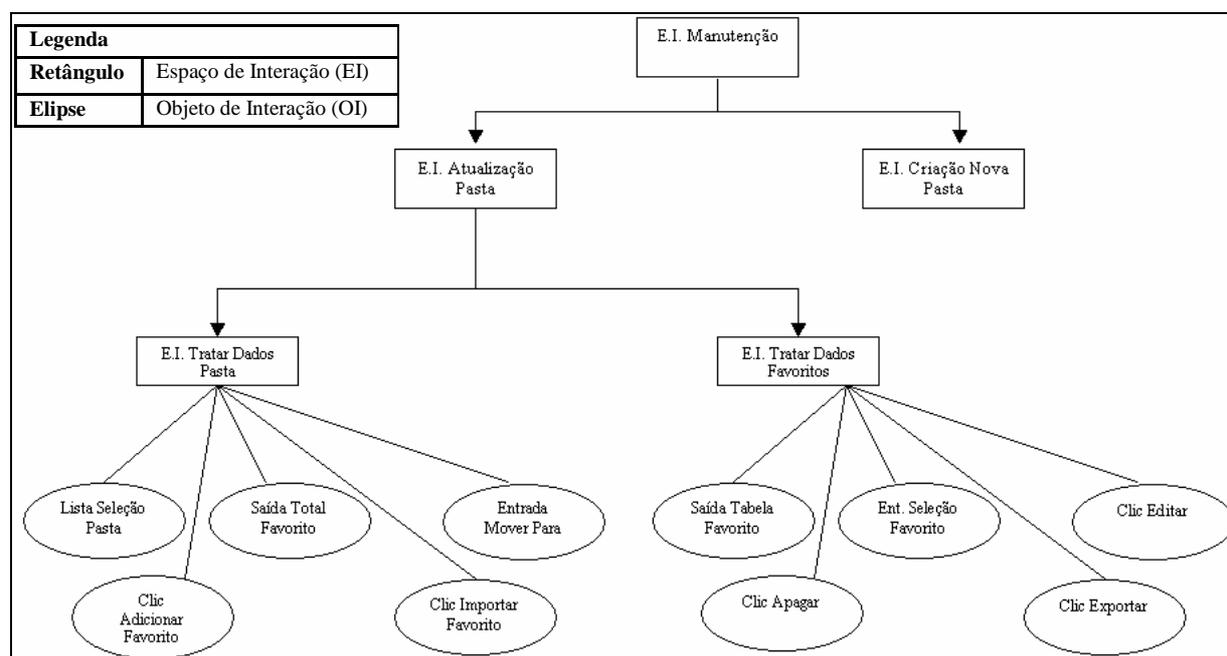


Figura 7 – MIC gerado para o MT

Para obter este modelo, as tarefas sequenciais *SelecionarPasta* e *AtualizarPasta* foram agrupadas no EI (espaço de interação) *Atualização de pasta*, ao se aplicar a regra que agrupa tarefas sequenciais num mesmo espaço, dependendo dos seus elementos associados. No nível imediatamente inferior do modelo de tarefa, não foi possível agrupar as três tarefas sequenciais, por estarem associadas a elementos diferentes. Então foram criados dois espaços de interação, para tratarem de dados de pasta e de dados de favoritos. Os objetos de interação são gerados para representarem as tarefas que devem ser disparadas pelo usuário e/ou dados manipulados.

3.4. Construção da nova interface

Durante a geração da nova interface, as recomendações utilizadas são aquelas que procuram apresentar ao usuário as diversas visões de trabalho já definidas, de forma coerente e significativa para ele. Estas recomendações são consideradas nos seguintes critérios ergonômicos: condução, consistência, significado dos códigos e denominações e compatibilidade.

Para o MIC do caso de estudo ilustrado na figura 7, a interface real associada está ilustrada na figura 8. Nela, pode-se perceber que houve algumas modificações, tais como: i) a inclusão das opções atualizar pasta, que é o EI (espaço de interação) corrente ou de criar nova pasta, navegando para um novo EI; ii) a criação de um item de seleção de pastas, não precisando haver o item: *ir para a pasta* (que havia anteriormente); iii) a supressão do campo “editar” em cada linha do favorito, já que a mesma na modelagem de tarefa, pode ficar no mesmo nível das outras opções (tais como: apagar, exportar), sendo executada após seleção de favorito(s); iv) a supressão do campo “apagar tudo”, já que ao selecionar tudo, basta somente clicar em apagar, conforme mostra o MT e; v) a organização de adicionar favorito e importar favorito no mesmo nível, já que para as mesmas não precisa ser feito seleção de favorito para serem executadas.



Figura 8 – Interface Gerada

Conclusão

Dos métodos de inspeção de usabilidade existentes, verificou-se que os processos de condução da avaliação consistem, geralmente, em um conjunto de avaliadores examinarem a interface, analisando o atendimento às heurísticas de usabilidade, por exemplo. Verificou-se também que não existe preocupação em corrigir os problemas encontrados, propondo um processo de reconstrução da interface. De acordo com experiências, percebeu-se que problemas de usabilidade, encontram-se além da própria interface real, que pode ter uma informação não legível. Eles também dizem respeito à definição das diversas possibilidades de interações, tais como: encadeamento entre as janelas, precedência de apresentação de informações, etc. Desta forma, neste artigo foi apresentado um novo processo de reconstrução conjunta da interface e da interação a partir da engenharia reversa da interface. Durante o processo de reengenharia, houve a desconstrução da interface e a construção de uma nova interface a partir de modelos conceituais e recomendações ergonômicas. Como trabalhos futuros, uma das principais questões a ser discutida é identificar os ganhos que se tem para a qualidade do sistema interativo e à usabilidade deste ao se adotar este tipo de avaliação. Outra questão importante é implementar o método, para agilizar a construção da nova interface de usuário.

Referências Bibliográficas

1. BARBOSA, S.D.J., de SOUZA, C.S., de PAULA, M.G., Silveira, M.S. **Modelo de Interação como Ponte entre o Modelo de Tarefas e a Especificação da Interface**. Proceedings of the V Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais, IHC 2002. Fortaleza, CE. 2002. pp.27–39.
2. BASTIEN, J.M.C., SCAPIN, D. **Ergonomic criteria for the evaluation of human-computer interfaces**. Tech. Rep. no 156. Rocquencourt, France: Institut National de Recherche en Inforjatique et en Automatique. 1993.
3. BELCHIOR, A. D., OLIVEIRA K., CERQUEIRA A., XEXÉO G., ROCHA A.R.C. **Definição de requisitos de qualidade de software**. Qualidade de Software. Teoria e Prática. Ed. Prentice hall. Pp 153-159. 2001.
4. BOGGS W., **Mastering UML with rational Rose**, Sybex, 1999.
5. BORNENAVE, J. D. & PEREIRA, A. M. - Estratégias de Ensino Aprendizagem; Editora Vozes, 1998.
6. CARDOSO, O., **Uma metodologia para apoiar a modelagem unificada de sistemas interativos: Uma aplicação de pesquisa operacional**. Dissertação de Mestrado. UNIFOR. 2003.
7. DEWEY, J.. **Como Pensamos**. São Paulo: Companhia Editora Nacional, 1959.
8. FURTADO, E.S. **Mise en oeuvre d'une méthode de conception d'interfaces adaptatives pour des systèmes de supervision à partir des Spécifications Conceptuelles**. Thèse de doctorat. França. Março. 1997.
9. HAMMOUCHE, H. **De la modélisation des tâches utilisateurs au prototype de l'interface Homme-machine**. Thèse de docteurs. Université Paris VI, décembre, 1995.
10. ISO 9241. **Ergonomic Requirements for Office Work with Visual Display Terminals**. Geneva, Switzerland: International Standards Organization. 1996.
11. KAROULIS, A., POMBORTISIS, A., **Heuristic Evaluation of Web-Based ODL Programs**. Usability Evaluation of Online Learning Programs, Ed. Claude Ghaoui. 2003.
12. KOLSKI, C., **Ingénierie des interfaces homme-machine**. Hermes. 1993.
13. MATTOS, F.L.C. Uma metodologia para formação continuada de professores universitários no contexto de um sistema computacional multiagentes. Dissertação de Mestrado em Educação. UFC, 2001.
14. MEDEIROS, F.P. **Projeto e Implementação do Módulo TAOS-Graph da Ferramenta ITAOS para Análise e Modelagem da Tarefa**. Dissertação de Mestrado. UFCG. 2003.
15. PATERNÒ, F. "Model-Based Design and Evaluation of Interactive Applications"; London, Springer-Verlag. 2000.ROCHA, A.R.C., Maldonado, J.C., Weber K.C., Qualidade de Software. Teoria e Prática. Ed. Prentice hall. 2001.
16. SILVA, C.R., Fontenelle, M.A. **Avaliação Ergo-pedagógica do ambiente de aprendizagem de apoio a cursos a distância: CADInet**. Encontro de pesquisa e pós-graduação. UNIFOR. 2003.
17. VANDERDONCKT, J. **Developing Milestones toward a Tool For Working With Guidelines**. Interacting with Computers 12, 2. (pp 81-118). 1999

18. VANDERDONCKT, J. **Accessing guidelines information with Sierra.** In Proceedings of Fifth International Conference on Human-Computer Interaction 1995. London: Chapman & Hall, p. 311-316. 1995.
19. WARNIER J.D. *Lógica de Construção de Sistemas.* Editora Campus. 1984.