

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS E TECNOLOGIA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

**MEDITE - UMA METODOLOGIA
ORIENTADA A MODELOS PARA CONCEPÇÃO
DE INTERFACES ERGONÔMICAS**

CLAUDIA VERONICA SEREY GUERRERO

Campina Grande - PB

Fevereiro de 2002

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS E TECNOLOGIA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

**MEDITE - UMA METODOLOGIA
ORIENTADA A MODELOS PARA CONCEPÇÃO
DE INTERFACES ERGONÔMICAS**

Dissertação submetida à Coordenação de Pós-Graduação em Informática do Centro de Ciências e Tecnologia da Universidade Federal da Paraíba, Campus II como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (MSc).

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Engenharia de Software

CLAUDIA VERONICA SEREY GUERRERO

Orientador: Bernardo Lula Jr., Dr.

Campina Grande - PB

Fevereiro de 2002

Ficha Catalográfica

GUERRERO, Claudia Veronica Serey
G934M

MEDITE – Uma Metodologia Orientada a Modelos para Concepção de Interfaces Ergonômicas

Dissertação (Mestrado) – Universidade Federal da Paraíba, Centro de Ciências e Tecnologia, Coordenação de Pós-Graduação em Informática, Campina Grande – Paraíba, Fevereiro / 2002.

79 p. Il.

Orientador: Bernardo Lula Júnior

Palavras Chaves:

1. Interface Homem-Máquina
2. Metodologias de Concepção
3. Ergonomia

CDU – 519.683

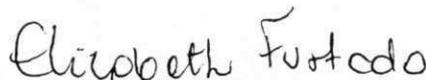
**MEDITE – UMA METODOLOGIA ORIENTADA A MODELOS PARA
CONCEPÇÃO DE INTERFACES ERGNÔMICAS**

CLAUDIA VERONICA SEREY GUERRERO

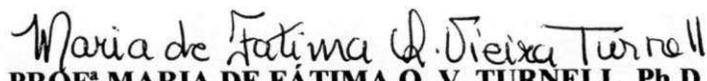
DISSERTAÇÃO APROVADA EM 22.02.2002



**PROF. BERNARDO LULA JÚNIOR, Dr.
Orientador**



**PROFª MARIA ELIZABETH S. FURTADO, Drª
Examinadora**



**PROFª MARIA DE FÁTIMA Q. V. TURNELL, Ph.D
Examinadora**

CAMPINA GRANDE – PB

*Dedico este trabalho à
minha família que tanto amo.*

AGRADECIMENTOS

Ao meu professor orientador Bernardo Lula Jr., por toda a sua dedicação, apoio, e discussões na realização deste trabalho.

À professora Fátima Turnell pela colaboração e pelos sempre bem pertinentes comentários sobre o nosso trabalho.

Aos meus pais Jorge Serey e Julia Guerrero pelo incentivo e apoio durante todo esse período e pela constante preocupação com a minha formação. Ao meu irmão pelas inúmeras discussões sobre os mais diversos assuntos.

Ao meu namorado Tarig Ali, o meu sincero agradecimento por me dar força, carinho e incentivo desde o início do mestrado.

A todos os colegas do mestrado do DSC, ao pessoal do LIA e do LABPETRI.

A todos os funcionários do DSC, a Dona Inês pelos lanches e cafezinhos.

À Aninha, Vera e à Coordenação do Curso.

A minha amiga Ivette pelo apoio desde o início e pela constante preocupação e interesse sobre o nosso trabalho.

Aos integrantes do GIHM pela paciência, comparecimento e participação nos seminários.

Aos meus amigos de Campina Grande e aqueles que estão um pouco mais longe.

À Deus pela força.

Enfim, a todos aqueles que contribuíram na realização deste trabalho, o meu muitíssimo obrigada.

SUMÁRIO

LISTA DE FIGURAS	III
LISTA DE QUADROS.....	V
RESUMO	VI
ABSTRACT	VII
1 INTRODUÇÃO.....	1
1.1 CONCEPÇÃO DE INTERFACES.....	3
1.2 METODOLOGIAS PARA CONCEPÇÃO DE INTERFACES ERGONÔMICAS	4
1.3 HIPÓTESES	5
1.4 OBJETIVOS DO TRABALHO	5
1.4.1 <i>Objetivo Principal</i>	5
1.4.2 <i>Objetivos Específicos</i>	5
1.5 METODOLOGIA DE TRABALHO	5
1.6 ESTRUTURA DO TRABALHO.....	6
2 METODOLOGIAS BASEADAS NA TAREFA.....	7
2.1 INTRODUÇÃO	7
2.2 TRIDENT.....	9
2.3 ADEPT.....	11
2.4 ERGO-START	13
2.5 ALACIE	15
2.6 MCI - MÉTODO PARA CONCEPÇÃO DE INTERFACES	17
2.7 DISCUSSÃO	19
2.8 CONCLUSÃO.....	22
3 OS MODELOS DA TAREFA, INTERAÇÃO E ARQUITETURA: MAD* E EDITOR.....	24
3.1 MODELAGEM DA TAREFA	24
3.2 O MODELO MAD*	25
3.3 MODELOS DE INTERAÇÃO E DE ARQUITETURA.....	29
3.3.1 <i>Modelo de Interação</i>	29
3.3.2 <i>Modelo de Arquitetura</i>	30
3.4 MODELO EDITOR.....	31
3.4.1 <i>Modelo unificado de interação e de arquitetura</i>	31
3.4.2 <i>Elementos do Modelo EDITOR</i>	32
3.5 CONCLUSÃO.....	35

4	BASE DE REGRAS ERGONÔMICAS.....	37
4.1	ORIGEM DAS REGRAS.....	37
4.2	ESTRUTURA E TIPOS DE REGRAS.....	39
4.3	EXEMPLOS DE REGRAS.....	39
4.3.1	<i>Regras para Construção da Árvore EDITOR.....</i>	<i>40</i>
4.3.2	<i>Regras para Definição dos Atributos.....</i>	<i>41</i>
4.4	CONCLUSÃO.....	45
5	A METODOLOGIA MEDITE.....	47
5.1	INTRODUÇÃO.....	47
5.2	EXEMPLO: UM TUTORIAL NA WEB.....	48
5.3	DESCRIÇÃO DE MEDITE.....	48
5.3.1	<i>ETAPA 1: Análise e Modelagem da Tarefa.....</i>	<i>49</i>
5.3.2	<i>ETAPA 2: Especificação conceitual inicial da Interação.....</i>	<i>51</i>
5.3.3	<i>ETAPA 3: Definição dos Atributos (Especificação conceitual total da Interação).....</i>	<i>52</i>
5.3.4	<i>ETAPA 4: Geração do protótipo.....</i>	<i>53</i>
5.3.5	<i>ETAPA 5: Avaliação.....</i>	<i>54</i>
5.4	CONCLUSÃO.....	55
6	CONCLUSÃO.....	57
6.1	DISCUSSÃO DOS RESULTADOS.....	58
6.2	PROPOSTAS DE CONTINUIDADE.....	61
	REFERÊNCIAS BIBLIOGRÁFICAS.....	63
	APÊNDICE A: MODELAGEM MAD* DA TAREFA EXEMPLO.....	68
	APÊNDICE B: MODELAGEM EDITOR DO EXEMPLO / PROTÓTIPO.....	70

LISTA DE FIGURAS

Figura 1: Abordagem baseada na tarefa [Ham95].....	9
Figura 2: Visão geral de TRIDENT	10
Figura 3: Modelos e processos de ADEPT.....	12
Figura 4: Particularização de ERGO-START na abordagem AIDE	15
Figura 5: Metodologia de concepção utilizada em ALACIE	16
Figura 6: Ciclo de concepção no Método MCI [Sou99]	18
Figura 7: Principais conceitos de MAD*	26
Figura 8: Níveis da UT	26
Figura 9: Decomposição do Sistema interativo.....	30
Figura 10: Modelo de Seeheim.....	30
Figura 11: um agente <i>Editor</i> é um agente PAC composto	33
Figura 12: um agente <i>Visão</i> é um agente PAC composto	33
Figura 13: um agente <i>Objeto_de_Interação</i> pode ser simples ou composto.....	34
Figura 14. Organização multi-agente de um <i>Editor</i>	34
Figura 15: Metodologia MEDITE	49
Figura 16: Tarefa-Raiz Consultar Tutorial na Web.....	49
Figura 17: Tarefa Estudar	50
Figura 18: Árvore EDITOR Parcial da tarefa "Consultar Tutorial na Web"	52
Figura 19: Árvore EDITOR completa da tarefa "Consultar Tutorial na Web"	53
Figura 20: Estrutura concebida de acordo com o modelo (Figura 19) gerado	53
Figura 21: Tarefa-Raiz Consultar Tutorial na Web.....	68
Figura 22: Tarefa Estudar	68
Figura 23: Tarefa Fazer Exercícios	68
Figura 24: Tarefa Ver Referências	68

Figura 25: Tarefa Visitar Sites.....	68
Figura 26: Tarefa Fazer Download.....	69
Figura 27: Tarefa Solicitar Ajuda.....	69
Figura 28: Árvore EDITOR completa da tarefa "Consultar Tutorial na Web"	70
Figura 29: Protótipo janela "Consultar Tutorial na Web"	70
Figura 30: Árvore EDITOR completa da tarefa "Estudar"	71
Figura 31: Protótipo janela "Estudar"	71
Figura 32: Árvore EDITOR completa da tarefa "Estudar Assunto"	72
Figura 33: Protótipo janela "Estudar Assunto"	72
Figura 34: Árvore EDITOR completa da tarefa "Fazer Exercícios"	73
Figura 35: Protótipo janela "Fazer Exercícios"	73
Figura 36: Árvore EDITOR completa da tarefa "Ver Referências"	74
Figura 37: Protótipo janela "Ver Referências"	74
Figura 38: Árvore EDITOR completa da tarefa "Visitar Sites"	75
Figura 39: Protótipo janela "Visitar Sites"	75
Figura 40: Árvore EDITOR completa da tarefa "Fazer Download"	76
Figura 41: Protótipo janela "Fazer Download"	76
Figura 42: Árvore EDITOR completa da tarefa "Solicitar Ajuda"	77
Figura 43: Protótipo janela "Solicitar Ajuda".....	77
Figura 44: Árvore EDITOR completa da tarefa "Solicitar Ajuda sobre Assunto"	78
Figura 45: Protótipo janela "Solicitar Ajuda sobre Assunto"	78
Figura 46: Árvore EDITOR completa da tarefa "Solicitar Ajuda sobre Site".....	79
Figura 47: Protótipo janela "Solicitar Ajuda sobre Site".....	79

LISTA DE QUADROS

Quadro 1: Resumo das Metodologias.....	19
Quadro 2: Exemplos de Regras para Construção da Árvore EDITOR do Tutorial	51
Quadro 3: Exemplos de Regras para Definição dos Atributos do Tutorial.....	52

RESUMO

Este trabalho apresenta a proposta de uma metodologia (MEDITE) para concepção de interfaces ergonômicas que utiliza modelos conceituais em cada uma das etapas do processo. Trata-se de uma metodologia que a partir do modelo da tarefa, utilizando regras formuladas a partir do conhecimento ergonômico já existente, guia a construção da especificação conceitual e arquitetural da interface segundo um modelo bem definido. MEDITE se caracteriza principalmente pela simplicidade do modelo de interação utilizado e pela objetividade das regras ergonômicas, o que permite ao projetista um rápido e fácil entendimento do método. Os modelos adotados são: MAD* para análise, descrição e modelagem das tarefas do usuário e EDITOR como modelo unificado de interação e de arquitetura.

ABSTRACT

This work presents a proposal of a model-guided methodology to design ergonomical user interface (MEDITE). MEDITE proposes the use of conceptual models in all stages of the design process. This methodology guides the conceptual and architectural specification of the interface according to a well defined model. It begins with the user task model, and applies ergonomical rules formulated from the existing ergonomical knowledge. Its main characteristics are the simplicity of the interaction model and the objectivity of the rules. These make the methodology easy to be understood and learnt. Two models have been adopted in this approach: MAD* for analysis, description and modelling of the user tasks; and EDITOR as the interaction model unified with the interface architecture model.

Capítulo 1

1 INTRODUÇÃO

A evolução das tecnologias de comunicação e informação, assim como a utilização de computadores, tem levado um número crescente de pessoas (usuários) a interagir com produtos e sistemas informatizados nas diversas áreas da atividade humana.

Computadores e softwares fazem parte de nossas vidas, saímos do tempo em que os usávamos apenas como processadores de textos. Vivemos numa época em que o computador é utilizado como ferramenta de trabalho, meio de comunicação, educação, entretenimento, comércio, etc. [Win96].

Um sistema computacional, para um usuário qualquer, é uma ferramenta e tem a finalidade de auxiliá-lo na realização de uma determinada tarefa. Porém, o uso dessa ferramenta exige do usuário conhecimento e habilidades não apenas no domínio da tarefa, mas também no uso do próprio sistema. Aprender e usar o sistema torna-se, portanto, tarefa adicional para o usuário e deve ser então minimizada.

Essa minimização exige esforço ou do usuário ou do sistema. Se o esforço exigido for feito pelo usuário, com custos que ultrapassem sua capacidade ou motivação, pode ocasionar uma degradação do desempenho da atividade ou mesmo levar ao usuário a abandoná-la. Portanto, esse esforço deve ser minimizado agindo-se sobre a "imagem" do sistema, isto é, dotando-se o sistema de uma interface de comunicação com o usuário (interface para o usuário) que leve em conta suas capacidades, habilidades e seus objetivos.

É nesse sentido que vem se exigindo cada vez mais dos desenvolvedores desses produtos e sistemas uma preocupação com a qualidade da interface. Quando se fala em qualidade naturalmente fala-se em usabilidade: fácil de usar, fácil de entender, fácil de aprender.

São três as maneiras de se obter essa qualidade:

- facilitando a construção da interface através de prototipadores rápidos, seguido de um processo consistente de avaliação;
- utilizando métodos de concepção, que permitem especificar a interface a partir do conhecimento do usuário e de sua tarefa;
- ou mesmo utilizando as duas formas acima citadas.

Em qualquer dos casos acima apontados, a utilização do conhecimento ergonômico é fundamental. Esse conhecimento pode ser encontrado nas mais diversas formas, entre elas temos: *guidelines*, diretrizes de projeto, guias de estilo, critérios, padrões, etc. Como exemplos de organização desse conhecimento podemos citar os trabalhos de Smith & Mosier [SM86], Scapin [Sca86], Bastien & Scapin [BC93], Cybis [Cyb94], ISO 9241 [ISO93], etc.

No primeiro caso, a utilização da Ergonomia está diretamente ligada ao produto. Uma *interface é ergonômica* se ela é adaptada ao trabalho efetivo do usuário. Neste sentido, ela deve permitir o máximo de eficácia durante a realização dos objetivos do usuário [Ham95]. A avaliação consiste em averiguar o respeito, ou não, à, por exemplo, critérios ergonômicos. Nesse sentido esses trabalhos têm se mostrado bastante úteis para diagnosticar falhas no produto e conseqüentemente no projeto. Este é o processo mais utilizado atualmente.

No segundo caso, a utilização da Ergonomia está ligada ao processo de especificação da interface. Fala-se, portanto, em *processo ergonômico*, que é definido como sendo aquele que integra as tarefas e as características do usuário desde o início da construção da interface, tanto em relação à utilidade quanto à usabilidade. Trata-se de um processo que oferece maiores chances de resultar em uma interface ergonômica ([Ham95] [Gam98]).

No terceiro caso, a utilização da Ergonomia ocorre tanto durante o processo quanto diretamente no produto (avaliação).

No entanto, várias são as dificuldades existentes para levar a cabo a segunda e a terceira maneira de se obter a qualidade: dificuldade por parte dos projetistas de compreender e aplicar o conhecimento ergonômico. Muitas vezes são informações difíceis de entender, de usar, pouco nítidas, estereotipadas e principalmente difíceis de aplicar ou mesmo sem ligação com o processo de concepção ([Sca88] [Cou90]). Segundo Scapin [Sca88], essas dificuldades devem-se, em parte, à falta de formalização da descrição das tarefas e do conhecimento ergonômico; além disso, a forma atual como

esse conhecimento é apresentado é muito geral não fazendo referência explícita à tarefa, obrigando o projetista a traduzir os conceitos lá emitidos em elementos de seu contexto de trabalho (modelo de tarefa, modelo de interação).

Diversas metodologias são propostas hoje na literatura para ajudar os projetistas a superarem essas dificuldades. Essas metodologias apresentam um quadro metodológico de concepção que define e sistematiza as etapas do processo integrando de diversas formas o conhecimento ergonômico.

1.1 Concepção de Interfaces

O projeto de sistemas computacionais tem sido campo praticamente exclusivo dos profissionais de informática. A concepção de sistemas prioriza as exigências da informática antes de responder àquelas relacionadas ao usuário. Os profissionais de informática (projetistas) se empenham antes de tudo em definir as funções lógicas de um sistema sem de fato se preocuparem com as necessidades, habilidades físicas, cognitivas e objetivos do usuário. Essa preocupação, em geral, não faz parte de sua formação.

A preocupação com as interfaces para o usuário é normalmente considerada apenas em etapas finais do projeto ou como uma parte separada deste. O resultado é que a interface passa a ser um “apêndice” do sistema. Contribuem sobremaneira com essa situação o fato das metodologias tradicionais de concepção de sistemas não levarem em conta o usuário, do ponto de vista de suas próprias características e do ponto de vista de seus objetivos; além da crença generalizada que uma “maquilagem” gráfica é suficiente para proporcionar ao sistema as características: “fácil de aprender” e “fácil de usar”.

Se a natureza das funções do sistema não completam as faculdades do usuário, se sua organização não corresponde à estrutura mental de resolução deste usuário, nenhum efeito de apresentação poderá esconder tais carências [Cou90]. Segundo Norman [Nor83], o foco do projeto de interfaces deve ser a tarefa que o usuário irá desempenhar com auxílio do sistema e o projeto da interface deve ser então centrado no usuário e nos seus objetivos (a tarefa a realizar).

Assim, para conceber uma interface para o usuário, o projetista deve elaborar uma descrição tão precisa quanto possível da tarefa e dos processos cognitivos¹ do

¹ O termo “processo cognitivo” é empregado para denotar as diversas etapas do caminhar da informação, desde a sua recepção, passando pelo tratamento até a ação sobre o meio [Cyb96].

usuário envolvidos na sua realização e em seguida concretizar fielmente essa descrição no sistema. Se essas descrições estiverem corretas e devidamente representadas, pode-se dizer que o sistema passa a se constituir numa “extensão do cérebro humano” [Nan90] [Cou90]).

1.2 Metodologias para Concepção de Interfaces Ergonômicas

Várias metodologias têm surgido na literatura com o objetivo de integrar o conhecimento ergonômico no processo de concepção de interfaces. Metodologias como TRIDENT ([BH94] [BH95]), ADEPT ([MPW92] [JWMP93] [WJKCM93]), ERGO-START [Ham95], ALACIE [Gam98], e MCI [Sou99] se enquadram nesse contexto. Essas metodologias fazem uso de modelos (modelo da tarefa, modelo do operador, modelo da interação, modelo arquitetural, etc.), de regras / tabelas (representação do conhecimento ergonômico) e de ferramentas computacionais para apoiar a produção das especificações em cada etapa do processo.

As metodologias baseadas na tarefa do usuário vêm se constituindo em um expressivo apoio à concepção de interfaces ergonômicas. Elas possibilitam, a partir da descrição da tarefa, do perfil do usuário e de princípios ergonômicos, a construção de uma especificação executável (protótipo) da interface levando em conta os objetivos do usuário.

No entanto, apesar do avanço que representam como suporte aos projetistas no processo de concepção de interfaces ergonômicas, algumas necessidades, dificuldades e limitações ainda persistem em alguns aspectos importantes do processo definido por essas metodologias:

- Necessidade de experiência ou conhecimento em Ergonomia;
- Dificuldades de escolha e de utilização de regras ergonômicas em relação às etapas do processo. Em geral, a taxonomia das regras não tem uma relação única com as etapas do processo;
- Dificuldades de passagem/ transformação do modelo da tarefa para um modelo conceitual da interação;
- Dificuldades na utilização e aplicação da metodologia na ausência das ferramentas;
- Dificuldades em modificar o código da interface.

1.3 Hipóteses

A observação e constatação desses problemas nos levaram a um estudo mais aprofundado das metodologias citadas. Esse estudo resultou na formulação das seguintes hipóteses de trabalho sobre as causas dos problemas e dificuldades mencionadas acima:

Hipótese 1: Ausência de um relacionamento claro entre o modelo da tarefa e o modelo de interação utilizados.

Hipótese 2: Ausência de uma descrição gráfica adequada do modelo da interação utilizado que induza a uma visualização mental da estrutura da apresentação da interface.

Hipótese 3: Ausência de um modelo explícito capaz de representar a arquitetura do código da interface.

1.4 Objetivos do Trabalho

1.4.1 Objetivo Principal

Propor uma metodologia para a concepção de interfaces ergonômicas no contexto das abordagens baseadas na tarefa, visando superar os problemas e dificuldades encontradas nas metodologias consideradas, tomando como base as causas apontadas pelas hipóteses de trabalho acima listadas.

1.4.2 Objetivos Específicos

- Definir a estrutura de uma base de regras ergonômicas;
- Criar uma base de regras ergonômicas a partir de recomendações ergonômicas oriundas de fontes diversas.
- Definir um processo de obtenção de uma especificação conceitual baseada no Modelo EDITOR [Lu192] a partir de uma especificação da tarefa baseada no Modelo MAD* [Ham95].

1.5 Metodologia de Trabalho

A metodologia utilizada no desenvolvimento deste trabalho consistiu dos seguintes passos:

1. Análise de diferentes metodologias de concepção de interface baseadas no modelo da tarefa, com o intuito de entender o processo utilizado por cada uma delas, conhecer os seus modelos e as ferramentas utilizadas
2. Identificação de problemas gerais no uso dessas metodologias
3. Identificação das possíveis causas para os problemas identificados
4. Proposição de uma nova metodologia para concepção de interfaces ergonômicas (desde os modelos, estrutura e tipos de regras ergonômicas e processos) que minimizasse os problemas apresentados pelas metodologias analisadas.
5. Aplicação da metodologia a um exemplo simples, com o objetivo de ilustrar o processo e confrontar os resultados com as hipóteses geradas.

1.6 Estrutura do Trabalho

O restante deste trabalho está organizado em 6 capítulos. O Capítulo 2 apresenta uma descrição sucinta de diferentes metodologias baseadas na tarefa (seus processos, modelos, ferramentas) e uma discussão sobre suas limitações e dificuldades de utilização.

O Capítulo 3 apresenta os dois modelos utilizados na metodologia que propomos: o Modelo MAD* [Ham95], como sendo o modelo de tarefa e o Modelo EDITOR [Lul92] adotado como modelo unificado de interação e de arquitetura.

O Capítulo 4 apresenta a estrutura geral de uma Base de Regras Ergonômicas de auxílio à concepção de interfaces para a metodologia proposta. São definidos os tipos de regras que constituem essa base e sua estrutura. Ainda nesse capítulo é apresentado como exemplo, um conjunto de regras elaboradas segundo a proposta da base.

O Capítulo 5 apresenta a metodologia (MEDITE) proposta para a concepção de interfaces baseada na tarefa com o objetivo de gerar interfaces ergonômicas e superar os problemas apresentados pelas outras metodologias estudadas e descritas no Capítulo 2 deste trabalho.

Finalmente, no Capítulo 6, são discutidos os resultados do trabalho, suas limitações e apresentadas propostas de continuidade e de trabalhos futuros.

Capítulo 2

2 METODOLOGIAS BASEADAS NA TAREFA

2.1 Introdução

A necessidade de desenvolver aplicações úteis, utilizáveis e acessíveis a um grande número de usuários ocasionou o surgimento de uma variedade de ferramentas, linguagens, modelos e metodologias de ajuda à concepção de interfaces para o usuário, tendo como principais objetivos à otimização do ciclo de desenvolvimento e a melhoria da qualidade dessas aplicações [Gam98].

No que diz respeito às metodologias de concepção, houve uma evolução dos ciclos clássicos (ex.: o modelo em cascata (de Royce citado em [Boe81])) da Engenharia de Software em direção aos ciclos iterativos com uma forte utilização de protótipos.

São várias as abordagens utilizadas, entre elas, podemos citar: a abordagem clássica, a abordagem iterativa (ex.: o modelo em espiral [Boe88]), a abordagem baseada na avaliação (ex.: o modelo Estrela [HH93]), abordagem baseada na prototipagem e abordagem baseada em modelos.

Vale ressaltar que as abordagens citadas acima podem ser combinadas dando lugar a novas abordagens, ditas híbridas. Como exemplo, temos as abordagens baseadas na avaliação que são também iterativas [Fur97].

A abordagem baseada em modelos (Szekely citado em [Gam98]) é aquela em que a concepção parte de especificações declarativas (os modelos). Estes modelos tornam explícitos os resultados de análise e de projeto, que podem ter diferentes funções, como por exemplo: representar e descrever as características do futuro sistema,

avaliar as interfaces geradas, propor a arquitetura de um sistema, etc. Esta abordagem apresenta como principais vantagens:

- a possibilidade de uma fácil automatização do processo através do uso de ferramentas específicas;
- consistência e reutilização;
- e o desenvolvimento iterativo.

Quando o processo de concepção tem como elemento base a integração da representação da tarefa do usuário, ou seja, o modelo de tarefa, fala-se da abordagem baseada na tarefa. Esta, é uma especialização da abordagem baseada em modelos, com a particularidade de que o principal modelo utilizado é o da tarefa do usuário [Gam98].

As metodologias que se baseiam no modelo da tarefa do usuário constituem um expressivo apoio à concepção de interfaces ergonômicas. Elas permitem, a partir da descrição da tarefa, do perfil do usuário e de princípios ergonômicos, a construção de uma especificação executável (protótipo) da interface levando em conta os objetivos do usuário.

Neste contexto, a análise e modelagem da tarefa não têm o objetivo de avaliar, mas de descrever uma tarefa precisamente com o intuito de entender a “lógica do usuário”. Descrever a tarefa de forma precisa e exaustiva abre espaço para a aplicação de regras ergonômicas que associem elementos obtidos na análise da tarefa a elementos da interação, ou seja, torna possível relacionar objetivos e características da tarefa às características da interface que se pretende conceber ([Sca88] [Seb95]).

As principais etapas do processo definido por essas metodologias, como ilustra a Figura (1) abaixo, são:

- análise da tarefa e das características do usuário;
- definição das representações abstratas da interação;
- definição de especificações para a implementação;
- e avaliação do protótipo da interface.

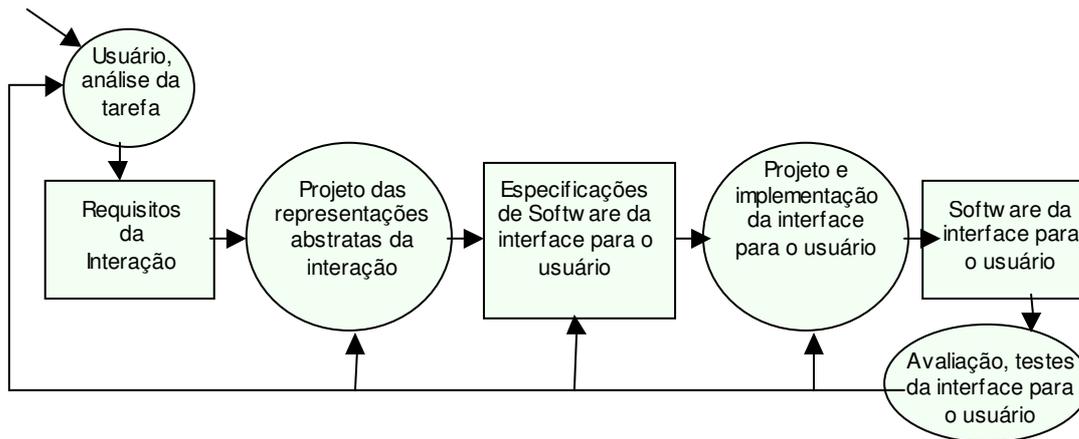


Figura 1: Abordagem baseada na tarefa [Ham95]

Entre as metodologias que se enquadram nesse contexto, citamos: TRIDENT ([BH94] [BH95]), ADEPT ([MPW92] [JWMP93] [WJKCM93]), ERGO-START [Ham95], ALACIE ([GS97] [Gam98]), e MCI [Sou99].

Para auxiliar o projetista no processo de concepção de interfaces, essas metodologias, em geral, definem e fazem uso de diferentes modelos (modelos de tarefa, modelos de interação, modelo do operador, modelo arquitetural, etc.), de regras / tabelas para representar o conhecimento ergonômico e de ferramentas computacionais para apoiar a produção das especificações em cada etapa do processo.

Apresentamos a seguir, de forma sucinta algumas dessas metodologias e são discutidos os problemas detectados nelas, confrontando-os com as hipóteses apresentadas no capítulo anterior.

2.2 TRIDENT

TRIDENT ([BH94] [BH95]) (*Tools foR an Interactive Development EnvironmeNT*) é uma metodologia e um ambiente de suporte para o desenvolvimento de aplicações altamente interativas. TRIDENT consiste de um conjunto de:

- processos que definem uma abordagem completa e contínua para o desenvolvimento das aplicações;
- produtos que resultam desses processos;
- modelos básicos, nos quais esses processos se fundamentam;
- e ferramentas interativas que dão suporte aos processos e ajudam o projetista na construção da interface para o usuário.

A Figura (2) abaixo ilustra uma visão geral da metodologia TRIDENT.

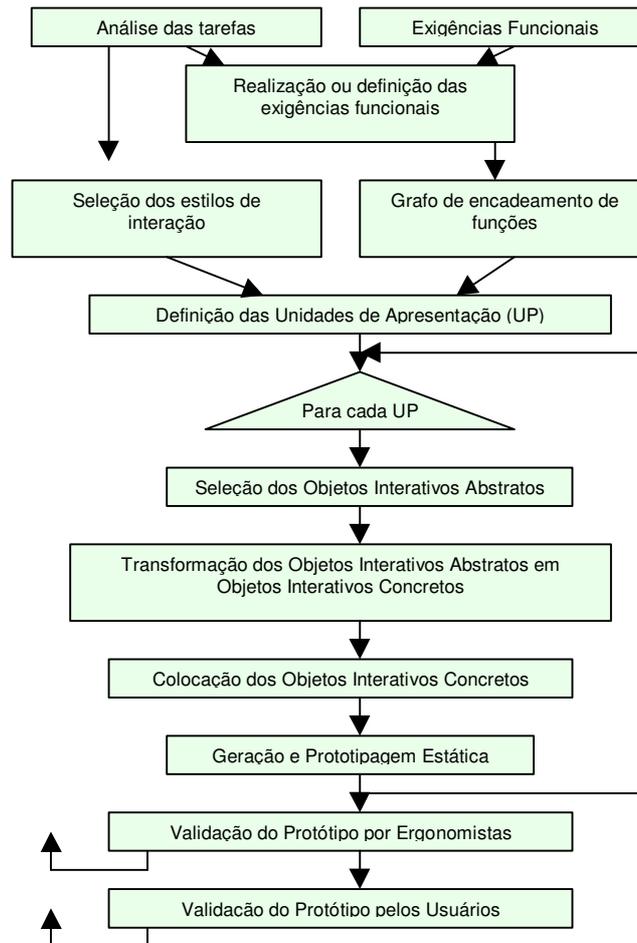


Figura 2: Visão geral de TRIDENT

TRIDENT parte da análise da tarefa (identificação de objetivos, sub-objetivos, procedimentos e objetos) e das exigências funcionais da aplicação existente para produzir o ACG (*Activity Chaining Graph*) - Grafo de Encadeamento da Atividade. O ACG descreve o fluxo da informação entre as funções do domínio da aplicação que é necessários para executar o principal objetivo na análise da tarefa e mostra a sincronização e as informações de entrada e saída de cada uma das funções.

A partir de regras ergonômicas é produzida a **apresentação**, cujo conteúdo é organizado em uma arquitetura hierárquica fortemente ligada aos elementos definidos no grafo ACG, e é composta pelos seguintes conceitos:

- *Unidade de Apresentação (UP)* (ligada às funções semânticas identificadas no ACG);
- *Janela* (identificadas por meio de regras ergonômicas de modo a respeitar um certo número de critérios ergonômicos (compatibilidade, carga de trabalho, etc.);

- *Objeto Interativo Abstrato* (OIA) (no seio de cada janela, faz-se corresponder um OIA (simples ou composto), a cada informação de entrada/saída);
- e *Objeto Interativo Concreto* (OIC) (a transformação dos OIA em OIC depende da plataforma sobre a qual se trabalha. A colocação de cada OIC é definido com o auxílio de regras ergonômicas de colocação). Os OIC gerados podem ser recuperados no editor gráfico, permitindo ao projetista retrabalhar a apresentação a fim de melhor adapta-las às necessidades do usuário.

Uma outra etapa consiste na derivação da arquitetura da aplicação interativa (modelo de arquitetura da interface). Ela especifica de maneira rigorosa as ligações entre os elementos da tarefa do usuário e aqueles da arquitetura da aplicação. Dois princípios são estabelecidos segundo esta abordagem: cada elemento de arquitetura deve ser derivado da análise da tarefa e uma autonomia deve existir entre os elementos que representam os componentes da aplicação e aqueles que representam os componentes da interface.

O modelo de aplicação (de arquitetura) é composto por três objetos: os *Objetos de Controle* (OC), os *Objetos de Aplicação* (OA) e os *Objetos de Interação* (OI). As instâncias destes três objetos assim como suas relações semânticas são definidas de maneira rigorosa.

A última etapa de TRIDENT consiste em especificar os três tipos de diálogo possíveis nas aplicações: entre elementos de uma mesma janela, entre duas janelas diferentes e entre duas unidades de apresentação diferentes.

Um protótipo da apresentação é gerado a partir das especificações das janelas e de seus OIC. Este protótipo é em seguida validado por ergonomistas e depois testado pelos usuários finais. A maioria das etapas de TRIDENT são assistidas por ferramentas de geração automática.

2.3 ADEPT

ADEPT (*Advanced Design Environment for Prototyping with Tasks*) ([MPW92] [JWMP93] [WJKCM93]) é um ambiente para concepção de interfaces baseado na tarefa e centrado no usuário. A metodologia proposta em ADEPT se desenvolve em várias

etapas de modelagem, partindo da análise da tarefa, passando pela especificação conceitual da interface até a geração do protótipo.

A Figura (3) abaixo ilustra os modelos e os processos de ADEPT.

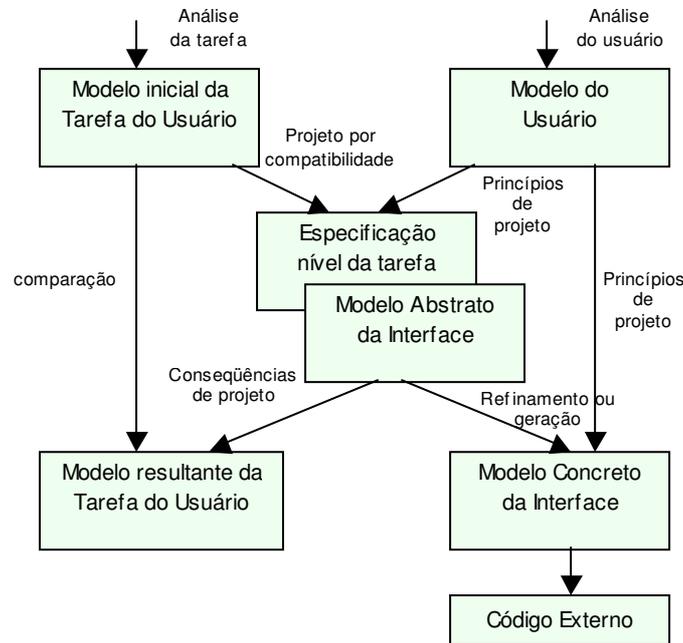


Figura 3: Modelos e processos de ADEPT

Para cada uma de suas etapas, ADEPT faz uso de diferentes modelos:

- *Uma versão modificada de TKS (Task Knowledge Structure) [JJW88] é adotada como modelo de representação do conhecimento sobre a tarefa dos usuários. Esta representação se constitui no "ponto de partida" para o projeto da interface;*
- *O Modelo do Usuário, que provê informações gerais sobre as características dos usuários. Este modelo constitui uma imagem do usuário típico modelado em termos de suas habilidades, conhecimentos e estilos de processamento da informação. Este modelo é usado para influenciar e auxiliar na seleção de alternativas de projeto;*
- *AIM (Abstract Interface Model) - Modelo Abstrato da Interface que provê um alto nível de descrição da interface (especificação conceitual da interface). Trata-se de um estágio intermediário entre a descrição da tarefa e o código do sistema. Apresenta uma estrutura simples com apenas dois tipos de componentes: Grupos e Objetos Abstratos de Interação (AIOs). Um Grupo é um conjunto de componentes podendo incluir outros Grupos,*

definindo assim uma estrutura hierárquica. O *Grupo* da raiz representa a interface inteira e os AIOs formam as folhas da árvore AIM.

- e o *CIM (Concrete Interface Model)* - Modelo Concreto de Interface trata-se do mais baixo nível conceitual da interface do usuário, com um nível mais detalhado de objetos de interação, *layout* de telas e sequenciamento.

ADEPT não apresenta um modelo de arquitetura da interface, o método é suportado computacionalmente pelo ADEPT *Demonstrator*, um ambiente prototipador de interfaces, que contém:

- editores gráficos de árvores que permitem a edição e visualização dos modelos da tarefa e a elaboração do modelo da interface;
- uma ferramenta para modelar cada grupo de usuários em termos de suas habilidades, conhecimentos e estilos de processamento da informação. Estas características são associadas à regras de projeto (conhecimento de usabilidade e *guidelines* para projeto da interface) que resultam em um conjunto de regras aplicáveis que devem ser utilizadas para tomar decisões entre diferentes alternativas de soluções para o projeto;
- e por fim, um interpretador que gera o modelo concreto da interface.

O processo de ADEPT refina elementos vindos da tarefa do usuário de forma a obter elementos de uma interface abstrata e posteriormente os objetos concretos da interface.

ADEPT centra-se na idéia de que, sem ferramentas para suporte, muitas das técnicas são complicadas e tem o seu uso limitado. As várias ferramentas do ADEPT *Demonstrator* suportam o processo de projeto ou através da manipulação de modelos ou facilitando as passagens de um estágio de modelagem a outro.

2.4 ERGO-START

ERGO-START (*methodolgiE oRientée erGonomie du lOgiciel: depuiS la description des Tâches utilisAteurs jusqu'à la Realisation d'InTerface*) [Ham95] é uma metodologia para concepção e avaliação ergonômica de interface para o usuário. Baseia-se em um conjunto de conceitos e técnicas provenientes tanto da Ergonomia (para modelagem) como da Engenharia de Software (para realização). Trata-se de um método que a partir da descrição das tarefas, produz uma especificação conceitual da interface utilizando mecanismos que se fundamentam na Ergonomia.

Particulariza-se na Abordagem AIDE (Figura 4), e define os seguintes modelos e mecanismos:

- MAD* - *Un Modèle Analytique de Description de TâcheS Utilisateurs Orienté Spécification d'Interface* (Modelo Analítico de Descrição de Tarefas do Usuário orientado à Especificação de Interface) para modelar e descrever as tarefas estruturadas de forma hierárquica. MAD* integra também aspectos que ajudam na especificação da interface, tais como: papel do operador, centralidade da tarefa, etc.;
- ICS - *Interface Conceptuelle de Spécification* (Interface Conceitual de Especificação) para modelar e especificar conceitualmente a interface. ICS especifica a interface do ponto de vista do usuário, através da representação das tarefas sob forma de grafos de estados de interação. A arquitetura ICS é composta de sete componentes classificados de acordo com sua função e reagrupados em três níveis hierárquicos, onde cada nível controla e refina o nível seguinte (assim como em CLG [Mor81]).
 - no nível tarefa é definido o componente Esquema_de_Procedimentos (SP);
 - no nível semântico são definidos os componentes: Procedimento (P), Ação (A) e Objeto_Funcional (OF);
 - e no nível perceptivo são definidos os componentes perceptuais acoplados aos componentes semânticos: Componente_Perceptual (CP), Estado_Interface (EI) e Tela (E).
- Mecanismo AIDE - caracteriza-se principalmente por integrar a Ergonomia de maneira operacional no processo de construção do sistema. São duas as fases de AIDE, constando cada uma delas de três etapas estruturadas de acordo com os níveis ICS:
 - (i) Especificação da Interface Conceitual que consiste em especificar os componentes perceptuais ICS com o auxílio de regras ergonômicas em cada uma das etapas;
 - (ii) e a Geração da Interface Física, que consiste em traduzir a estrutura lógica gerada na etapa anterior em uma estrutura física (protótipo), fazendo uso também de regras e critérios ergonômicos. As regras e critérios ergonômicos são extraídos de recomendações oriundas de diversas fontes ([Sca86][ISO93]).

ERGO-START apresenta como suporte computacional as seguintes ferramentas:

- MAD*-Toolkit para a etapa de modelagem da tarefa. Esta ferramenta implementa o meta-modelo objeto das tarefas Object-MAD*, permite a construção e edição de árvores MAD*, além de verificar a completude e a coerência das descrições;
- ICS ToolKit para a etapa de modelagem da interface. Esta ferramenta implementa a arquitetura ICS e permite estabelecer representações executáveis

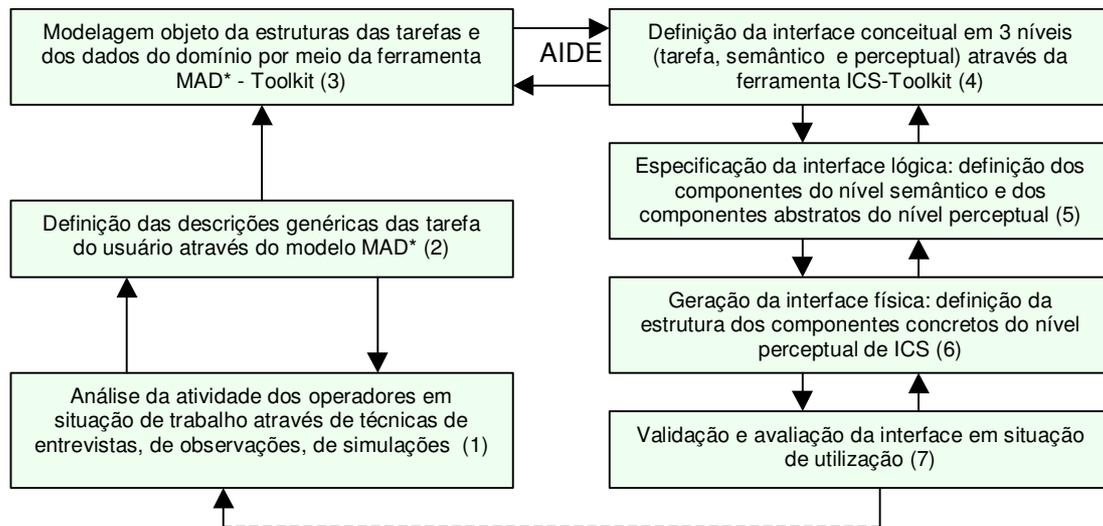


Figura 4: Particularização de ERGO-START na abordagem AIDE

A metodologia ERGO-START não apresenta de forma explícita um modelo de arquitetura, apoiando-se em ferramentas para a produção do protótipo.

2.5 ALACIE

ALACIE (*Atelier Logiciel d'Aide à la Conception d'Interfaces Ergonomiques*) ([GS97] [Gam98]) é uma ferramenta que implementa os modelos, os componentes e os algoritmos de uma metodologia que se inspira em uma abordagem de concepção seguida por ergonomistas [Seb92]. Essa metodologia tem como principais etapas:

- a análise da tarefa existente (modelagem, revisão e validação);
- a tradução das informações obtidas na análise da tarefa em termos de uma interface abstrata, que será auxiliada por um conjunto de regras com o objetivo de organizar a interface de uma forma ergonômica;
- e a especificação da interface concreta a partir de restrições dos elementos abstratos, ou seja, a geração de protótipos.

A Figura 5 abaixo ilustra as etapas, modelos e ferramentas de ALACIE:

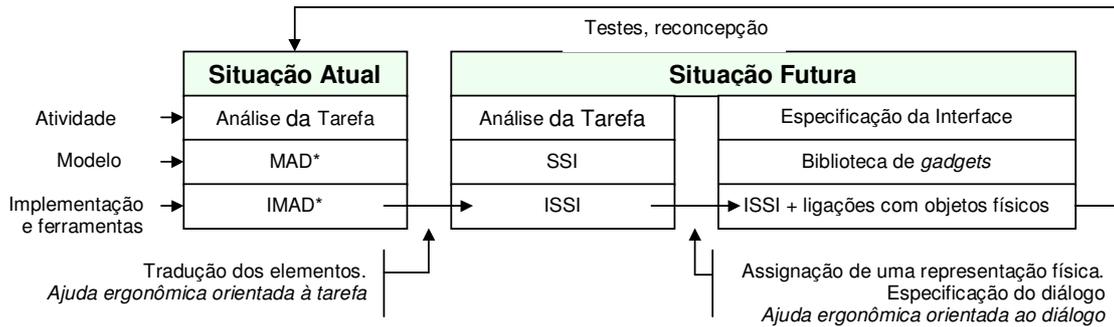


Figura 5: Metodologia de concepção utilizada em ALACIE

Os modelos utilizados nesta metodologia são:

- Uma versão modificada do modelo MAD* [Ham95] para análise e descrição da tarefa. Essas modificações dizem respeito principalmente à nomenclatura, inserção e exclusão de atributos;
- modelo SSI (*Specification Semantique de l'Interface*) [GS97] que se inspira fortemente no modelo ICS [Ham95] para a especificação conceitual da interface. SSI conserva uma organização em Esquemas_de_Procedimentos (SP) e se diferencia do modelo ICS principalmente por apresentar uma estrutura simplificada com dois níveis de abstração (tarefa e semântico) e uma arquitetura hierárquica composta apenas de 3 componentes (enquanto ICS apresenta 7): ssiSP, ssiP e ssiOF.

Esta metodologia não especifica um modelo de arquitetura da interface.

Regras ergonômicas orientadas à tarefa e orientadas ao diálogo ([Ham95], [Alo96]) podem ser utilizadas para ajudar, respetivamente no processo de tradução do modelo MAD* ao modelo SSI e no processo de especificação do diálogo.

O processo de concepção é auxiliado pelas seguintes ferramentas computacionais:

- ferramenta IMAD* que implementa o modelo MAD*. Esta ferramenta se encarrega dos aspectos que dizem respeito à descrição da tarefa e sua validação;
- e a ferramenta ISSI para implementação e manipulação do modelo SSI. Esta ferramenta ajuda o projetista na tradução dos elementos e na geração da interface (protótipo).

O principal objetivo de ALACIE é ajudar os projetistas no processo de descrição-tradução-especificação, a partir das informações obtidas durante o processo. ALACIE insere-se no quadro de ferramentas baseadas em modelos e mais particularmente no quadro de ferramentas baseada na tarefa.

2.6 MCI - Método para Concepção de Interfaces

Trata-se de um método para o projeto da interação que aborda tanto o nível conceitual, (a identificação das funções necessárias, o sequenciamento destas funções e a definição do fluxo da interação), quanto o nível perceptivo (projeto da representação visual para o usuário).

Este método tem como principal característica o ciclo de concepção centrado na avaliação, de forma que os resultados de cada etapa são avaliados antes de se prosseguir para a próxima. Ou seja, se desenvolve em um conjunto de etapas que se apóiam na construção e avaliação dos modelos de cada uma das etapas que compõem a especificação da interface.

As principais etapas deste método são:

Análise das características do usuário - nesta etapa é definida a classe de usuários em termos das tarefas que serão realizadas e do conhecimento necessário para realizá-las. O resultado é um conjunto denominado "perfil do usuário" que contém as características relevantes para a interação. Esta análise se fundamenta no modelo sintático-semântico de [Shn98].

Construção do modelo da tarefa - nesta etapa é utilizado o modelo MAD [SP89].

Avaliação da consistência do modelo da tarefa - consiste em verificar a completude da tarefa modelada.

Concepção do modelo de Interação - o modelo da interação é obtido a partir da descrição MAD da tarefa, associando-se inicialmente, através de uma tabela, objetos da tarefa a objetos da interação em seguida são descritas em detalhes as janelas e os objetos que fazem parte daquela janela.

Avaliação da consistência e completude do Modelo da Interação - consiste numa análise ergonômica do modelo, verificando-se a coerência, e a consistência através de recomendações de projeto (diretrizes e padrões) para o contexto tratado.

Construção do modelo de navegação - consiste em sintetizar as situações de transição mais comuns em uma interface com o usuário (diálogo). Para a modelagem da navegação é utilizado o formalismo CPN - Redes de Petri Coloridas, essas redes se caracterizam por apresentar uma representação formal, possuir uma descrição gráfica e serem passíveis de execução, ou seja, permite uma simulação.

Verificação do modelo de navegação - consiste em verificar se atende a um conjunto de propriedades de usabilidade definidas em determinado contexto. A ferramenta DESIGN/CPN é utilizada para construção e verificação do modelo de navegação.

Construção do protótipo - a construção do protótipo se fundamenta no modelo de interação e no modelo da navegação e utiliza a ferramenta de prototipação *DevGuide*.

Validação do protótipo quanto a aspectos ergonômicos do projeto visual - é realizada através da ferramenta FAIUNIX, e consiste em analisar o arquivo de protótipo e obter informações sobre o projeto visual, que são comparadas com recomendações ergonômicas.

Preparação do Protótipo para validação - nesta etapa é gerado o código do protótipo que será validado junto ao usuário.

Validação do protótipo junto ao usuário final - consiste em sessões de testes com os usuários e com base na captura de dados da interação através da coleta automática de dados.

A Figura 6 abaixo ilustra o ciclo de concepção do método e as ferramentas utilizadas.

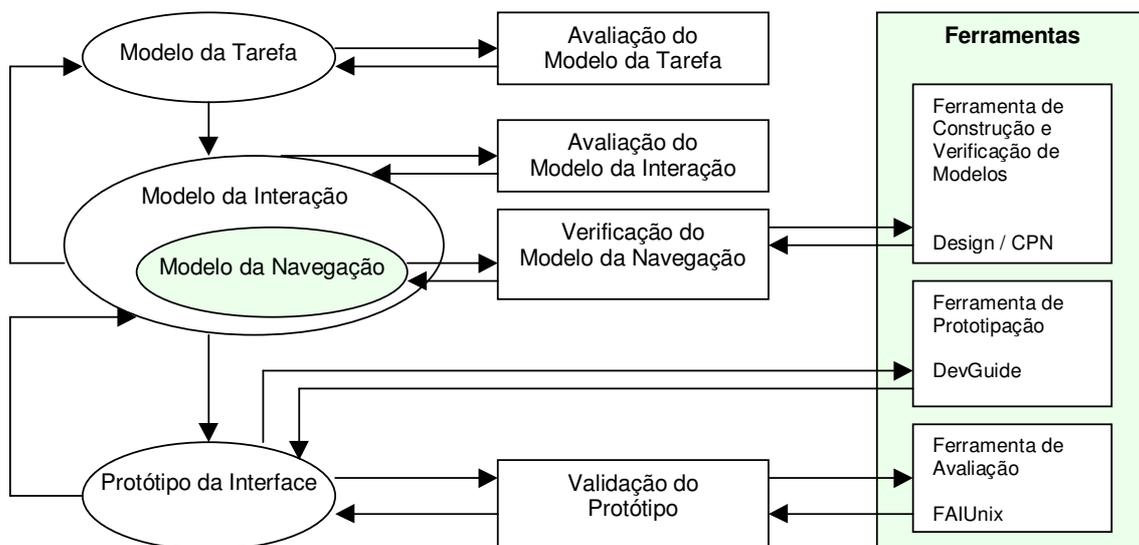


Figura 6: Ciclo de concepção no Método MCI [Sou99]

O processo consiste na realimentação de informações entre as etapas de construção e avaliação até que os objetivos de usabilidade sejam atingidos.

2.7 Discussão

As metodologias apresentadas acima se enquadram na abordagem baseada na tarefa, ou seja, partem da descrição da tarefa do usuário (modelo da tarefa) para produzir uma especificação (conceitual ou externa) da interface. O Quadro (1) abaixo apresenta uma síntese das metodologias no que diz respeito aos modelos e demais recursos utilizados.

		MODELOS E RECURSOS				
		Modelo da Tarefa	Modelo da Interação / Navegação	Modelo de Arquitetura	Fatores Humanos	Suporte Computacional
METODOLOGIAS	TRIDENT	ACG	UP, Janela, OIA, OIC	OC, OA, OI	Regras ergonômicas	Ferramentas de Geração automática
	ADEPT	TKS	AIM / CIM		Regras de Projeto	ADEPT Demonstrator
	ERGO-START	MAD*	ICS	Objeto	Regras Ergonômicas	MAD* - ToolKit e ICS ToolKit
	ALACIE	MAD*	SSI	Objeto	Regras Ergonômicas	IMAD* e ISSI (ALACIE)
	MCI	MAD	Redes de Petri Coloridas (navegação)		Avaliação dos produtos gerados, através de recomendações (diretrizes, guias para projeto, padrões, usuário final)	Design/CPN DevGuide FAI/Unix

Quadro 1: Resumo das Metodologias

De acordo com a tabela acima (Tabela 1), constata-se que todas as metodologias citadas fazem o uso de modelos em suas principais etapas, sendo estes importantes diferenciais entre uma metodologia e outra. Todas essas metodologias fazem uso de modelos de tarefa (ACG, TKS, MAD, MAD*) para dar início a especificação da interface. Esses modelos são bastante similares, descrevem a tarefa de forma hierárquica, apresentam uma sintaxe gráfica de descrição e proporcionam uma descrição completa e coerente da tarefa (em termos de objetivos, estratégias, procedimentos, ações, objetos da tarefa, etc.).

As metodologias descritas são um auxílio aos projetistas durante o processo de concepção e realização de interfaces ergonômicas (que levem em conta as necessidades e características do usuário). Contudo, ainda apresentam algumas dificuldades e limitações durante o seu uso:

a) Necessidade de experiência ou conhecimento em Ergonomia

O processo de especificação da interface conceitual na metodologia MCI é realizado pelo projetista sem o auxílio de mecanismos / procedimentos (regras) fundamentados em critérios ou recomendações ergonômicas durante a construção das especificações. Portanto, durante as etapas de construção dos modelos, espera-se do projetista alguma experiência na área de interfaces e/ou conhecimento prévio sobre Ergonomia de Software, de forma que seja capaz de relacionar os elementos vindos da tarefa com os elementos da interação de forma ergonômica. Caso contrário, o resultado da especificação produzida poderá não condizer com as necessidades e características do usuário. No caso desta metodologia (MCI), o conhecimento ergonômico é utilizado nas etapas de avaliação, verificação e validação das especificações produzidas (através de critérios e padrões ergonômicos), ou seja, o processo consiste na realimentação de informações entre as etapas de construção e avaliação até que os objetivos de usabilidade sejam atingidos. Trata-se, portanto, de um processo iterativo de avaliação.

A metodologia ALACIE disponibiliza e o uso de regras para a especificação da interface, mas não impõe esse uso. Caso não sejam utilizadas regras para efetuar a tradução de um modelo a outro, pressupõe-se, portanto, que o projetista tenha experiência e conhecimento prévio sobre Ergonomia.

As demais metodologias citadas não exigem, a princípio, conhecimento ergonômico ou experiência prévia do projetista, pois fornecem mecanismos / instrumentos dotados desse conhecimento através de regras que são utilizadas para passar de um modelo ao outro.

b) Dificuldades de escolha e de utilização de regras ergonômicas em relação às etapas do processo.

ERGO-START apresenta o conhecimento ergonômico sob a forma de regras de produção extraídas de recomendações ergonômicas. Essas regras são classificadas de acordo com as etapas da abordagem e em relação aos níveis do modelo conceitual de interação ICS. São regras que ajudam na especificação dos estilos de interação, diálogo e apresentação. No entanto, nem sempre é óbvio qual a regra e em que momento deve ser aplicada: regras classificadas em um determinado nível podem ser utilizadas no nível seguinte, confundindo o projetista. Essa dificuldade deve-se em parte a complexidade do modelo de interação utilizado (diversos níveis e componentes), não permitindo a identificação de regras que relacionem os modelos utilizados de forma clara e simples. Em ALACIE, o projetista pode fazer uso de regras ergonômicas

definidas em ([Ham95], [Alo96]), contudo o problema aqui discutido é minorado devido à simplificação do modelo de interação utilizado.

c) Dificuldade de passagem / transformação do modelo da tarefa para um modelo conceitual de interação

Em ADEPT, o modelo de interação utilizado (AIM) não é explícito, ou seja, apesar de sua estrutura ser semelhante àquela apresentada pelo modelo de tarefa (TKS) (ambas são árvores hierárquicas), não estrutura seus componentes de forma a distinguir a apresentação, o diálogo e a aplicação. O modelo AIM não induz uma visualização da estrutura da interface que se está projetando, dificultando assim o processo de transformação do modelo da tarefa ao modelo conceitual da interação.

Em ERGO-START, o modelo de interação (ICS) não tem uma relação clara com o modelo de tarefa (MAD*) utilizado. Enquanto MAD* se estrutura sob a forma de uma árvore hierárquica, ICS se apresenta sob a forma de um grafo de estados, dificultando assim o relacionamento entre elementos dos dois modelos. Além disso, a estrutura gráfica do modelo ICS não permite ao projetista uma visualização da estrutura da apresentação da interface.

MCI não apresenta um modelo explícito de interação, apresenta uma descrição / especificação sob a forma de uma tabela, que não dá ao projetista uma visão da estrutura da apresentação da interface, dificultando assim o processo de transformação do modelo da tarefa para o modelo da interação.

Por outro lado, o modelo de interação utilizado em ALACIE (SSI), é uma simplificação do modelo ICS, o número de componentes foi reduzido, facilitando e simplificando assim a sua utilização. Tais mudanças ocasionaram uma maior e melhor compatibilidade com o modelo da tarefa, facilitando assim o processo de transformação de um modelo ao outro [Gam98].

d) Dificuldades na utilização e aplicação da metodologia na ausência das ferramentas

A maioria das metodologias citadas apresenta suportes computacionais: ERGO-START possui ferramentas computacionais para a etapa de modelagem de tarefas e da interação e prevê ainda a construção de um sistema especialista baseado em regras ergonômicas para dar suporte à etapa de passagem do modelo MAD* para o ICS (esta ferramenta vem a suprir a dificuldade apontada anteriormente); MCI dispõe de ferramentas para construção e verificação da navegação e de geração do protótipo. TRIDENT, ADEPT e ALACIE destacam-se por possuir um ambiente completo e

integrado para a geração de interfaces, desde editores e visualizadores dos modelos da tarefa e da interação, ferramenta para modelar o usuário com o auxílio das bases de conhecimento, à interpretadores e geradores de protótipos.

Contudo, várias dessas metodologias ao invés de serem guiadas por modelos em determinadas etapas se apóiam e dependem das ferramentas computacionais. Etapas como a passagem do modelo da tarefa para o modelo de interação (como em TRIDENT, ADEPT, ERGO-START, ALACIE) e a de geração do protótipo (como em ADEPT, ERGO-START, ALACIE, MCI) dependem fortemente das ferramentas definidas para produzir as especificações desejadas, o que apesar da grande ajuda prestada, podem inviabilizar a utilização da metodologia no caso da ausência desse suporte computacional, pois são etapas complexas, com modelos implícitos e/ou complexos ou mesmo sem modelos para essas etapas.

e) Dificuldades em modificar o código da interface

Em geral, a maioria destas metodologias não apresenta um modelo de arquitetura bem definido. As metodologias são baseadas principalmente em ferramentas computacionais que produzem (o código da interface) que não reflete nem implicitamente nem explicitamente o modelo de interação usado. A ausência de um modelo de arquitetura tem conseqüências diretas no processo de avaliação do protótipo, porque uma modificação nesse estágio implica mudanças no modelo de interação e não diretamente no protótipo da interface.

Das metodologias estudadas, apenas TRIDENT apresenta explicitamente um modelo de arquitetura da interface.

2.8 Conclusão

Descrevemos neste capítulo diferentes metodologias para concepção de interfaces ergonômicas que se baseiam no modelo de tarefa. Em seguida, apresentamos as dificuldades e limitações detectadas no que diz respeito à sua utilização, apontando as principais hipóteses sobre as causas para esses problemas.

Com base nessa análise, elaboramos um conjunto de requisitos para a definição de uma metodologia que evite os problemas e as dificuldades apontadas buscando soluções de acordo com as hipóteses apresentadas no Capítulo 1.

Requisitos - Metodologia

1. Deve partir da descrição da tarefa do usuário;
2. O processo de especificação deve ser simples e iterativo, com etapas bem definidas;
3. Cada etapa deve produzir um artefato segundo um modelo bem definido: modelo de tarefa, modelo de interação, modelo de arquitetura;
4. Deve integrar e utilizar o conhecimento ergonômico de forma adequada, ou seja durante o processo de construção da interface, e não somente na avaliação;
5. Deve ser independente de ferramentas computacionais.

Requisitos - Modelos

1. O modelo de interação deve ser simples e refletir a estrutura da tarefa;
2. O modelo de interação deve proporcionar uma descrição gráfica da apresentação da interface, que induza no projetista uma visualização mental da estrutura da interface;
3. A estrutura do modelo de interação deve refletir de forma clara os componentes: apresentação, diálogo e aplicação;
4. O modelo de arquitetura deve ter uma estrutura que reflita a estrutura do modelo da interação.

Requisitos - Integração da Ergonomia

1. Deve apresentar uma base de regras ergonômicas que auxilie o projetista durante o processo de passagem do modelo da tarefa ao modelo de interação;
2. As regras devem ser classificadas de acordo com as etapas de especificação;
3. As regras devem obedecer a um mesmo estilo e formato de apresentação.

Capítulo 3

3 OS MODELOS DA TAREFA, INTERAÇÃO E ARQUITETURA: MAD* e EDITOR

Neste capítulo serão descritos os modelos utilizados na metodologia proposta (MEDITE): o modelo MAD* ([Ham95] [Gam98]) utilizado na etapa de análise e modelagem da tarefa e o modelo EDITOR [Lul92] como modelo unificado da interação e da arquitetura.

3.1 Modelagem da Tarefa

Analisar a tarefa significa estudar os detalhes de sua natureza, seu propósito, as partes que a compõem e a forma e a ordem em que devem ser realizadas. Envolve também o fluxo de informação e o papel do usuário [Sou99]. Na concepção de interfaces, a análise da tarefa tem o objetivo de descrever uma tarefa precisamente com o intuito de entender a “lógica do usuário”. Descrevendo a tarefa de forma precisa e exhaustiva abre-se espaço para a aplicação de regras ergonômicas que associem elementos obtidos na análise da tarefa com elementos da interação, ou seja, torna possível relacionar objetivos e características da tarefa com as características da interface que se pretende conceber ([Sca88] [Seb95]).

Técnicas ou abordagens formais podem ser utilizadas para representar o conhecimento do usuário e as ações que ele deve executar para cumprir suas tarefas com determinado sistema [Haa00].

Vários formalismos têm surgido com o objetivo de integrar a análise e a descrição da tarefa ao processo de concepção de interfaces. Entre eles podemos destacar: TKS [JJW88], MAD [SP89], ETAG ([Tau88] [Tau90] [HVV92] [Haa00]), TAOS ([Med95] [MKL00]) e MAD* ([Ham95] [Gam98]). Todos assumem que o

conhecimento sobre uma tarefa é estruturado de acordo com o paradigma do planejamento hierárquico.

O TAOS - *Task and Action Oriented System* ([Med95] [MKL00]) é um formalismo desenvolvido no GIHM (Grupo de Interface Homem-Máquina / UFPB) e validado em relação ao MAD para análise da tarefa no contexto de concepção de interfaces homem-computador [Kaf00]. Atualmente, o TAOS encontra-se em processo de validação em relação a MAD* e de desenvolvimento de uma ferramenta computacional que o implemente.

Nesse sentido, adotamos para este trabalho o Modelo MAD* para a descrição e modelagem das tarefas dada sua simplicidade, adequação aos propósitos da modelagem e por sua ampla utilização no GIHM. A seguir, o descrevemos em detalhes: seus objetivos, principais conceitos, atributos e construtores.

3.2 O Modelo MAD*

O Modelo MAD* - Modelo Analítico de Descrição de Tarefas do Usuário orientado à Especificação de Interface - [Ham95] é uma extensão do Método MAD [SP89].

MAD* tem como principais objetivos:

- suprir as lacunas e as deficiências do Método MAD (utilização de alguns de seus atributos, construtores, etc.); (mais detalhes em ([Ham95] [Gam98]));
- redefinir o Método MAD em termos informáticos (MAD*-Object);
- e facilitar a integração da tarefa do usuário no processo de concepção de interfaces.

O Modelo MAD*, assim como MAD, descreve e modela as tarefas de forma hierárquica e possui uma sintaxe gráfica de descrição.

MAD* é definido basicamente por quatro conceitos (Figura 7):



Figura 7: Principais conceitos de MAD*

Unidade-Tarefa (UT): suporte de descrição e de execução de uma tarefa MAD*; corresponde a um objetivo, distinguindo-se pelo nível de abstração, ou seja, a semântica associada à cada um deles. São distinguidos 3 níveis de UT's (Figura 8).

- UT de alto nível: objetivo global da tarefa
- UT de nível intermediário: diz respeito à estratégia de procedimentos de execução da tarefa
- UT de baixo nível: corresponde a uma ação da tarefa

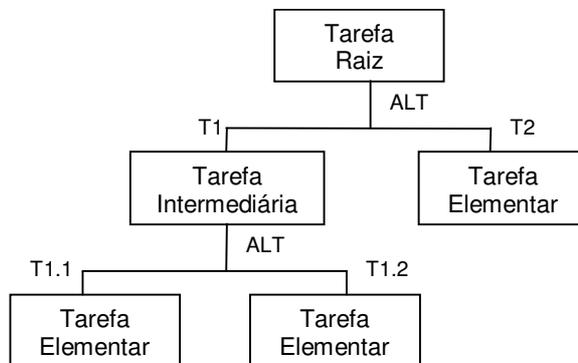


Figura 8: Níveis da UT

Corpo da Unidade-Tarefa: trata-se das condições de entrada e de saída da tarefa assim como suas estruturas; inclui também os elementos do corpo da tarefa.

- Corpo da tarefa: (nome, objetivo, modo, ator, tempo [início da execução, fim da execução e a duração da execução], prioridade, interruptibilidade e estado de execução)
- Condições de entrada da tarefa: (estado inicial, pré-condições, condições desencadeantes, repetir tarefa até a condição de parada)

- Condições de saída da tarefa: (estado final, pós-condições, resultado da execução)

Decomposição da Unidade-Tarefa: diz respeito às relações inter-tarefas (estruturais ou temporais)

- As relações estruturais: permitem definir o ordenamento das tarefas (tarefas independentes)

ET (executar todas as sub-tarefas) – tarefas paralelas independentes

OU (executar pelo menos uma das sub-tarefas)

ALT (executar somente uma das sub-tarefas) – tarefas alternativas

- As relações temporais: permitem definir a sincronização das tarefas (tarefas dependentes)

SEQ (tarefas seqüenciais) – dependência forte entre as tarefas - executadas uma pós a outra conforme ordem dada

PAR (tarefas paralelas dependentes) - a ordem de execução é restrita

SIM (tarefas simultâneas e partilhamento das tarefas) – as sub-tarefas são executadas ao mesmo tempo, inicia e finalizam juntas, implica vários operadores.

Inclui dois parâmetros: Tarefa e Operador (T1, Op1)

Os atributos da especificação da interface: integram os aspectos que visam ajudar a exploração de uma árvore de tarefas em vista da concepção da interface. Têm por objetivo integrar as recomendações orientadas a tarefa, encontradas nos guias ergonômicos, permitindo orientar as primeiras decisões de concepção da futura interface.

Estes dizem respeito ao Corpo da unidade Tarefa e à Decomposição da tarefa

- as *restrições de diálogo* permitem definir a ordem na qual as tarefas devem ser executadas. Ajuda ao projetista na definição e refinamento do sequenciamento das sub-tarefas de uma tarefa, trata-se das relações ET e PAR.
- os atributos *tipo* (mental e sensório-motor) e *modalidade* (manual, automática ou interativa) da tarefa são importantes pois ajudam na especificação das formas de interação.

- o atributo *Papel do operador* abstrai as características pertinentes aos diferentes estereótipos de operadores em suas estratégias de realização das tarefas. Uma mesma tarefa pode ser realizada de maneiras diferentes por diferentes operadores de acordo com suas aptidões e habilidades. Em MAD* trata-se de atributos que descrevem o operador em termos de sua competência e sua experiência a executar a tarefa e ao utilizar as ferramentas para a execução da tarefa. Quatro atributos definem o papel do operador:

Papel – define a função do operador da tarefa

Experiência da tarefa – indica o nível do usuário. Os valores são: iniciante, médio e complexo;

Experiência com o sistema – indica se o usuário já utilizou sistemas similares. Os valores são: elementar, médio e complexo.

Competência – indica o conjunto das ações que o usuário executa (eventualmente com outros usuários) durante a realização da tarefa.

Meios de interação – permite especificar as escolhas dos meios de interação para as tarefas que podem ser informatizadas. Podem ser: linguagem de comando, toque de funções, pergunta-resposta, formulários, multi-janelas, manipulação direta, interação icônica. Cada tarefa é associada a um estilo de interação. Vários estilos podem ser combinados. Para cada estilo de interação define-se o nível do operador: novato, médio, experiente.

- atributo *Centralidade da tarefa* - esta noção é importante na especificação da interface pois permite estruturar e hierarquizar a apresentação. Dois atributos a descrevem:

Frequência – número de vezes que a tarefa é executada. Os valores podem ser em termos de interação e no tempo: todos os dias, horas..., ou de valores *ad hoc* tais como muito elevado, elevado, médio, fraco.

Entidades Importantes – sub-conjuntos dos objetos da tarefa, que intervêm no momento da execução das tarefas.

Uma modelagem MAD* se apresenta sob a forma de uma árvore hierárquica de tarefas e cada uma dessas tarefas apresenta um descritor, onde são representados seus demais atributos.

3.3 Modelos de Interação e de Arquitetura

Apresentamos nesta seção os conceitos de modelos de interação e de arquitetura, e em seguida introduzimos e descrevemos o modelo EDITOR [Lul92] como modelo unificado de interação e de arquitetura a ser utilizado em nossa abordagem.

3.3.1 Modelo de Interação

Segundo Hammouche [Ham95] a interação pode ser percebida como uma colaboração interdependente entre dois parceiros: usuário e sistema. As descrições (modelos) desses parceiros são diferentes, notadamente em termos de suas linguagens e de suas representações respectivas do domínio. É por isso que a definição de um modelo de interação deve explicitar as relações entre esses dois modelos. Do ponto de vista do usuário, o modelo de interação deve ser baseado em representações lhe permitindo interpretar o funcionamento do sistema. Essas representações têm por objetivo colocar em correspondência a linguagem do usuário e a linguagem do sistema. A linguagem do sistema designa a linguagem da aplicação, o programa informático, que traduz os estados do sistema. Enquanto que a linguagem do usuário designa a linguagem das tarefas, no senso cognitivo do termo, que traduz os estados dos objetivos do usuário.

A partir da descrição da tarefa e do usuário pode-se realizar o projeto no nível conceitual, identificando as funções necessárias, o sequenciamento destas funções, a definição do fluxo da interação, os mecanismos de interação, os objetos com os quais o usuário deverá interagir e a alocação tarefa/função que decide quais partes da tarefa serão realizadas pelo sistema e quais serão realizadas pelo usuário. Estas decisões constituem o modelo da interação. Segundo Schlunbaum (citado em [Sou99]), os métodos para especificação formal podem ser utilizados pelos projetistas de interfaces para descrever o comportamento de um sistema sem que seja necessário realizar a sua implementação. A especificação possibilita a descrição formal dos requisitos e idéias de projetos em diferentes níveis de abstração [Sou99].

Diversos são os modelos utilizados para descrever a especificação conceitual de uma interface (ou seja, o modelo da interação) como exemplo temos: AIM (na metodologia ADEPT), ICS (em ERGO-START), SSI (em ALACIE). Cada um desses modelos apresenta diferentes particularidades e diferentes formas de descrição, alguns mais simples outros mais complexos, conforme visto no Capítulo 2.

3.3.2 Modelo de Arquitetura

Um modelo de arquitetura define um conjunto de componentes de software ao mesmo tempo sobre o plano estrutural e sobre o plano comportamental, cujas composições e articulações asseguram as modalidades de comunicação entre o usuário e a aplicação (sistema).

Diversos modelos têm sido propostos para descrever a organização de software de uma interface e na maioria deles o “princípio da independência do diálogo” é consensual. Este princípio diz respeito à separação modular entre o código da aplicação e o código da interface e trata-se de um princípio essencial que deve ser respeitado no processo de concepção de sistemas interativos. (Figura 9)

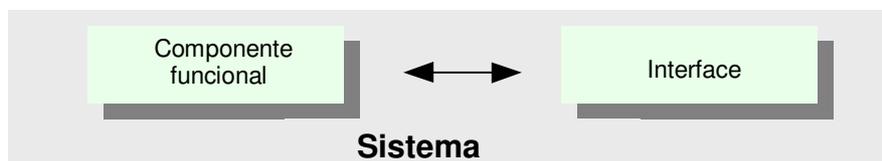


Figura 9: Decomposição do Sistema interativo

Essa decomposição é importante no sentido que permite: desenvolvimento e modificação independente dos módulos, uma concepção iterativa da interface, diversidade de competências e diversidade de plataformas e a possibilidade de reutilizar toda ou parte de uma interface para outras aplicações [Lul92].

O modelo *Input /Output* (Lantz, 1986 citado por [Lul92]) oferece uma visão sintética da organização do conjunto de rotinas responsáveis pela implantação de sistemas interativos e que estão atualmente disponíveis em todas as estações de trabalho, mas não fornece nenhuma ajuda na tarefa de organizar o módulo responsável por implantar as funções citadas acima.

O Modelo de Seeheim [Pfa85] é um dos primeiros modelos propostos para modelar uma interface para o usuário. Este modelo descreve a interface separando-a em três componentes (Figura 10):

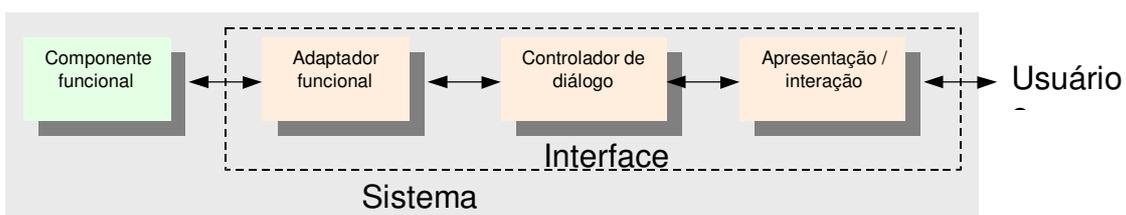


Figura 10: Modelo de Seeheim

No entanto, a visão lingüística sobre a qual este modelo repousa implica um comportamento seqüencial e uma organização centralizada que não corresponde ao estilo de diálogo das interfaces atuais (manipulação direta, por exemplo).

Por fim, o modelo multi-agente enfatiza a modularidade, o paralelismo e a comunicação por eventos, o que facilita os ajustes iterativos, a expressão de diálogos assíncronos e a expressão de retornos de informação característicos das interfaces gráficas atuais, além de encontrar nas linguagens orientadas a objetos sua ferramenta natural de realização [Lul92]. Um exemplo de modelo multi-agente é o Modelo PAC ([Cou87] [Cou90]). O Modelo PAC estrutura uma interface como uma hierarquia de agentes PAC. Um agente PAC elementar modela um objeto interativo elementar como um agente com três facetas: *Presentation (Apresentação)*, *Abstraction (Abstração)* e *Control (Controle)*. Se o objeto interativo é complexo, podem ser combinados diversos agentes elementares que formam um agente PAC composto. No entanto, este modelo é ainda bastante geral, ou seja ele não define satisfatoriamente o papel dos agentes na hierarquia.

3.4 Modelo EDITOR

Introduzimos nesta seção, o EDITOR, um modelo multi-agente que será utilizado na abordagem que propomos neste trabalho como modelo unificado de interação e de arquitetura. Aqui serão descritos sua estrutura e componentes.

3.4.1 Modelo unificado de interação e de arquitetura

O modelo EDITOR como modelo de arquitetura inspira-se no modelo PAC ([Cou87] [Cou90]) e como modelo conceitual de interação adota o conceito de *edição* como paradigma central para a interação [SBY87].

EDITOR é um modelo de arquitetura que alia os aspectos remarcáveis do modelo de Seeheim [Pfa85] às vantagens de um modelo multi-agente [Cou90]. O componente multi-agente de PAC permite modelar não somente a não-sequencialidade mas também as atividades paralelas oportunistas do usuário. PAC estrutura a arquitetura de um sistema interativo como uma hierarquia de agentes PAC: um sistema interativo é um agente PAC composto que por sua vez se decompõe em agentes intermediários até os agentes PAC elementares. Todo agente PAC possui três facetas: *Apresentação*, *Abstração* e *Controle*.

Em PAC, as três facetas funcionais (P, A, C) são distribuídas a todos os níveis de abstração: o controle é distribuído e recursivo. Assim, é possível em PAC múltiplas linhas de diálogo: ou seja, dá a possibilidade de levar a cabo várias linhas de comunicação com o sistema. Em suma, PAC é mais detalhado do que Seeheim e fornece uma arquitetura modular e distribuída. No entanto, o modelo PAC, assim como o modelo de Seeheim, é mais um modelo de realização que um modelo de concepção, pois não fornece nenhum meio ou indicação de ajuda à concepção de seus componentes: não veicula um modelo de interação nem explicita as etapas e/ou regras que permitam ao projetista identificar os seus componentes a partir da análise do domínio e/ou da tarefa.

O modelo EDITOR alia as vantagens arquiteturais do modelo PAC com as vantagens de um modelo conceitual de interação, permitindo a identificação de seus elementos a partir da descrição da tarefa através de regras claras e objetivas. EDITOR encontra na abordagem orientada a objeto um modelo computacional adequado para sua representação e realização.

3.4.2 Elementos do Modelo EDITOR

O Modelo EDITOR adota o conceito de edição como paradigma central para a interação: o processo de edição é considerado como sendo qualquer seqüência de interações entre um usuário e um sistema, onde o usuário percebe (vê, escuta, etc.) e controla (modifica, manipula, etc.) o estado da aplicação. Um editor é um componente de software que suporta este processo por apresentar ao usuário uma visão particular do estado da aplicação e mecanismos para controlar e modificar esse estado.

EDITOR define uma interface (editor) como sendo a composição de três agentes do tipo PAC, respectivamente *Editor*, *Visão*, e *Objeto_de_Interação*. Cada um desses agentes apresenta as facetas *Apresentação*, *Abstração* e *Controle*:

- *Apresentação*: define a maneira como o agente pode ser percebido (associa um objeto gráfico de uma biblioteca virtual gráfica (*virtual graphical toolkit*)).
- *Abstração*: define o conceito e a função da aplicação associados ao agente (associa um objeto ou função da aplicação).
- *Controle*: define o diálogo entre as outras duas facetas e mantém a coerência entre elas.

3.4.2.1 O agente Editor

Um agente *Editor* define um conjunto de visões sobre a aplicação (agentes *Visão*) apresentadas e organizadas em um quadro (ou janela) na tela. Esta definição é traduzida por um agente PAC composto, como ilustrado na figura 11 abaixo. Sua faceta *Apresentação* é definida a partir de um objeto *quadro* de uma biblioteca virtual. A faceta *Abstração* permite designar um objeto da aplicação e definir as modalidades de comunicação com ele. A faceta *Controle* gerencia a cooperação e as restrições entre os agentes *Visão* e serve de ponte aos eventos trocados entre os seus componentes *Apresentação* e *Abstração*.

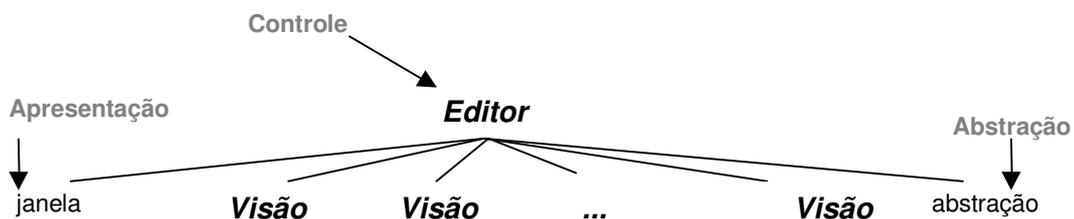


Figura 11: um agente *Editor* é um agente PAC composto

3.4.2.2 O agente Visão

Um agente *Visão* é um agente PAC composto que define o modo como é reagrupado e visualizado um conjunto de entidades (agentes *Objetos_de_Interação*). Como todo agente PAC, apresenta as três facetas. A faceta *Apresentação* define por um lado os eventos recebidos do usuário e por outro a sua imagem a partir de um objeto sub-janela ou barra de menu da biblioteca virtual que lhe é associado. A faceta *Abstração* permite designar um objeto da aplicação e definir as modalidades de comunicação com ele. A faceta *Controle* define as modalidades de troca (relações) entre *Abstração* e *Apresentação* (sub-janelas), como também entre os *Objetos_de Interação*.

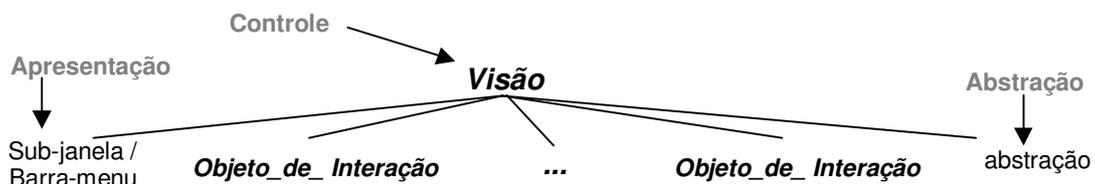


Figura 12: um agente *Visão* é um agente PAC composto

3.4.2.3 O agente Objeto_de_Interação

Um agente *Objeto_de_Interação* define uma forma de visualizar e manipular um objeto da aplicação a partir de objetos interativos de uma biblioteca virtual. A faceta *Apresentação* define (i) os eventos que o *Objeto_de_Interação* é susceptível de receber provenientes do usuário (entrada) e (ii) sua imagem através de um objeto específico de uma biblioteca virtual que lhe é associado. A faceta *Abstração* define um elemento da aplicação e as modalidades de comunicação entre eles. Enfim, a faceta *Controle* define as modalidades de trocas (de dados e de controle) e de transformação de formalismo entre os componentes.

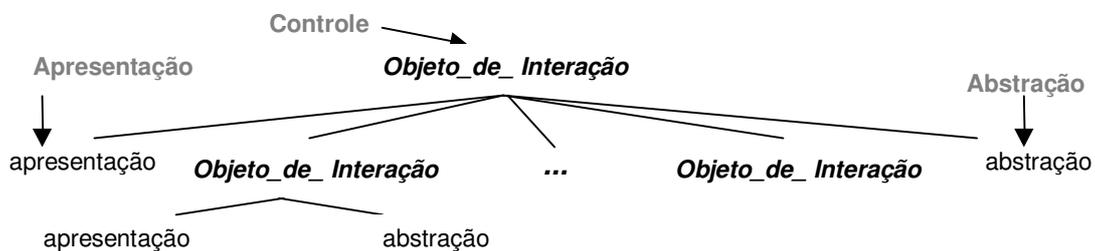


Figura 13: um agente *Objeto_de_Interação* pode ser simples ou composto

Conforme vimos, os agentes *Editor* e *Visão* são agentes PAC complexos e os agentes *Objetos_de_Interação* podem ser simples ou compostos. A Figura 14 abaixo ilustra a descrição completa da organização multi-agente de uma interface definida segundo o modelo. Esta organização multi-agente hierárquica de uma interface reflete a composição visual de sua imagem e a natureza de seus elementos.

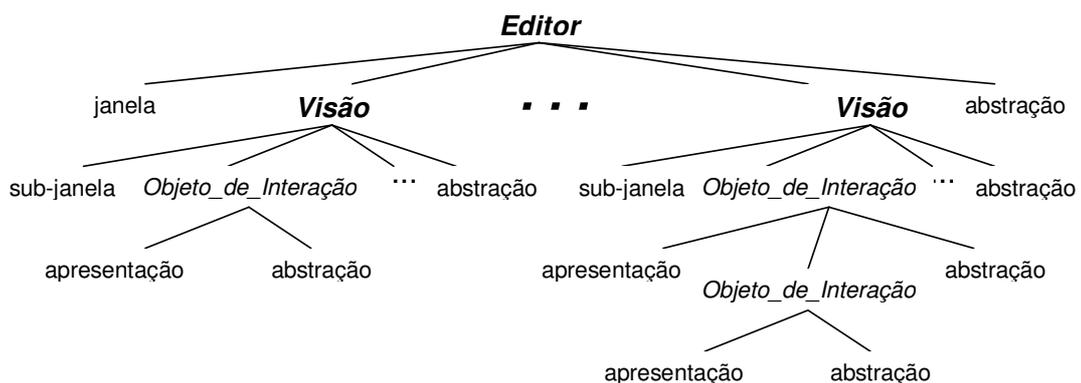


Figura 14. Organização multi-agente de um *Editor*

O modelo EDITOR como modelo conceitual da interação está de acordo com a Teoria da Ação [Nor86] na qual o processo de realização de uma tarefa através de um sistema computacional engloba duas etapas fundamentais e que se sucedem durante a interação: avaliação e execução. Como modelo de arquitetura da interface ele reflete

fielmente os conceitos e estrutura do modelo da interação e tem no paradigma de programação orientada a objeto um mecanismo natural de realização.

3.5 Conclusão

Apresentamos neste capítulo o modelo MAD* como formalismo para análise e modelagem de tarefas e o modelo EDITOR como modelo de interação e de arquitetura utilizado na metodologia proposta.

A descrição e a modelagem da tarefa segundo o modelo MAD* possibilita uma descrição completa e coerente da tarefa (objetivos, estratégias, procedimentos, objetos...) que nos permite entender a "lógica do usuário" utilizada para a realização de sua tarefa. Essa descrição deve ser fielmente considerada durante o processo de concepção de interfaces. MAD* integra em sua descrição elementos de suma importância em nosso contexto de estudo: atributos orientados à especificação da interface, informações sobre o operador, estilos de interação para a tarefa descrita, entre outros.

Como modelo conceitual de interação, o EDITOR guarda com o modelo da tarefa MAD* duas relações de proximidade:

- estrutural, ou seja, uma descrição EDITOR tem a mesma estrutura (árvore) que uma descrição MAD*;
- e procedimental, ou seja, o processo de construção da árvore EDITOR é similar ao processo de construção da árvore MAD* (descendente e ascendente).

Essas duas relações de proximidades facilitam a identificação de regras de produção para transformar uma descrição MAD* em uma descrição EDITOR.

O Modelo EDITOR proporciona uma descrição gráfica clara e intuitiva da interface ao projetista: a organização multi-agente hierárquica da interface reflete a composição visual de sua imagem e a natureza de seus elementos [Lul92].

Visto como modelo de arquitetura, o EDITOR permite uma decomposição uniforme (*Apresentação*, *Abstração* e *Controle*) em todos os níveis de definição da interface, refletindo fielmente a estrutura do modelo da interação, e tem no modelo de programação orientado a objeto uma ferramenta natural de realização. Ou seja, a estrutura do código do protótipo gerado guarda uma relação de isomorfismo com o modelo conceitual da interação utilizado, o que facilitará para o projetista da interface

possíveis modificações no protótipo, e pode ser implementada por qualquer ferramenta (linguagem) de desenvolvimento de software orientado a objeto.

Enfim, seja como modelo de interação ou como modelo de arquitetura, EDITOR é um modelo bem definido, formalizado através de uma gramática livre-de-contexto [Lul92] e satisfaz os requisitos (relacionadas à modelos) listados no Capítulo 2.

Capítulo 4

4 BASE DE REGRAS ERGONÔMICAS

Neste capítulo apresentamos a estrutura geral da Base de Regras Ergonômicas de auxílio à concepção de interfaces para a metodologia proposta. Aqui são definidos os tipos de regras que constituem essa base e sua estrutura. Ainda neste capítulo é apresentado como exemplo, um conjunto de regras elaboradas segundo a proposta da base.

4.1 Origem das Regras

Um número considerável de estudos em Ergonomia e em Psicologia Cognitiva tem contribuído significativamente nas duas últimas décadas para a compreensão e a análise dos aspectos envolvidos na interação homem-computador. Estes estudos têm gerado vários tipos de contribuições no que diz respeito ao desenvolvimento de software, das quais podem-se destacar [Mat95]:

- Guias de recomendações - são publicações que agrupam recomendações derivadas e validadas empiricamente. São bastante genéricos e sem preocupação com a aplicação sistemática das recomendações propostas. Ex.: Guia de Smith & Mosier [SM86], Guia de Scapin [Sca86];
- Normas ou Padrões - são documentos oficiais ou não, disponíveis publicamente. que fornecem requisitos para o projeto (e avaliação) de interfaces. Ex.: ISO 9241 [ISO93];
- Guias de estilo - são documentos que normalmente são produzidos por uma organização e disponibilizados comercialmente. Contém a descrição de objetos de determinadas formas de interação, tais como, janelas, caixas de

diálogo, uso do teclado e mouse. Ex.: Guia de Estilos para Serviços de Informação em Ciência e Tecnologia [Par97];

- Critérios - que auxiliam ao diagnóstico da avaliação de interfaces homem-computador. Ex.: *Ergonomic Criteria for the Evaluation of Human-Computer Interfaces* [BC93].

É grande a variedade e a quantidade desses trabalhos, que vêm sendo de grande utilidade para projetistas e avaliadores de interface, principalmente para estes últimos. Utilizados como guias de avaliação de interfaces em técnicas como a avaliação heurística, teste de usabilidade, conformidade com recomendações, exploração cognitiva, inspeção por padrão, entre outras, têm se mostrado bastante úteis para diagnosticar falhas no produto e conseqüentemente no projeto. Um trabalho bastante interessante foi desenvolvido recentemente, o qual propõe uma abordagem híbrida para avaliar a usabilidade de interfaces [Que01].

No entanto, em sua forma atual, esses trabalhos são difíceis de serem utilizados no processo de concepção. Principalmente por não fazerem referência explícita à tarefa nem a um modelo de interação particular. Torna-se necessário portanto, transformar essas recomendações em regras aplicáveis que façam correspondência explícita entre as características da tarefa e as características do modelo de interação utilizado ([Sca88] [Seb95] [Rob96]). Tal procedimento é utilizado em ([Ham95] [Gam98] [Fur97]). Essas regras podem então ser organizadas na forma de uma base de conhecimento e utilizadas manualmente e/ou automaticamente (Sistemas Especialistas) no processo de geração da representação conceitual da interface segundo um modelo de interação específico.

As bases de conhecimento sobre Ergonomia são importantes pois fornecem uma ajuda relevante aos projetista de interfaces, principalmente aqueles que não tem uma formação e conhecimento em Ergonomia, pois permitem a utilização desse conhecimento durante o processo de concepção de interfaces de uma forma simples e direta.

Portanto, o objetivo da nossa Base de Regras é proporcionar aos projetistas um auxílio na utilização do conhecimento ergonômico durante o processo de transformação do modelo da tarefa (MAD*) ao modelo da interação (EDITOR) levando em conta os objetivos, características e necessidades do usuário.

4.2 Estrutura e Tipos de Regras

A estrutura das regras que farão parte da Base é bastante simples e usual. São regras do tipo SE - ENTÃO relacionando elementos e atributos do modelo MAD* ou da Árvore EDITOR a elementos e atributos do Modelo EDITOR.

A base de regras que propomos contém 2 tipos de regras:

- **Regras para Construção da Árvore EDITOR** - São regras que relacionam atributos e elementos do modelo MAD* aos atributos e elementos do modelo EDITOR. São regras que dizem respeito à produção inicial da interação, ou seja, à construção da árvore EDITOR (por exemplo, ver Regras 2, 3, 4 mais adiante).

Estas regras definem cada agente do modelo: *Editor*, *Visão* e *Objeto_de_Interação* a partir da decomposição da árvore MAD* e determinam aspectos relacionados à estrutura e estilos de interação.

- **Regras para Definição dos Atributos** - Estas podem ser: regras que relacionam atributos e elementos do modelo MAD* aos atributos e elementos do modelo EDITOR; ou regras que relacionam a Árvore EDITOR construída anteriormente aos atributos e elementos do modelo EDITOR.

Estas regras definem os atributos das facetas de cada agente: *Apresentação* (por exemplo, localização, formato, tipo e tamanho de fonte); *Abstração* (em relação ao domínio da aplicação); e *Controle* que define o diálogo.

4.3 Exemplos de Regras

As regras ergonômicas que serão apresentadas foram extraídas de diversas fontes da Ergonomia ([Cyb94] [ISO93] [Par97] [Sca86] [SM86] [Val00]) e foram elaboradas segundo a proposta da Base descrita acima. Estas regras se propõem a auxiliar o processo de transformação do modelo da tarefa ao modelo da interação.

Em cada regra são destacados os elementos e atributos que estão sendo relacionados, facilitando assim a identificação das regras no momento de aplicação. Para cada regra são apresentadas as recomendações de onde se originou.

4.3.1 Regras para Construção da Árvore EDITOR

Regra 01: SE a tarefa for de alto nível (MAD*) ENTÃO definir um Editor (EDITOR) cuja *Apresentação* deve ser do tipo *Janela*.

Regra 02: SE a tarefa for de nível intermediário (MAD*) ENTÃO definir um Editor (EDITOR) cuja *Apresentação* deve ser do tipo *Sub-janela ou Janela*.

Recomendação: A escolha de uma ou mais janelas é determinada pela decomposição da tarefa em sub-tarefas [Sca86].

Regra 03: SE as sub-tarefas forem ligadas pelo construtor OU / ALT (MAD*) ENTÃO definir para esse conjunto de tarefas uma Visão (EDITOR) cuja *Apresentação* deve ser do tipo *Barra de Menu* e cada sub-tarefa deve ser um Objeto de Interação (EDITOR) do tipo *item-menu* dessa *Visão*.

Recomendação: Considerar a seleção por menu para tarefas que envolvem a escolha entre um conjunto restrito de alternativa de ações, que requer pequena entrada de dados, onde os usuários devam ter um pequeno treinamento e onde a resposta do computador é relativamente rápida [SM86].

Regra 04: SE as sub-tarefas forem ligadas pelo construtor SEQ (MAD*), ENTÃO definir uma Visão (EDITOR) apenas para a primeira sub-tarefa da seqüência. As demais sub-tarefas tomarão o lugar de uma outra *Visão* ou a *Visão* correspondente à primeira sub-tarefa. (depende do diálogo).

Recomendação: o diálogo deve apresentar ao usuário somente a informação relacionada para a conclusão da tarefa [ISO93].

Regra 05: SE a tarefa for de alto nível ou de nível intermediário (MAD*) ENTÃO definir uma Visão (EDITOR) específica, cuja *Apresentação* será do tipo *Região*. O Objeto de Interação desta *Visão* deve ter uma *Apresentação* do tipo *Caixa de Texto*. (Este objeto deve conter texto de orientação, explicando em que estado o usuário se encontra, e quais os procedimentos que deve tomar naquele estado).

Recomendação: Quando uma seleção é feita, deve ser oferecido um feedback imediato (e.g. mensagem, explicação da opção obtida) [Sca86]

Regra 06 SE o objetivo (MAD*) da sub-tarefa for selecionar um item dentre uma lista pré-definida ENTÃO a Visão (EDITOR) relativa à essa sub-tarefa deve ter uma *Apresentação* do tipo *Barra de Menu*, que por sua vez deve conter Objetos de Interação (EDITOR) do tipo *item-menu* cuja *Apresentação* deve ser do tipo *texto*.

Regra 07 SE o objetivo (MAD*) da sub-tarefa for selecionar um item entre uma lista de itens de número desconhecido ENTÃO a Visão (EDITOR) relativo à essa sub-tarefa deve ter uma *Apresentação* do tipo *Região*, que por sua vez deve conter um único Objeto de Interação (EDITOR) do tipo *Menu* cuja *Apresentação* deve ser do tipo *Texto*.

Recomendação: Estruturas de menus devem ser aplicadas dependendo da característica da tarefa: (i) se um número limitado de escolhas são relevantes para a conclusão da tarefa dentro de um contexto específico; (ii) opções devem ser apresentadas para que a tarefa seja executada efetivamente [ISO93].

Regra 08: SE a centralidade da tarefa for importante e sua frequência elevada (MAD*) ENTÃO optar por uma interação icônica (tipo de menu) (EDITOR)

OU (alternativa de solução)

Regra 09: SE a centralidade da tarefa for importante e sua frequência elevada (MAD*) ENTÃO optar por um diálogo de tipo menu hierárquicos (tipo de menu) (EDITOR).

Recomendação: Propor uma seleção fácil para as opções importantes. Os menus hierárquicos permitem o acesso imediato à opções críticas ou freqüentemente utilizadas [Sca86].

4.3.2 Regras para Definição dos Atributos

Regra 10: SE a tarefa for de alto nível (MAD*) ENTÃO o tamanho, formato e localização da janela (atributos EDITOR) será o pré-definido pelo tipo de plataforma utilizado. Ex. Área do Browser, tratando-se de sistemas para Web.

Regra 11: SE a tarefa for de nível intermediário (MAD*) ENTÃO o tamanho, formato e localização das sub-janelas (atributos EDITOR) serão definidos dentro de uma região da janela de forma padronizada.

Recomendação: os formatos, os tamanhos e localizações das janelas de um determinado sistema devem ser padronizados de forma a minimizar o número de operações do usuário no momento de executar a tarefa [ISO93].

Recomendação: todas as janelas do sistema devem ter uma aparência consistente [ISO93].

Regra 12: SE a frequência da tarefa é elevada (MAD*) ENTÃO dispor a estrutura de menus em área que permita acesso imediato do usuário. (atributos EDITOR - localização)

Recomendação: Quando a hierarquia de menus é muito utilizada, projetar sua estrutura de forma que o usuário tenha acesso imediato às opções críticas e frequentemente utilizadas [SM86].

Regra 13: SE a frequência da tarefa for elevada (MAD*) ENTÃO dispor esses itens no alto da tela (atributos EDITOR - localização).

Recomendação: os itens mais utilizados estarão no alto da tela e os menos utilizados abaixo [Sca86].

Regra 14: SE o número de opções a escolher para concluir ou prosseguir uma determinada tarefa for grande (oito ou mais) (Árvore EDITOR parcial), ENTÃO agrupar logicamente as opções e optar pela orientação vertical. (atributos EDITOR - orientação)

Regra 15: SE o número de opções a escolher para concluir ou prosseguir uma determinada tarefa for pequeno (sete ou menos) (Árvore EDITOR parcial), ENTÃO pode-se optar pela orientação horizontal. (atributos EDITOR - orientação).

Recomendação: Se o menu contém um grande número de opções (oito ou mais) e estas opções podem ser logicamente agrupadas, opções devem ser agrupadas em função de categorias lógicas que possam ser entendidas pelo usuário [ISO93].

Recomendação: A orientação preferencial de um menu deve ser a vertical, a orientação horizontal deve ser considerada quando o número e o comprimento das opções for reduzido [Cyb94].

Regra 16: SE as tarefas implicam ações similares (MAD*) ENTÃO propor procedimentos similares de acesso às opções de menus. (atributos EDITOR - diálogo)

Recomendação: As apresentações devem ser homogêneas: localização similar das janelas, formato de telas, procedimentos similares de acesso às opções de menus [Sca86].

Recomendação: Os menus devem ser apresentados sucessivamente em um mesmo lugar, mais que simultaneamente em diferentes lugares da tela [Sca86].

Recomendação: O formato dos menus deve ser homogêneo a cada nível de menu e uma função de acesso ao nível imediatamente superior deve ser fornecido [Sca86].

Regra 17: SE uma determinada opção de menu (Árvore EDITOR parcial) for escolhida, ENTÃO sua apresentação deve ser destacada (atributos EDITOR). Ex.: mudar a cor, etc.

Recomendação: Quando o usuário seleciona uma opção de menu, se não há resposta natural e imediata, então o computador deve mostrar algum sinal de tal entrada [SM86].

Regra 18: SE a orientação do menu for vertical (Árvore EDITOR), ENTÃO alinhar as opções de escolha à esquerda e de cima para baixo (atributos EDITOR - alinhamento).

Regra 19: SE a orientação do menu for horizontal (Árvore EDITOR), ENTÃO dispor as opções de escolha da esquerda para a direita no canto superior esquerdo (atributos EDITOR - alinhamento).

Recomendação: Respeite as convenções de layout de leitura da informação. Em países ocidentais isto significa da esquerda para a direita e de cima para baixo. As informações mais importantes devem estar localizadas no canto superior esquerdo [Val00].

Recomendação: o texto das opções deve ser alinhado em colunas e justificado à direita [Sca86].

Regra 20: SE a cor da fonte (Árvore EDITOR) utilizada for escura ENTÃO utilizar background com cores claras e vice-versa (atributos EDITOR - formato / cor).

Recomendação: Se for usado um fundo colorido, selecione as cores do texto de modo a obter o contraste mais forte entre o texto e o fundo. Isso aumenta a visibilidade e a legibilidade do texto [Par97].

Regra 21: SE o Objeto de Interação for do tipo Ícone (Item-menu) (Árvore EDITOR parcial) ENTÃO mostrar uma identificação verbal para cada Ícone, centralizada e em sua margem inferior (atributos EDITOR - formato).

Recomendação: Se ícones são usados para representar ações de controle em menus, mostrar identificação verbal em cada ícone para assegurar que o significado seja entendido pelo usuário [SM86].

Recomendação: Identifique os ícones com uma descrição breve e de fácil compreensão, disponha-a centralizada em sua margem inferior e evite abreviações e termos não familiares [Val00].

Regra 22: SE o Objeto de Interação for do tipo Ícone (Item-menu) (Árvore EDITOR parcial) ENTÃO apresentar todos os ícones com mesmo tamanho, estilo, contorno e fonte de luz.(atributos EDITOR - tipo/ tamanho / formato)

Recomendação: Mantenha uma aparência visual consistente em todos os ícones, utilizando o mesmo tamanho, estilo, contorno e fonte de luz [Val00].

Regra 23: SE o Objeto de Interação for do tipo Texto (Item-menu) ou Caixa de Texto ENTÃO utilizar sempre a mesma fonte, de tamanho entre 12 a 20 pontos (atributos EDITOR - tipo / tamanho). Recomenda-se o uso de fontes com serifa (Ex.: Times New Roman.)

Recomendação: No corpo do texto não devem ser misturadas fontes de famílias diferentes [Val00].

Recomendação: O corpo do texto deve ser escrito em fontes proporcionais e tamanho legível (dependendo da fonte 12 a 20 pontos) [Val00].

Recomendação: Para o corpo do texto deve-se utilizar fontes com serifa, pois elas apresentam maior legibilidade e velocidade de leitura [Cyb94].

Recomendação: O formato de apresentação do texto deve ser consistente de uma tela para outra [Val00].

Regra 24: SE o Objeto de Interação for do tipo Texto (Item-menu) ou Caixa de Texto (Árvore EDITOR parcial) ENTÃO utilizar cor

fonte contrastante com a cor de fundo (atributos EDITOR - formato).

Recomendação: Formate o texto empregando uma cor de fundo suave, mantendo assim a boa legibilidade dos caracteres [Val00].

Regra 25: SE o Objeto de Interação for do tipo Texto (Item-menu) (Árvore EDITOR parcial) ENTÃO utilizar o estilo sublinhado para indicação de links de hipertexto (atributos EDITOR - formatação).

Recomendação: Utilize de preferência o estilo sublinhado para indicação de links de hipertexto [Val00].

Regra 26: SE o Objeto de Interação for do tipo Texto (Árvore EDITOR parcial) ENTÃO o texto deve ser justificado à esquerda (atributos EDITOR - alinhamento).

Regra 27: SE o Objeto de Interação for do tipo Caixa de Texto (Árvore EDITOR parcial) ENTÃO o texto deve ser justificado à esquerda (atributos EDITOR - alinhamento).

Recomendação: O texto deve ser justificado à esquerda [Sca86].

4.4 Conclusão

Apresentamos neste capítulo a Base de Regras Ergonômicas a ser utilizada na metodologia MEDITE. Trata-se de uma base de regras de produção (do tipo SE ENTÃO) extraídas de diversas fontes da Ergonomia que auxiliam o projetista durante o processo de passagem da descrição MAD* à descrição EDITOR da interação.

A Base é composta por dois tipos de regras: (i) para Construção da Árvore; (ii) e para Definição dos Atributos do modelo EDITOR.

Essa classificação segue fielmente as etapas do processo de especificação da interface conceitual, o que facilita ao projetista a sua utilização, ou seja, a identificação de qual regra e em que momento deve ser aplicada, assim como possibilita a automatização de grande parte do processo de especificação da interface conceitual.

Ainda neste capítulo foi apresentado um exemplo de uma Base de Regras. São regras formuladas a partir de diferentes e diversas fontes da Ergonomia (recomendações, padrões, etc.). No entanto, essa base pode ainda ser acrescida de outras

regras extraídas de outras fontes, que sigam essa estrutura, ou mesmo podem ser criadas outras Bases, específicas para determinados contextos e aplicações.

Contudo, vale salientar que, assim como as recomendações são sujeitas a controvérsias, podendo até ser contraditórias em diferentes contextos, as regras podem sofrer desse mesmo problema. Apesar disso, consideramos que regras desse tipo são de grande ajuda aos projetistas, principalmente para aqueles que não têm experiência e conhecimento sobre Ergonomia, pois além de serem de mais fácil aplicação por fazerem referências explícitas aos modelos utilizados, mostram a grande variedade de problemas e pontos a serem levados em conta durante a concepção de interfaces para o usuário, como também contribuem para a formação do projetista.

Enfim, a Base de Regras Ergonômicas proposta para a metodologia MEDITE satisfaz os requisitos listados no Capítulo 2.

Capítulo 5

5 A METODOLOGIA MEDITE

5.1 Introdução

Neste capítulo apresentamos a nossa proposta de uma metodologia para a concepção de interfaces ergonômicas orientada a modelos e baseada na análise e descrição da tarefa, a metodologia MEDITE (MAD* + EDITOR + ERGONOMIA).

MEDITE utiliza conhecimento ergonômico para auxiliar a construção da interação segundo um modelo conceitual (EDITOR) a partir da descrição da tarefa (modelo MAD*). O conhecimento ergonômico é estruturado e representado sob a forma de regras de produção que relacionam aspectos e elementos da tarefa com as características da interface que se pretende projetar (as regras apresentadas no capítulo anterior são um exemplo desse conhecimento).

O objetivo principal de MEDITE é auxiliar o projetista (principalmente aquele que não tem conhecimento sobre Ergonomia ou a equipe de desenvolvimento não dispõe de ergonômistas) no processo de especificação de interfaces ergonômicas ou que agreguem um grau elevado de conhecimento ergonômico. Portanto MEDITE deve guiar o projetista passo a passo, segundo modelos bem definidos, durante a construção a fim de obter interfaces que reflitam os objetivos, as características e as necessidades do usuário.

Para uma melhor compreensão da metodologia utilizamos um exemplo simples de forma a ilustrar cada etapa do processo. A seguir, descrevemos de forma sucinta o exemplo "Um Tutorial na Web" [GL01].

5.2 Exemplo: Um Tutorial na Web

As possibilidades do uso da Internet através de ferramentas gráficas, como a WWW (*World Wide Web*), para disseminação de informações técnicas e científicas, na promoção do ensino à distância ou no apoio ao ensino tradicional são inúmeras [MH96]. Os Tutoriais disponíveis na Web são exemplos de meios amplamente utilizados como recursos didáticos de auxílio à aprendizagem.

Na disciplina "Teoria da Computação" do Curso de Ciência da Computação da UFPB, percebeu-se a necessidade de um recurso extra-classe que pudesse ajudar os alunos numa melhor absorção dos conceitos teóricos-abstratos no tema referente à "Máquina de Turing".

Com esse objetivo, propôs-se a concepção de um site na Internet (um tutorial) onde toda a informação sobre o tema proposto pudesse ser disponibilizada de forma mais intuitiva, agradável e numa linguagem mais próxima aos alunos, e que o sistema proporcionasse aos seus usuários, os alunos da disciplina, uma boa usabilidade.

Fez-se uma pesquisa com o objetivo de saber quais eram as necessidades dos usuários em relação à esse sistema e compreender com que objetivos, os tutoriais na Web são consultados.

Constatou-se que o usuário de um Tutorial na Web deseja:

- Adquirir conhecimento a respeito do tema do Tutorial;
- Praticar, buscar problemas e exercícios referentes ao tema;
- Buscar outras referências a respeito do tema ou relacionadas à ele;
- Estudar tópicos específicos do tema;
- Tirar suas dúvidas a respeito do tema;
- Contatar pessoas que possam ajudá-lo.

Portanto, são estes os principais objetivos, entre outros, de um Tutorial na Web. Considerando que as demais etapas do processo de concepção (desde a descrição do perfil do usuário, análise de sistemas similares, etc...) são etapas já realizadas, nos centramos nas demais etapas que são os objetivos da Metodologia MEDITE, descrita logo a seguir.

5.3 Descrição de MEDITE

Apresentamos nesta parte os processos, os produtos, as etapas e os modelos envolvidos na metodologia MEDITE. A Figura 15 abaixo ilustra os *processos* através

de círculos, por meio de retângulos os *produtos* gerados e as *ferramentas conceituais* (modelos) utilizadas em cada processo.

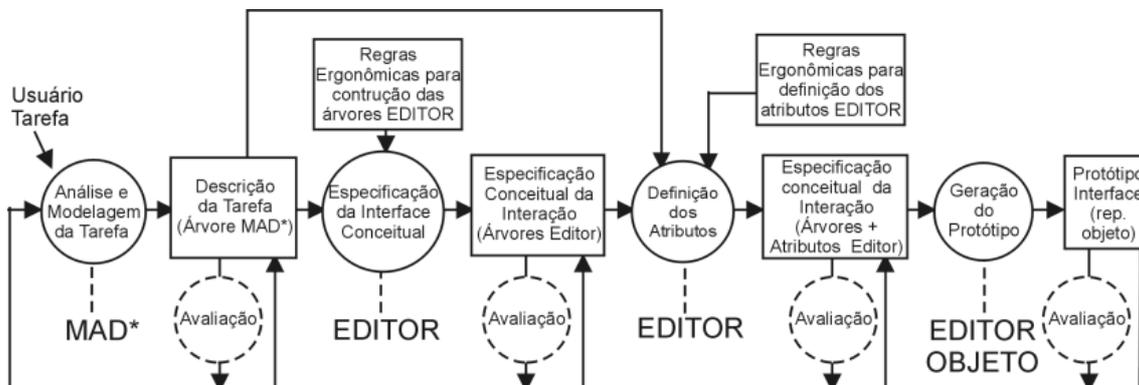


Figura 15: Metodologia MEDITE

MEDITE define o processo de construção de interfaces em 5 etapas: análise e modelagem da tarefa, especificação conceitual da interação (construção da Árvore EDITOR), definição dos atributos, geração do protótipo e avaliação.

5.3.1 ETAPA 1: Análise e Modelagem da Tarefa

Esta etapa consiste na análise e modelagem da tarefa. O objetivo é identificar, através do usuário e do domínio da tarefa, os objetivos do sistema que se pretende conceber. Em seguida descrever precisamente a tarefa com o intuito de entender a “lógica do usuário”, ou seja, a maneira, os procedimentos e objetos que ele utiliza para executá-la. Esta etapa é realizada utilizando o Modelo MAD*.

Esta etapa tem como entrada os dados sobre o usuário e o domínio da tarefa. O produto gerado no final desta etapa é a descrição MAD* da tarefa (árvore MAD* e seus descritores). Apresentamos abaixo (Figuras 16 e 17) parte da descrição MAD* da tarefa "Consultar um Tutorial na Web" (o apêndice A apresenta a árvore MAD* completa do exemplo).

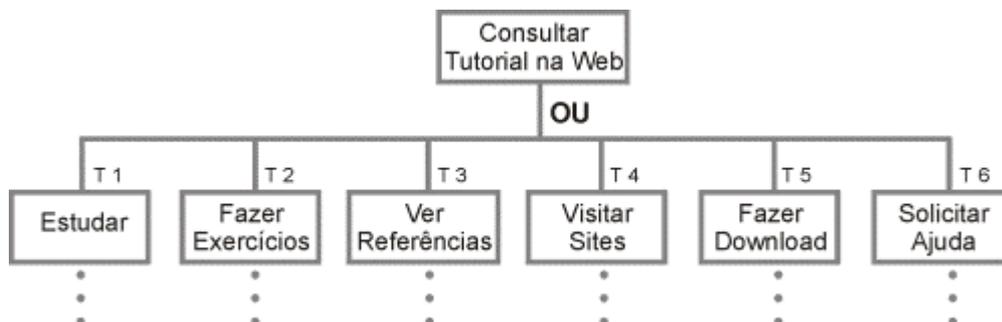


Figura 16: Tarefa-Raiz Consultar Tutorial na Web



Figura 17: Tarefa Estudar

Após a construção da árvore MAD* são definidos os elementos e atributos de cada uma das tarefas. Cada descritor pode ser identificado através de seu número e nome, conforme é apresentado abaixo:

(T 1) "ESTUDAR" (Nível Intermediário)

CORPO DA UNIDADE-TAREFA

CORPO DA TAREFA

NÚMERO DA TAREFA:	T 1
NOME	Estudar
OBJETIVO	Estudar o tema contido no Tutorial sobre Máquina de Turing
MODO	Facultativa
ATOR	Aluno (alunos de teoria da computação)
PRIORIDADE	3 (0 a 3)
INTERRUPTIBILIDADE	Interruptível com reinício em curso
ESTADO DE EXECUÇÃO	-----

CONDIÇÕES DE ENTRADA DA TAREFA

ESTADO DO MUNDO INICIAL	Ver Estado do Mundo Inicial da T 0 Ícone Estudar
PRÉ-CONDIÇÕES / CONDIÇÕES DESENCADEANTES	Ver Pré-condições da T 0 Ícone Estudar disponível = verdade

CONDIÇÕES DE SAÍDA DA TAREFA

ESTADO DO MUNDO FINAL	Browser, Tutorial, Aluno, Conhecimento ou Informação (assuntos e tópicos)
POÓS-CONDIÇÕES / RESULTADOS DA EXECUÇÃO	Browser utilizado = verdade Tutorial consultado = verdade Aluno satisfeito = verdade Conhecimento ou Informação adquiridos (assuntos e tópicos estudados) = verdade

DECOMPOSIÇÃO DA UNIDADE-TAREFA

RELAÇÃO:	SEQUENCIAL (SEQ)
LISTA DE SUB-TAREFAS:	T 1.1 - ESCOLHER ASSUNTO T 1.2 - ESTUDAR ASSUNTO
NÍVEL SUPERIOR:	T0 - CONSULTAR TUTORIAL
TAREFA ELEMENTAR:	Não

ATRIBUTOS ORIENTADOS À CONCEPÇÃO DA INTERFACE

RESTRICÇÕES DE DIÁLOGO	-----
TIPO	Mental e sensório-motor
MODALIDADE	Interativa
CENTRALIDADE	Importante Frequência: elevada Entidades importantes: browser, tutorial, aluno, conhecimento

PAPEL DO OPERADOR	Aluno Experiência com a tarefa: iniciante, média, [iniciante, média e alta] Experiência com o sistema: elementar e médio [elementar, médio e <i>expert</i>] Competência: selecionar um ícone entre vários, navegar na Internet
MEIOS DE INTERAÇÃO	Interação icônica

A descrição completa da tarefa "Consultar Tutorial na Web", (árvore MAD* e seus descritores) pode ser encontrada em [GL01].

5.3.2 ETAPA 2: Especificação conceitual inicial da Interação

Esta etapa consiste em produzir a especificação conceitual inicial da interação. Trata-se do processo de construção das árvores EDITOR, ou seja, é nesta etapa que cada agente do Modelo EDITOR (*Editor, Visão e Objeto_de_Interação*) é definido.

Esta etapa tem como entrada a descrição MAD* gerada na etapa anterior e é o momento da primeira inserção do conhecimento ergonômico, aqui representado sob a forma de regras de produção, como visto no capítulo anterior. O processo de transformação da árvore MAD* em árvore EDITOR é realizado e conceitualizado com facilidade por meio das Regras Ergonômicas para Construção da Árvore EDITOR (apresentadas no capítulo anterior), visto que a estrutura de um agente *Editor* guarda uma relação de proximidade (isomorfismo) com a estrutura de uma descrição MAD*.

São especificados neste momento aspectos relacionados com a estrutura, o sequenciamento, estilos de interação e layout das telas ou janelas. Nesta fase o projetista tem uma visão geral e inicial (esboço) das telas ou janelas da interface a ser construída.

Portanto, a partir da árvore MAD* e com o auxílio de regras ergonômicas (por exemplo, na tabela (2) abaixo) são construídas as árvores EDITOR (especificação conceitual parcial da interação).

<p>EXEMPLOS DE REGRAS ERGONÔMICAS PARA CONSTRUÇÃO DA ÁRVORE EDITOR</p> <p>Regra 01: SE a tarefa for de <u>alto nível</u> (MAD*) ENTÃO definir um <i>Editor</i> (Modelo Editor) cuja <i>Apresentação</i> deve ser do tipo <i>Janela</i>.</p> <p>Regra 02: SE a tarefa for de <u>nível intermediário</u> (MAD*) ENTÃO definir um <i>Editor</i> (Modelo Editor) cuja <i>Apresentação</i> deve ser do tipo <i>Sub-janela ou Janela</i>.</p> <p>Regra 03: SE as tarefas forem ligadas pelo construtor <u>OU</u> (MAD*) ENTÃO definir para esse conjunto de tarefas uma <i>Visão</i> (Modelo Editor) cuja <i>Apresentação</i> deve ser do tipo <i>Barra de Menu</i> e cada tarefa deve ser um <i>Objeto de Interação</i> (Modelo Editor) do tipo <i>item-menu</i> dessa <i>Visão</i>.</p>

Quadro 2: Exemplos de Regras para Construção da Árvore EDITOR do Tutorial

De acordo com as regras (1 e 2) acima e com o modelo MAD* da tarefa, é possível constatar que serão 10 árvores EDITOR (ver Apêndice B). Outras regras auxiliam na definição das *Visões* e *Objetos_de_Interação* de cada *Editor*.

O produto gerado nesta etapa é portanto, o conjunto de árvores (parciais) EDITOR. Apresentamos abaixo uma das árvores EDITOR do exemplo (Figura 18) geradas nesta etapa.

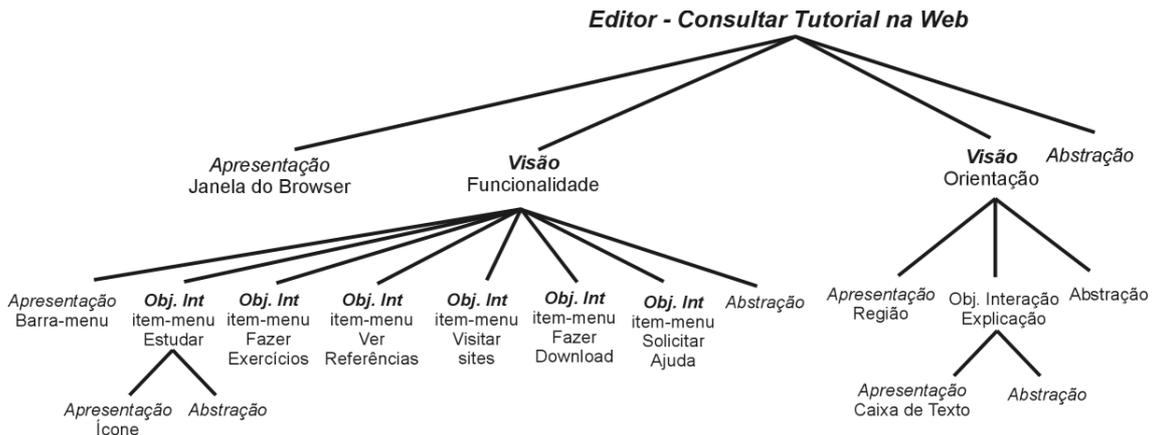


Figura 18: Árvore EDITOR Parcial da tarefa "Consultar Tutorial na Web"

5.3.3 ETAPA 3: Definição dos Atributos (Especificação conceitual total da Interação)

A 3ª etapa consiste na definição dos atributos das árvores EDITOR. Esta etapa tem como entrada a árvore gerada na etapa anterior e a descrição MAD* da tarefa gerada na primeira etapa. Nesta fase ocorre a segunda inserção do conhecimento ergonômico (as Regras Ergonômicas para Definição dos Atributos (Quadro 2)) a árvore EDITOR é complementada, ou seja, são definidos os atributos de cada uma das facetas:

- *Apresentação* (localização, formato, tamanho de fonte, etc.);
- *Abstração* (com relação ao domínio da aplicação) e;
- *Controle* que define o encadeamento do diálogo (inter e intra *Editores*).

EXEMPLOS DE REGRAS ERGONÔMICAS PARA DEFINIÇÃO DOS ATRIBUTOS

Regra 14: SE o número de opções a escolher para concluir ou prosseguir uma determinada tarefa for grande (oito ou mais) (Árvore EDITOR parcial), ENTÃO agrupar logicamente as opções e optar pela orientação vertical. (orientação) (atributos EDITOR).

Regra 15: SE o número de opções a escolher para concluir ou prosseguir uma determinada tarefa for pequeno (sete ou menos) (Árvore EDITOR parcial), ENTÃO pode-se optar pela orientação horizontal. (orientação) (atributos EDITOR).

Quadro 3: Exemplos de Regras para Definição dos Atributos do Tutorial

representados por uma estrutura orientada a objeto, conforme mostrado em [Lul92]. A ferramenta EDITOR OBJETO faria a implementação objeto das árvores EDITOR.

No Apêndice B é apresentado junto à modelagem EDITOR o protótipo (janelas) do Tutorial na Web construído utilizando a metodologia MEDITE.

5.3.5 ETAPA 5: Avaliação

Percebe-se através da Figura (15) acima que após a geração de cada produto pode ocorrer uma atividade de Avaliação. Esta atividade é distribuída em todas as etapas anteriores da metodologia e consiste na avaliação de cada produto de cada etapa.

- **Avaliação da descrição da tarefa MAD***

Nesta fase, a avaliação consiste em verificar junto ao usuário se a árvore e descritores MAD* gerados correspondem à sua lógica de execução da tarefa. Cabe também ao projetista: verificar a completude e coerência da modelagem, eliminar tarefas não informatizáveis, modificar e melhorar as tarefas de acordo com a existência do novo sistema. Não se pode esquecer que o objetivo da concepção não é de replicar a tarefa existente, mas utilizar os objetivos fundamentais como base para a concepção, para em seguida adaptar os procedimentos existentes às novas necessidades [Gam98].

- **Avaliação da árvore (parcial) EDITOR**

Nesta etapa a avaliação consiste em verificar se as regras foram aplicadas corretamente, assim como a coerência e a completude da árvore EDITOR em relação à árvore MAD*. São avaliados o número de *Editores*, *Visões*, *Objetos_de_Interação*.

Nesta etapa é importante que o projetista utilize a sua experiência e bom senso para verificar se as árvores geradas (a partir do modelo MAD* e a aplicação das regras) apresentam coerência com os objetivos da concepção.

- **Avaliação da árvore (completa) EDITOR**

Nesta etapa a avaliação consiste em avaliar os atributos que foram definidos, a coerência entre as árvores EDITOR geradas e as árvores MAD*. O projetista deve verificar também a coerência entre as árvores e às regras utilizadas no processo de transformação. Nesta etapa, o modelo EDITOR já permite uma visualização (esboço) das janelas do sistema que se está projetando, que devem ser levadas ao usuário de forma que este possa participar do processo de concepção, validando dessa forma a descrição.

- **Avaliação do protótipo**

A avaliação do protótipo deve ser realizada junto ao usuário através de técnicas de avaliação (testes de usabilidade) ou ainda através de: inspeção por padrão, avaliação heurística, conformidade com recomendações, exploração cognitiva, abordagem híbrida). Dependendo do tipo de problema que for encontrado o projetista poderá retornar à etapa imediatamente anterior ou mesmo à 1ª etapa. Problemas que dizem respeito à tarefa, necessitam de um retorno à 1ª etapa, por exemplo: se durante a avaliação do protótipo, o usuário sente a necessidade de um outro caminho ou forma de realizar determinada tarefa, isto implica em mudanças na descrição da tarefa (o ideal seria que estes problemas aparecessem na avaliação daquela etapa, daí a importância da participação do usuário). Se durante a avaliação são detectados, por exemplo, problemas de legibilidade ou cores, deve haver um retorno à 3ª etapa, ou seja na etapa de definição desses atributos

5.4 Conclusão

Apresentamos neste capítulo a proposta de uma metodologia para concepção de interfaces ergonômicas, MEDITE. Trata-se de uma abordagem iterativa por permitir retorno às etapas anteriores e incremental pois o produto gerado em cada etapa serve como entrada na próxima etapa. Permite uma concepção tanto descendente (da descrição da tarefa obtêm-se uma representação abstrata parcial inicial da interação) quanto ascendente (a representação é incrementalmente completada com a definição dos atributos).

MEDITE encaixa-se nas abordagens baseadas na tarefa, pois parte da descrição da tarefa do usuário para fazer a especificação da interação. Os processos de MEDITE são simples, bem definidos, produzindo no final de cada um deles um artefato segundo o modelo que conduz àquele processo (MAD*, EDITOR).

MEDITE utiliza o conhecimento ergonômico para auxiliar o projetista na especificação da interface. Aplicando regras o projetista pode relacionar elementos do modelo da tarefa (MAD*) com os elementos da interação (EDITOR), de uma forma mais simplificada devido à objetividade e classificação das regras. Contudo, vale salientar que essas regras servem como um auxílio e apoio para o projetista aplicar a Ergonomia no projeto das interfaces. No entanto, se este projetista tem experiência na área, ou mesmo algum conhecimento sobre Ergonomia, ele deve fazer uma avaliação

própria, verificando os resultados obtidos. Caso este projetista não tenha nenhuma experiência na área de Ergonomia, recomenda-se que apóie-se no uso das regras e leve os modelos e protótipo para apreciação do usuário. A participação do usuário durante o processo de especificação de interfaces é de fundamental importância. MEDITE facilita essa participação devido à utilização de modelos simples em cada uma das etapas.

Todos os processos de MEDITE podem ser realizados manualmente (é o caso do exemplo mostrado neste trabalho), ou seja, é uma metodologia independente de ferramentas computacionais. Evidentemente que dependendo do tamanho do sistema, o trabalho do projetista pode ser bastante cansativo para alcançar os seus objetivos, ou mesmo impraticável pois o projetista pode perder a noção de coerência e completude da descrição.

Verificamos portanto que a proposta de nossa metodologia satisfaz os requisitos listados no Capítulo 2.

Capítulo 6

6 CONCLUSÃO

Apresentamos neste trabalho a proposta de uma metodologia para concepção de interfaces ergonômicas, a metodologia MEDITE. O seu desenvolvimento surgiu de um estudo inicial de diferentes metodologias baseadas na tarefa para a concepção de interfaces. Nesse estudo verificamos algumas dificuldades e problemas em relação ao uso e aplicação dessas metodologias, o que nos fez gerar algumas hipóteses sobre as possíveis causas desses problemas e dificuldades.

Após uma análise mais detalhada de cada uma dessas metodologias, onde foram verificados os modelos utilizados, como é realizada a inserção do conhecimento ergonômico e os processos utilizados para a especificação da interface, constatamos que de fato, essas metodologias sofrem das dificuldades e problemas apontados nas hipóteses.

A partir dessas hipóteses geramos os requisitos necessários para a definição de uma metodologia que superasse tais problemas, assim como os requisitos relacionados especificamente à Ergonomia e aos modelos de interação e de arquitetura.

Em seguida descrevemos cada modelo utilizado na metodologia, os modelos de tarefa, de interação e de arquitetura, definimos a estrutura de uma Base de Regras Ergonômicas, elaboramos um conjunto de regras segundo a estrutura da Base e as aplicamos no exemplo "Um Tutorial na Web". Este exemplo foi utilizado para ilustrar o processo de especificação de uma interface segundo a metodologia (MEDITE) proposta.

Na próxima seção apresentamos uma discussão sobre os resultados obtidos.

6.1 Discussão dos Resultados

MEDITE é uma metodologia para concepção de interfaces ergonômicas orientada a modelos, ou seja, cada uma de suas etapas é guiada por um modelo bem definido (MAD* e EDITOR) produzindo no final da etapa um artefato. MEDITE se insere na abordagem baseada na tarefa, ou seja, parte da descrição da tarefa do usuário para dar início à especificação da interface.

O processo de especificação definido em MEDITE é iterativo, de forma que cada etapa permite um retorno às etapas precedentes com o objetivo de avaliar os produtos gerados, e incremental pois cada produto serve como entrada para a etapa seguinte. Apresenta uma especificação do tipo descendente (da descrição da tarefa obtêm-se uma representação abstrata parcial inicial da interação) e ascendente (a representação é incrementalmente completada com a definição dos atributos).

MEDITE permite uma especificação ergonômica da interação, ou seja, insere o conhecimento ergonômico ainda nas etapas de construção da interface conceitual, através de regras extraídas de recomendações, padrões, critérios que auxiliam ao projetista na transformação de um modelo ao outro.

Os modelos utilizados em MEDITE são: MAD* para a descrição da tarefa e o modelo EDITOR como modelo unificado de interação e de arquitetura. O modelo MAD* permite descrever a tarefa do usuário de forma completa e coerente e incorpora em sua descrição atributos importantes no contexto de nosso trabalho: os atributos orientados à especificação da interface.

O Modelo EDITOR utilizado como modelo unificado de interação e de arquitetura é um dos principais diferenciais de MEDITE em relação às outras metodologias. Como modelo de interação o Modelo EDITOR proporciona uma descrição gráfica clara e intuitiva da interface ao projetista: a organização multi-agente hierárquica da interface reflete a composição visual de sua imagem e a natureza de seus elementos [Lul92].

Como modelo de arquitetura, o EDITOR permite uma decomposição uniforme (*Apresentação, Abstração e Controle*) em todos os níveis de definição da interface, refletindo fielmente a estrutura do modelo da interação, e tem no modelo de programação orientado a objeto uma ferramenta natural de realização. Ou seja, a estrutura do código do protótipo gerado guarda uma relação de isomorfismo com a estrutura da interface conceitual, o que facilitará para o projetista da interface possíveis

modificações no protótipo, e pode ser implementada por qualquer ferramenta (linguagem) de desenvolvimento de software orientado a objeto.

Através da Base de Regras Ergonômicas definida para a metodologia MEDITE, o projetista pode especificar a interação de uma forma mais simples. A classificação das regras segue o processo de construção da interação (modelo EDITOR), portanto o projetista tem em mãos um mecanismo / procedimento fundamentado em recomendações / critérios ergonômicos que traduz os elementos da tarefa em elementos da interação. A padronização da estrutura da regra permite também uma fácil identificação desses atributos e elementos.

Contudo, apesar do processo definido na metodologia MEDITE ser um processo ergonômico, ou seja, leva em conta as necessidades, objetivos e características do usuário e as recomendações/critérios ergonômicos ainda no processo de especificação, é necessário que os produtos (descrições da tarefa, da interação, etc..) sejam avaliados / verificados, seja pelo próprio projetista (se tiver alguma experiência ou conhecimento em Ergonomia) e / ou pelo usuário. Por tratar-se de modelos simples de serem compreendidos, que permitem uma visualização da estrutura / apresentação da interface que se está construindo, estes podem ser avaliados diretamente com os usuários, estreitando assim a comunicação destes com os projetistas. A participação do usuário no processo deve ser facilitada, pois ele é quem melhor entende a sua tarefa.

Enfim, a existência de um modelo conceitual de interação simples (EDITOR), explícito e intuitivo, cuja estrutura espelha a estrutura da tarefa, a existência de um modelo arquitetural da interface (EDITOR) fiel ao modelo de interação e a objetividade das regras ergonômicas, permite ao projetista um rápido e fácil entendimento e utilização do método.

Em relação às metodologias apresentadas no Capítulo2, MEDITE procura evitar os problemas e dificuldades citados nas hipóteses, da seguinte forma:

- MEDITE apresenta um mecanismo fundamentado em critérios ergonômicos, que é utilizado pelo projetista ainda nas etapas de especificação. Na metodologia MCI esse conhecimento é utilizado, porém, durante a avaliação / verificação / validação das especificações obtidas em cada uma de suas etapas e não durante o processo de construção dessas especificações;
- A dificuldade em selecionar e aplicar regras encontrada nas metodologias ERGO-START e ALACIE, é minimizado em MEDITE pela presença de uma Base de Regras Ergonômicas cuja classificação é fiel às etapas de

especificação da interação (construção da árvore EDITOR e definição de seus atributos) e devido à padronização da estrutura das regras, evidenciando os atributos importantes;

- A dificuldade de passagem da tarefa para a interação detectado nas metodologias ADEPT, ERGO-START, MCI e ALACIE foi minimizado devido à utilização de um modelo de interação explícito e simples. As demais metodologias ou não apresentavam um modelo explícito (ADEPT), ou o modelo era bastante complexo (ERGO-START) ou não proporcionavam uma visualização da interface (MCI e ALACIE);
- Por se tratar de uma metodologia orientada a modelos, ela independe de ferramentas, ou seja, ela pode ser utilizada num processo manual de especificação de interfaces, o que na maioria das metodologias citadas era praticamente impossível, pois várias de suas etapas não eram definidas por um modelo, mas por uma ferramenta computacional. Se essa ferramenta não estivesse disponível, o uso da metodologia se tornava inviável mesmo para sistemas simples.

As vantagens da utilização de MEDITE estão centradas no uso de modelos conceituais para a compreensão de processos e produção de artefatos. Essa afirmação é fortalecida pelos diversos estudos da psicologia cognitiva que têm mostrado o papel fundamental do uso de modelos para o aprendizado, para a concepção e para a operação de dispositivos interativos complexos ([Ras80] [Pfa85] [Fal89]).

Outras metodologias têm sido desenvolvidas, como por exemplo MACIA (Metodologia de Assistência à Concepção e à realização de Interfaces Adaptativas) ([Fur97] [Fur99]) que se enquadra nos requisitos que elaboramos e listamos no Capítulo 2. MACIA é uma metodologia que descreve um processo para desenvolvimento de sistemas interativos de supervisão, onde as interfaces são geradas automaticamente a partir do modelo da tarefa integrando fatores humanos no processo.

MACIA parte da análise da tarefa, e em cada uma de suas etapas utiliza um modelo (de tarefa, de interação e de arquitetura). Utiliza o conhecimento ergonômico sob a forma de regras de produção que são classificadas de acordo com as etapas de especificação, assim como MEDITE. Os estágios de construção são vários, o que faz com que vários tipos de regras sejam definidas.

O principal diferencial entre MEDITE e MACIA é que, em MACIA são definidos 3 modelos (para a descrição das tarefas (extensão de MAD), para a

especificação da interação (MIC) e um outro para arquitetura (Modelo de Agentes)), enquanto que, em MEDITE são adotados 2 modelos (uma para descrição das tarefas (MAD*) e um modelo unificado de interação e de arquitetura (o Modelo EDITOR)).

MEDITE é uma metodologia que busca simplificar para o projetista o processo de concepção de interfaces com alto grau de usabilidade. Reduzindo o número de modelos e estágios de modelagem, procura facilitar, para o projetista, o entendimento, compreensão e utilização da metodologia. Através das etapas de avaliação e o uso de modelos simples e intuitivos, MEDITE facilita a comunicação entre projetistas e usuários, e torna possível a participação deste último durante o processo e não apenas na avaliação do protótipo.

Enfim, este trabalho alerta para uma relação de suma importância que deve ser levada em conta na concepção de sistemas. A relação entre a Ergonomia e a Informática. Torna-se fundamental portanto melhorar a formação em ambos os lados: os ergonomistas devem aprender como os sistemas são desenvolvidos, os projetistas devem conhecer as exigências dos sistemas que interagem com os humanos.

6.2 Propostas de Continuidade

Após a finalização deste trabalho apresentamos algumas propostas de continuidade:

- **Melhorar a metodologia em termos de especificação da navegação/diálogo.**

MEDITE define bem os modelos utilizados em cada etapa, trata-se de uma abordagem orientada a modelos. No entanto o modelo de navegação (diálogo) não é explícito, ou seja, não apresenta uma representação explícita das mudanças ocorridas na interface durante a interação. Algumas metodologias como MCI e MACIA especificam o diálogo utilizando Redes de Petri e diagramas de estados. Tais possibilidades poderiam vir a ser incorporadas na metodologia de forma a tornar explícito o modelo de navegação.

- **Validar a metodologia**

MEDITE não foi ainda validada. No momento atual ela vem sendo utilizada em dois trabalhos: na concepção da interface de um simulador de Redes de Computadores e na concepção da interface de uma ferramenta para análise e descrição da tarefa. Após a

finalização desses trabalhos seria interessante fazer uma análise junto aos projetistas e verificar como foi a utilização da metodologia e a avaliação da interface gerada.

- **Ferramentas para suporte à metodologia**

Uma das preocupações de MEDITE é a definição de uma abordagem baseada em modelos independente de ferramentas, devido ao fato de que se essas ferramentas não estiverem disponíveis impossibilitaria o uso da metodologia. No entanto, isso não implica em que não haja ferramentas para auxiliar o processo ou parte dele. Métodos e ferramentas já existentes na área poderiam ser incorporadas à metodologia de forma a facilitar o processo de especificação de interfaces. Nesse sentido, o Grupo de Interface Homem-Máquina (GIHM), vem trabalhando na concepção de uma ferramenta para a análise e modelagem da tarefa e prevê a concepção de uma ferramenta para auxiliar a modelagem da interação (implementação do Modelo EDITOR).

- **Regras ergonômicas para contextos específicos**

O conhecimento ergonômico utilizado por MEDITE é ainda bastante geral, ou seja, não define nenhum contexto específico de utilização. Poderiam ser elaboradas a partir do conhecimento ergonômico existente, bases de regras ergonômicas específicas à contextos específicos, por exemplo: ambientes para automação industrial, educação à distância, etc.

REFERÊNCIAS BIBLIOGRÁFICAS

- [Alo96] ALONSO, B. *Analyse des tâches mono et multi-opérateurs aériens* (Rapport convention INRIA / CENA C. 0007). Rocquencourt, France: INRIA, Projet PSYCHO-ERGO, 1996.
- [BC93] BASTIEN, J. M. C., SCAPIN, D. L., *Ergonomic Criteria for the Evaluation of Human-Computer Interfaces*. Tech. Rep. N. 156. Rocquencourt, France: INRIA, 1993.
- [BH94] BODART, F., HENNENBERT, A-M., et al., *A model-based approach to presentation: a continuum from task analysis to prototype*. Proceedings of the Eurographics Workshop on Design, Specification and Verification of Interactive Systems, Bocca di Magra: Eurographics Series, 1994.
- [BH95] BODART, F., HENNENBERT, A-M., et al., *Towards a systematic building of software architecture: the TRIDENT methodological guide*. In Palanque, P. & Bastide, R. (Eds.), Proceedings of the Eurographics Workshop on Design, Specification and Verification of Interactive Systems DSV-IS'95, Toulouse, France, 1995.
- [Boe81] BOEHM, B. W. *Software Engineering Economics*. New Jersey: Prentice Hall Inc, 1981.
- [Boe88] BOEHM, B. W., *A Spiral Model of Software Development and Enhancement*. IEEE Computer, 1988.
- [Cou87] COUTAZ, J. *The Construction of User Interfaces and the Object Paradigm*, The European Conference on Object Oriented Programming, ECOOP'87, Paris, França, 1987.
- [Cou90] COUTAZ, J. *Interfaces Homme-Ordinateur – Conception et Réalisation*, Bordas, Paris, 1990.
- [Cyb94] CYBIS, W. A. *A identificação de Objetos de Interfaces Homem-Computador e de seus Atributos Ergonômicos*. Tese de Doutorado. UFSC, Florianópolis, 1994.
- [Cyb96] CYBIS, W. A. *Ergonomia e Usabilidade de Software – Abordagem Ergonômica para IHC*. Florianópolis, 1996.

- [Fal89] FALZON, P., *Ergonomie Cognitive du Dialogue*, Presses Universitaires de Grenoble, France, 1989.
- [Fur97] FURTADO, M. E. S., *Mise en oeuvre d'une méthode de conception d'interfaces adaptatives pour des systèmes de supervision à partir de spécifications conceptuelles*, PhD thesis, Doctorat de productique et Informatique à l'Université Aix Marseille III, France, 1997.
- [Fur99] FURTADO, M. E. S., *Integrando Fatores Humanos no Processo de Desenvolvimento de Interfaces Homem-Computador Adaptativas*, II Workshop sobre Fatores Humanos em Sistemas Computacionais, Campinas, SP, Brasil, 1999.
- [Gam98] GAMBOA, F. R. *Spécification et Implémentation d'ALACIE: Atelier Logiciel d'Aide à la Conception d'Interfaces Ergonomiques*, Thèse de Doctorat, Paris XI, Octobre, 1998.
- [GL01] GUERRERO, C. V. S., LULA, B. Jr., *Um Tutorial na Web: obtenção do modelo da interação a partir do modelo da tarefa com o auxílio de regras ergonômicas*. Relatório Técnico RT- DSC-002/2001, (55 p.), UFPB/DSC/CCT Campina Grande, PB, outubro de 2001.
- [GS97] GAMBOA, F. & SCAPIN, D., *Editing MAD Task Descriptions for Specifying User Interfaces at Both Semantic and Presentation Levels*, in Proceedings DVS-IS'97, 4th International Eurographics Workshop on Design, Specification, and Verification of Interactive Systems, Granada Spain, 1997.
- [Haa00] HAAN, G. *ETAG - A Formal Model of Competence Knowledge for User Interface Design*, Ph.D. Thesis, Vrije Universiteit, Amsterdam, October, 2000.
- [Ham95] HAMMOUCHE, H. *De la modélisation des tâches utilisateurs au prototype de l'interface homme-machine*, Thèse de Docteur, Université Paris VI, Décembre, 1995.
- [HH93] HIX, D., HARTSON, H. R., *Developing User Interfaces, Ensuring Usability Through Product & Process*, John Willey & Sons, Inc., 1993.
- [HVV92] HAAN G., VAN der VEER G. C. e VAN VLIET J.C., *Formal Modelling Techniques in Human-Computer Interaction*. Acta Psychologica, 78, nos. 1-3, 26-76, North-Holland, Amsterdam, 1992.
- [ISO93] ISO 9241 *Ergonomic requirements for office work with visual display terminals*; International Standard ISO 9241 (Partes 10, 12, 14), 1993.

- [JJW88] JOHNSON, P., JOHNSON, H., WADDINGTON R. and SHOULS A. *Task-Related Knowledge Structures: Analysis, Modelling and Application*, Queen Mary College, University of London 1988.
- [JWMP93] JOHNSON, P., WILSON, S., MARKOPOULOS, P. & PYCOCK, J. ADEPT – *Advanced Design Environment for Prototyping with Task Models*, INTERCHI'93 Conference Proceedings, Amsterdam: ACM, 1993.
- [Kaf00] KAFURE, I. M. *Validação do Formalismo TAOS para a Concepção de Interfaces Homem-Computador*, Dissertação de Mestrado - COPIN, Universidade Federal da Paraíba, Campina Grande, PB, 2000.
- [Lul92] LULA, B. Jr. *Elaboration d'un Environnement de Génération Interactive d'Interfaces à Manipulation Directe pour le Language OBJLOG*, Thèse de Docteur, Université de Droit d'Economie et des Sciences d'Aix-Marseille III, Faculté des Sciences et Techniques de Saint – Jérôme, França, 1992.
- [Mat95] MATIAS, M., *Checklist: uma ferramenta de suporte à avaliação ergonômica de interfaces*. Dissertação de Mestrado, UFSC, Florianópolis, 1995
- [Med95] MEDEIROS, H. e ROUSSELOT F., *Un Outil D'Aide à la Modélisation de Concepts Dynamiques: Le Système TAME*; Journées Acquisition - Validation - Apprentissage, JAVA'95, 04/95, Grenoble, França, 1995.
- [MH96] MENDES, J. R., Ricardo e HEINECK, Luiz F. M. *Ensino e informação tecnológica na INTERNET*, XXIV Congresso Brasileiro de Ensino de Engenharia - COBENGE/96. Manaus, AM. Outubro, 1996.
- [MKL00] MEDEIROS, H., KAFURE, I. M e LULA, B. Jr., *TAOS: a Task-and Action Oriented Framework for User's Task Analysis in the Context of Human-Computer Interfaces*, Proceeding of SCCC 2000 – XX International Conference of the Chilean Computer Science Society, november 2000, Santiago, Chile, 2000.
- [Mor81] MORAN, T., *The Command Language Grammar, A Representation for the User Interface of Interactive Computer Systems*, International Journal of Man-Machine Studies, Vol. 15, pp. 3-50, 1981.
- [MPW92] MARKOPOULOS, P., PYCOCK, J., WILSON, S., *ADEPT - A task based design environment*, Queen Mary and Westfield College, UK, 1992.
- [Nan90] NANARD, J., *La Manipulation Directe en Interface Homme-Machine*, Thèse d'état, Université des Sciences e Techniques du Languedoc, décembre, 1990.

- [Nor83] NORMAN D. A., *Some Observations on Mental Models*. Mental Models, 7-14. Editado por Dedre Gentner & Albert L. Stevens, Lawrence Erlbaum Associates, Hillsdale, New Jersey. 1983.
- [Nor86] NORMAN, D. A., *Cognitive Engineering*, User Centered Systems Design (Ed. Norman, D. A. et Draper, S.) Lawrence Erlbaum Associates Ltd., 1986.
- [Par97] PARIZOTTO, R., *Guia de Estilos para Serviços de Informação em Ciência e Tecnologia*, Florianópolis, SC, novembro de 1997.
- [Pfa85] PFAFF, G. R. *User Interface Management Systems*, Eurographics Seminars, Springer-Verlag, 224 pages, 1985.
- [Que01] QUEIROZ, J. E. R., *Abordagem Híbrida para Avaliação da Usabilidade de Interfaces com o Usuário*, Tese de Doutorado, Coordenação de Pós-Graduação em Engenharia Elétrica - COPELE, UFPB, Campina Grande, junho de 2001.
- [Ras80] RASMUSSEN, J., *The Human as a Systems Component*, Human Interaction with Computers (Ed. Smith, H. T. et Green, T. R. G.), Academic Press, 1980.
- [Rob96] ROBERT, J. M. *Problèmes liés à la tâche dans la conception des interfaces humains-ordinateur*, Ecole Polytechnique de Montréal. WORKSHOP Ciências Cognitivas e a Concepção de Sistemas de Informação, Florianópolis – SC, fevereiro de 1996.
- [SBY87] SMITH R. G., BARTH, P. S. et YOUNG, R. L., *A substrate for Object-Oriented Interface Design*, Research Direction in Object-Oriented Programming (Ed. Shriver&Wegner), Computer Systems Serie, MIT Press, 1987.
- [Sca86] SCAPIN, D. L., *Guide Ergonomique de Conception des Interfaces Homme-Machine*, Rapports Techniques, Unité de Recherche INRIA - Rocquencourt, France, 1986.
- [Sca88] SCAPIN, D. L., *Vers des outils formels de description des tâches orientées conception d'interfaces*, Rapports de Recherche, Unité de Recherche, INRIA, Rocquencourt, France, 1988.
- [Seb92] SEBILLOTE, S., *Task analysis and formalization according to MAD: Hierarchical task analysis, method of data gathering and examples of task description*, Amsterdam 1992.

- [Seb95] SEBILLOTE, S., *Methodology Guide to Task Analysis with the Goal of Extracting Relevant Characteristics for Human-Computer Interfaces*, International Journal of Human-Computer Interaction, Le Chesnay Cedex, France, 1995.
- [Shn98] SHNEIDERMAN, B., *Designing the User Interface – Strategies for effective Humman-Computer Interaction*, Addison Wesley Publishing Co., 3rd Edition, 1998.
- [SM86] SMITH, S. L. and MOSIER, J. N., *Guidelines For Designing User Interface Software*, The MITRE Corporation Bedford, Massachusetts, USA, 1986.
- [Sou99] SOUSA, M. R. F. de, *Avaliação Iterativa da Especificação de Interfaces com Ênfase na Navegação*, Tese de Doutorado, Coordenação de Pós-Graduação em Engenharia Elétrica - COPELE, UFPB, Campina Grande, dezembro de 1999.
- [SP89] SCAPIN, D. L and PIERRET-GOLKBREICH, C., *Towards a Method for Task Description: MAD*, Unité de Recherche, INRIA, Rocquencourt, France, 1989.
- [Tau88] TAUBER, M. J., *On Mental Models and the User Interface*. In: van der Veer, G. C., Green, T. R. G., Hoc, J. M. and Murray, D. M. *Working with Computers: theory versus outcome*. Academic Press, London. Pp. 89-119, 1988.
- [Tau90] TAUBER, M. J., *ETAG: Extended Task Action Grammar - a language for the description of the user's task language*. In: Diaper, D., Gilmore, D., Cockton, G. and Shackle, B. *Proceedings Interact'90*, pp. 163 - 168. Elseviers, North-Holland, Amsterdam, 1990.
- [Val00] VALIATI. E. R. A., *Guia de Recomendações para o desenvolvimento de interfaces com usabilidade em Softwares Educacionais do tipo hipertexto/hipermídia Informativo*, Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, 2000.
- [Win96] WINOGRAD, T., *Bringing Design to Software*. New York, 1996.
- [WJKCM93] WILSON, S. JOHNSON, P., KELLY, C., CUNNHINGAM, J., MARKOPOULOUS, P., *Beyond Hacking: a Model Based Approach to User Interface Design*. In *Proceedings of HCI'93*, J. Alty, D. Diaper and S. Guest (eds), Cambridge University Press, 1993.

APÊNDICE A: Modelagem MAD* da tarefa exemplo

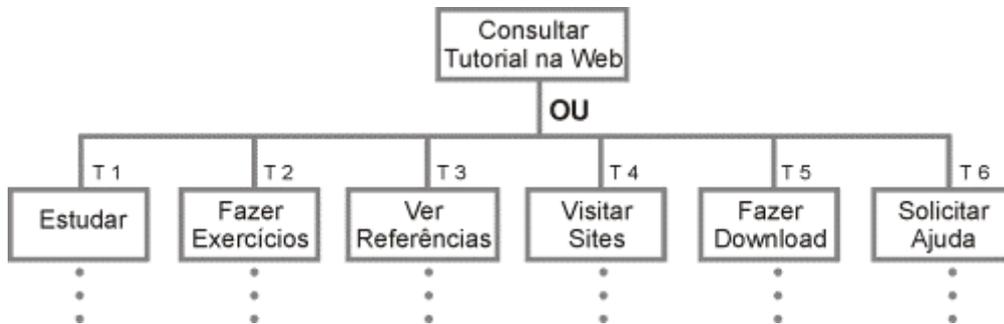


Figura 21: Tarefa-Raiz Consultar Tutorial na Web



Figura 22: Tarefa Estudar



Figura 23: Tarefa Fazer Exercícios



Figura 24: Tarefa Ver Referências

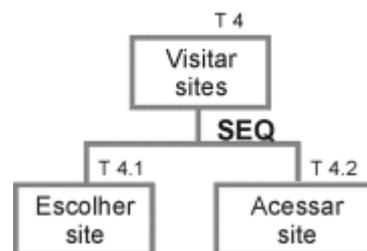


Figura 25: Tarefa Visitar Sites



Figura 26: Tarefa Fazer Download

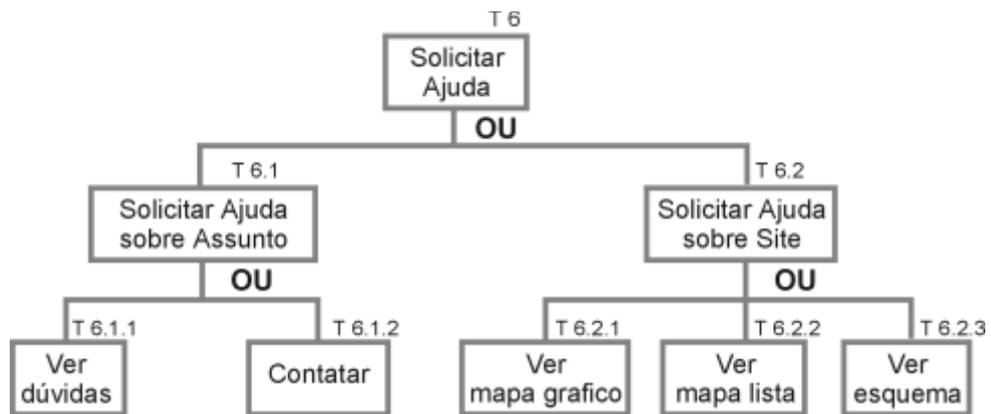


Figura 27: Tarefa Solicitar Ajuda

Regras utilizadas na passagem da descrição MAD* à descrição EDITOR: 2, 4, 5, 7, 9, 11, 14, 16, 17, 18, 20, 23, 24, 25, 26, 27.

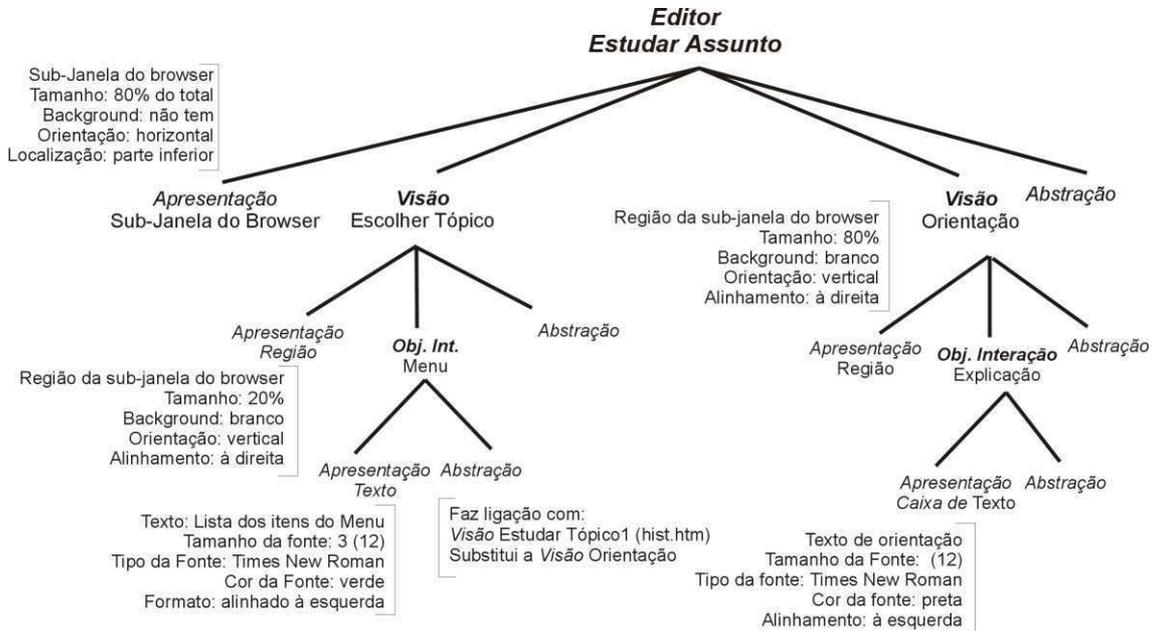


Figura 32: Árvore EDITOR completa da tarefa "Estudar Assunto"

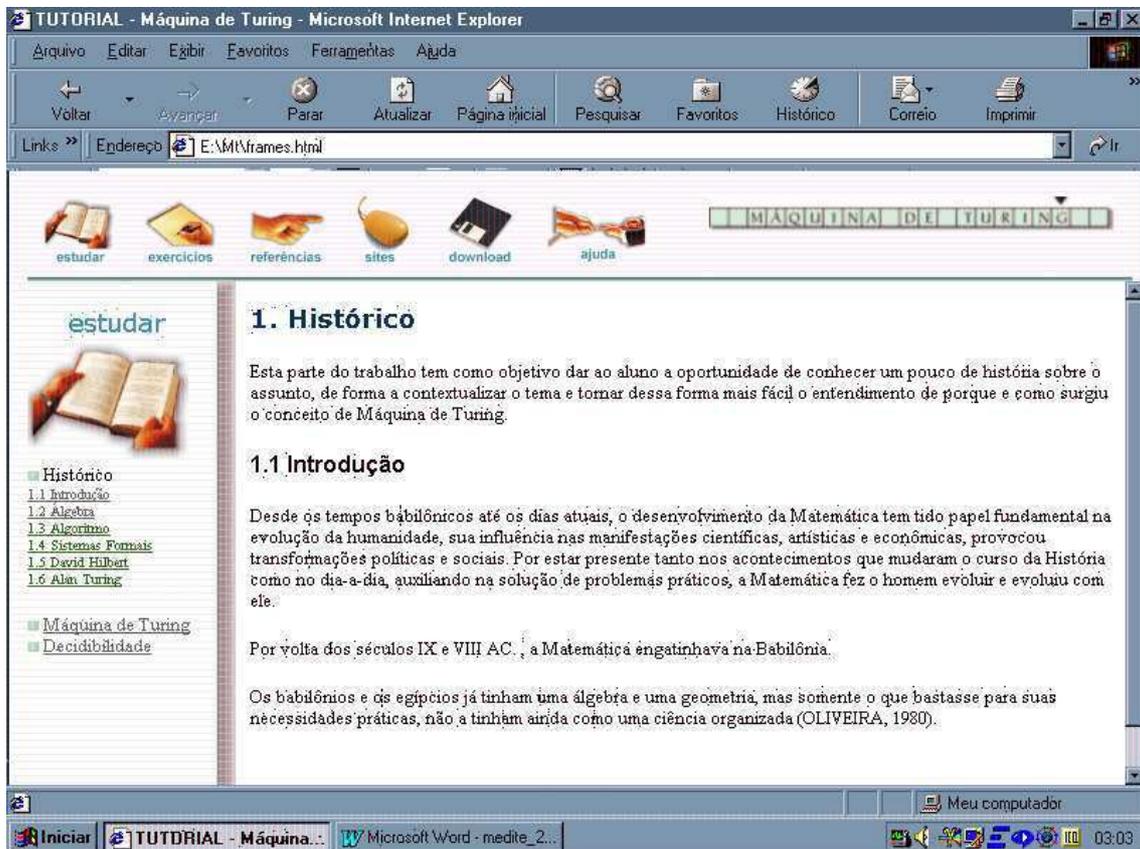


Figura 33: Protótipo janela "Estudar Assunto"

Regras utilizadas na passagem da descrição MAD* à descrição EDITOR: 2, 4, 5, 7, 9, 11, 14, 16, 17, 18, 20, 23, 24, 25, 26, 27.

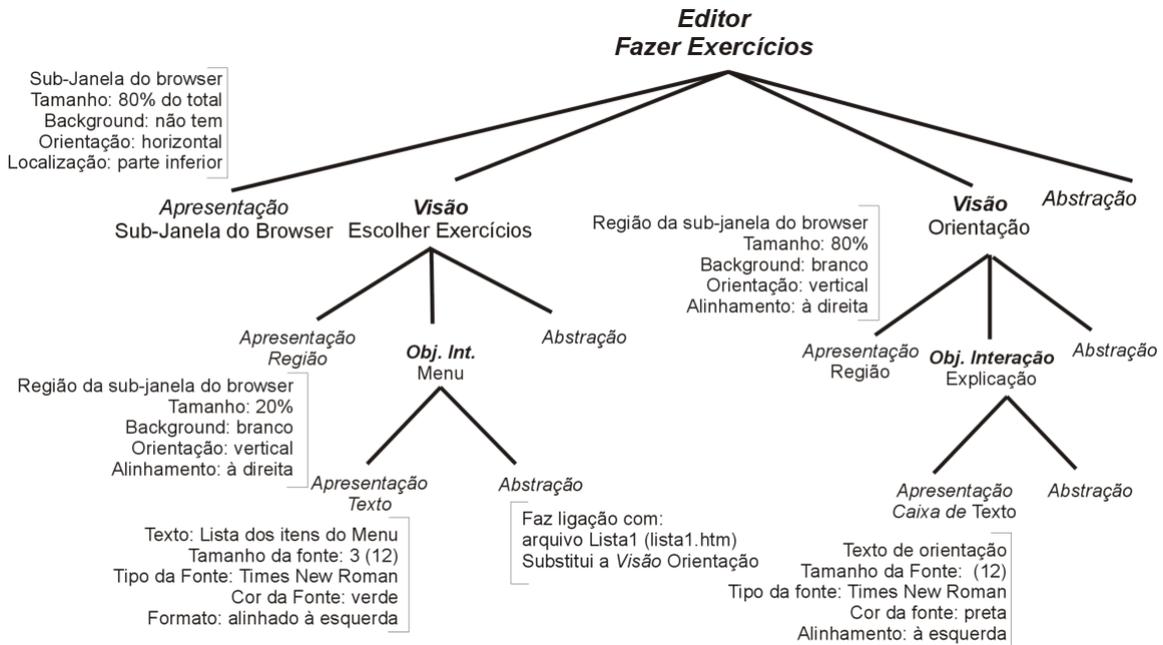


Figura 34: Árvore EDITOR completa da tarefa "Fazer Exercícios"

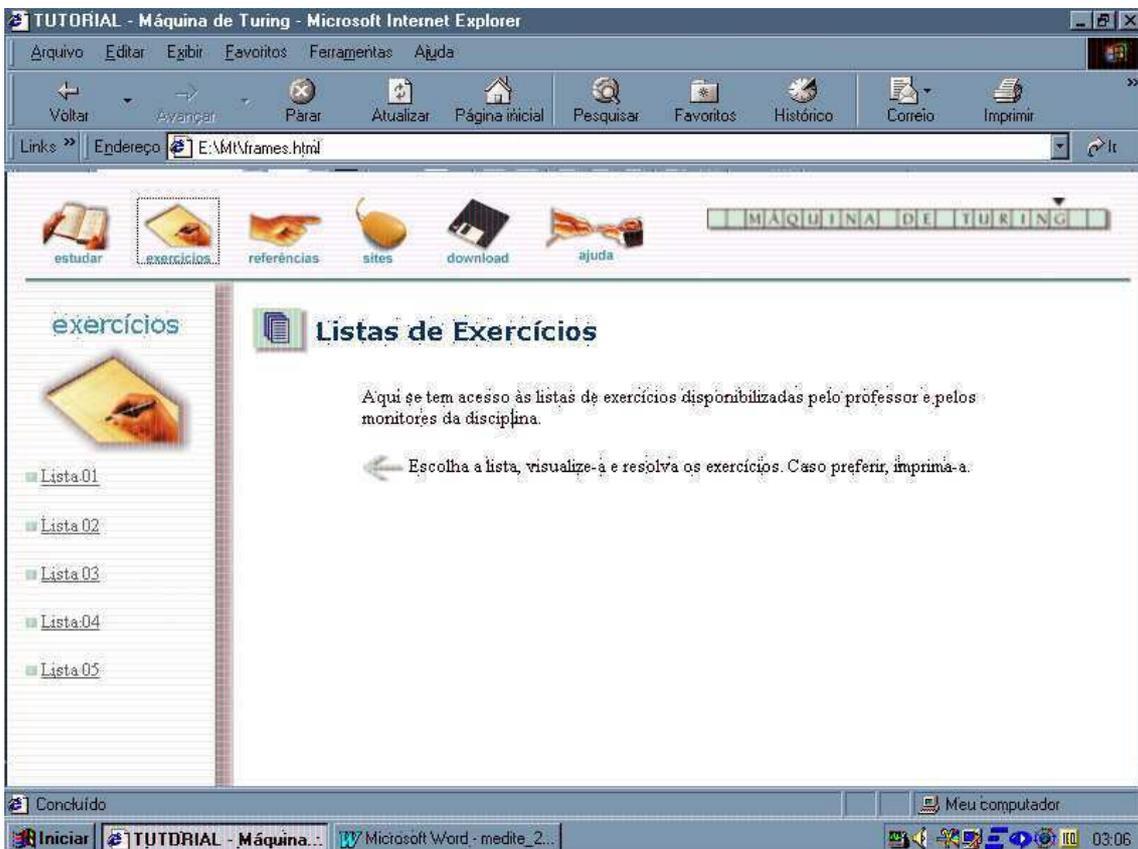


Figura 35: Protótipo janela "Fazer Exercícios"

Regras utilizadas na passagem da descrição MAD* à descrição EDITOR: 2, 4, 5, 6, 9, 11, 12, 16, 17, 18, 20, 23, 24, 25, 26, 27.

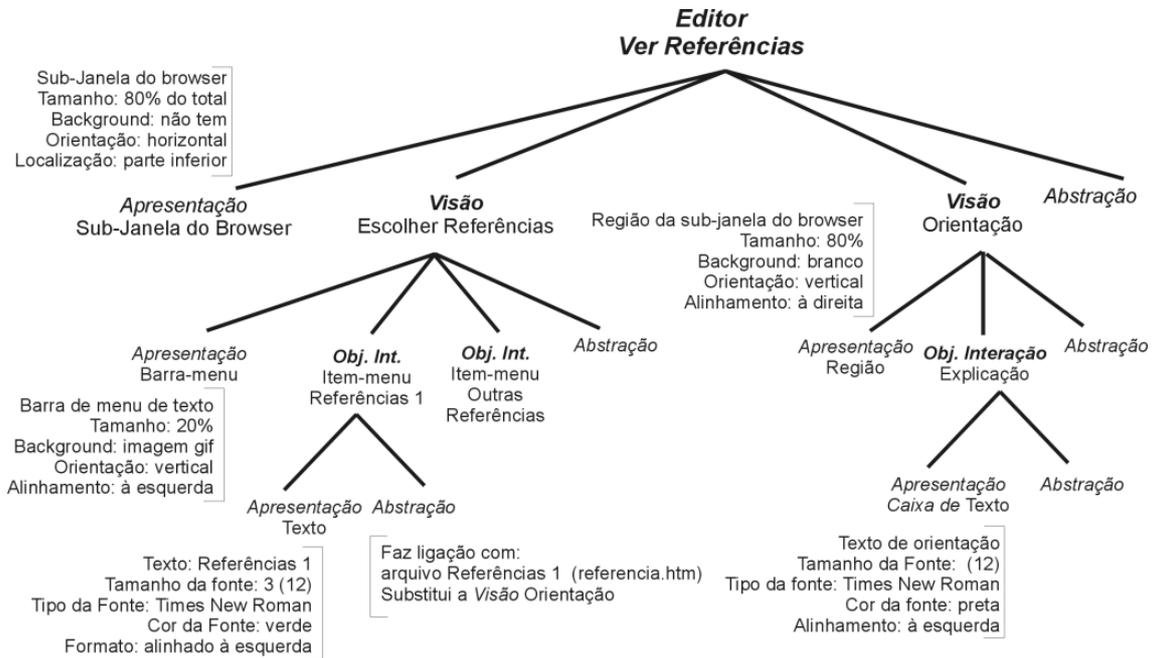


Figura 36: Árvore EDITOR completa da tarefa "Ver Referências"

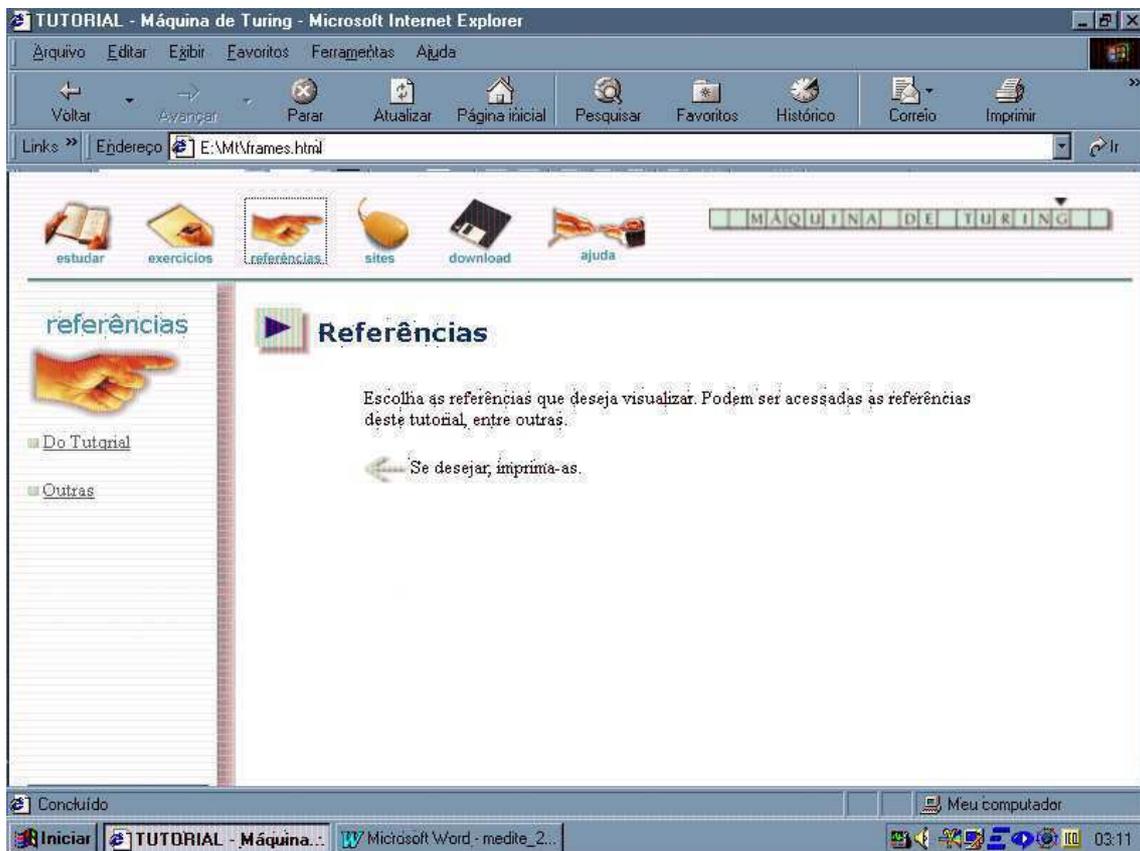


Figura 37: Protótipo janela "Ver Referências"

Regras utilizadas na passagem da descrição MAD* à descrição EDITOR: 2, 4, 5, 6, 9, 11, 12, 16, 17, 18, 20, 23, 24, 25, 26,27.

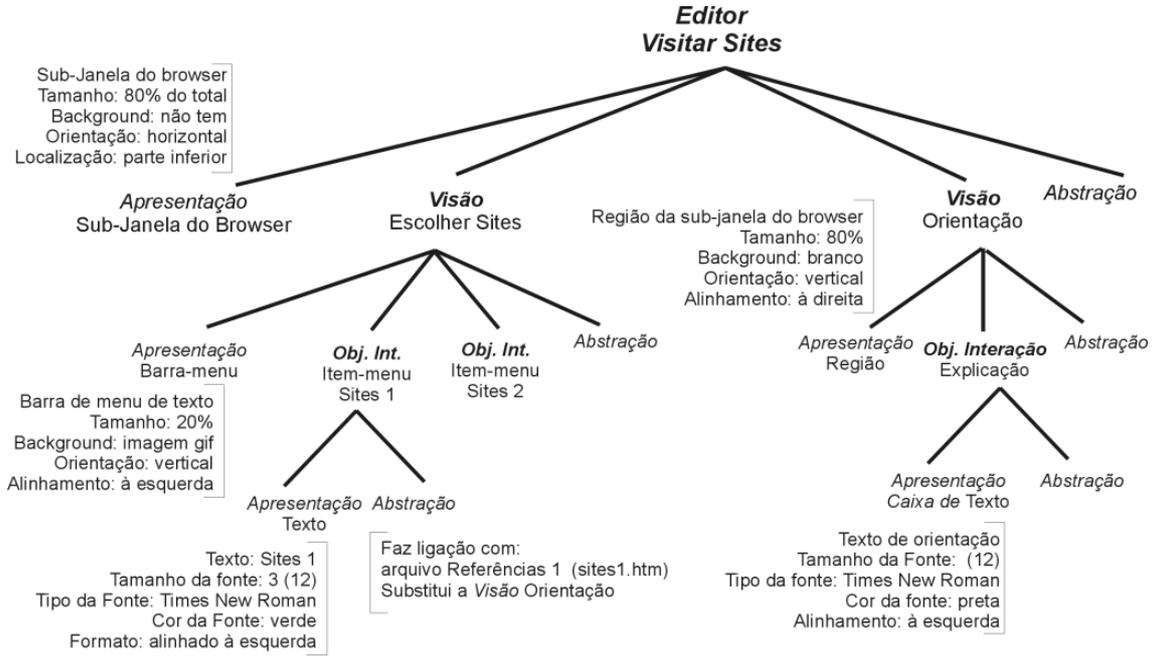


Figura 38: Árvore EDITOR completa da tarefa "Visitar Sites"

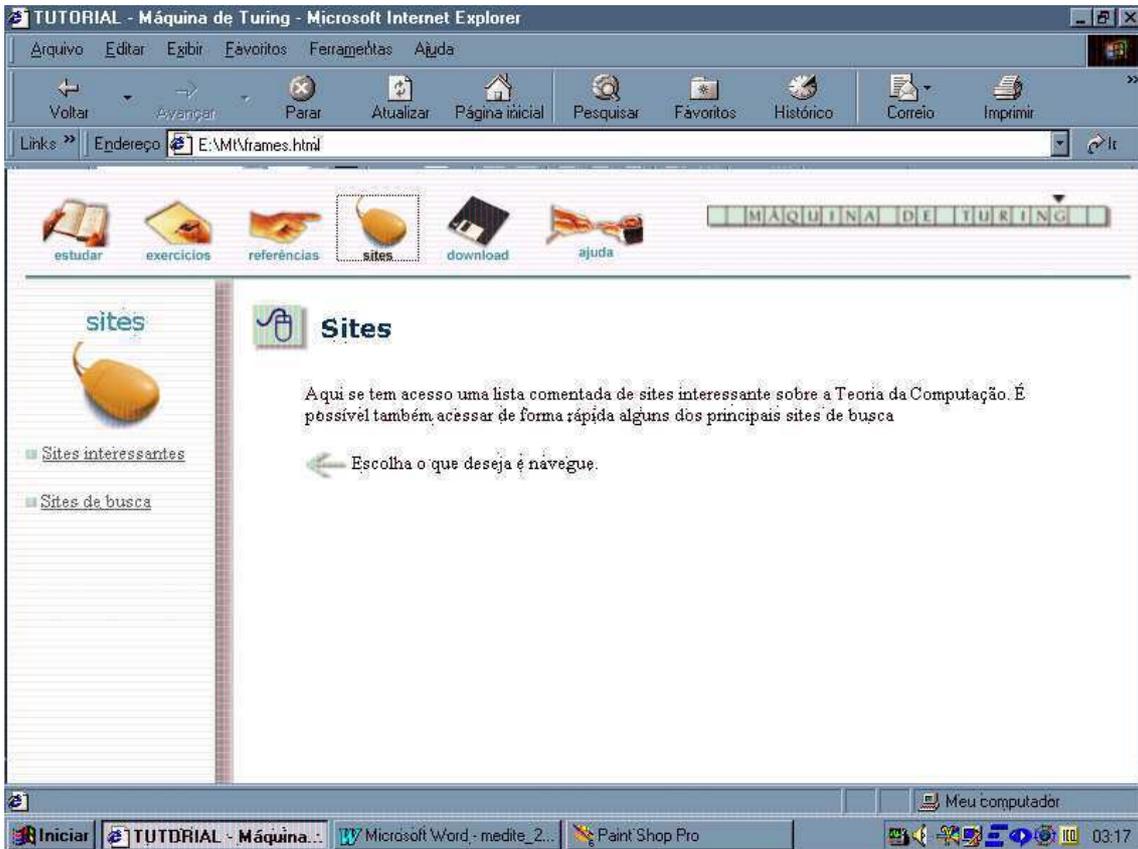


Figura 39: Protótipo janela "Visitar Sites"

Regras utilizadas na passagem da descrição MAD* à descrição EDITOR: 2, 4, 5, 6, 9, 11, 12, 16, 17, 18, 20, 23, 24, 25, 26,27.

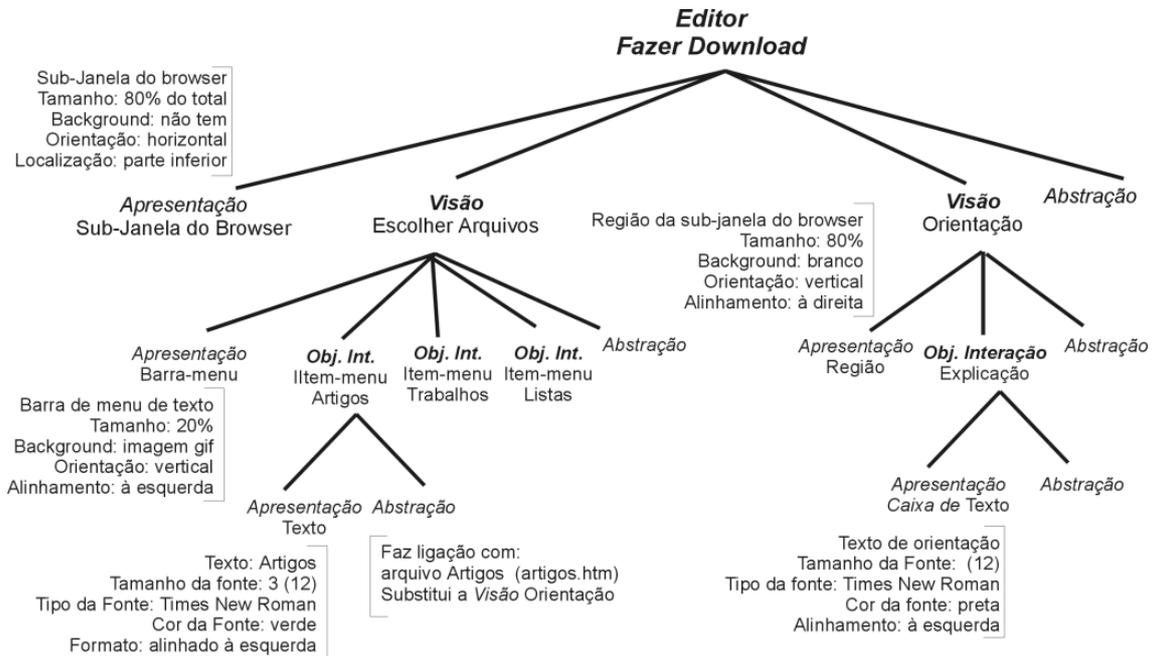


Figura 40: Árvore EDITOR completa da tarefa "Fazer Download"

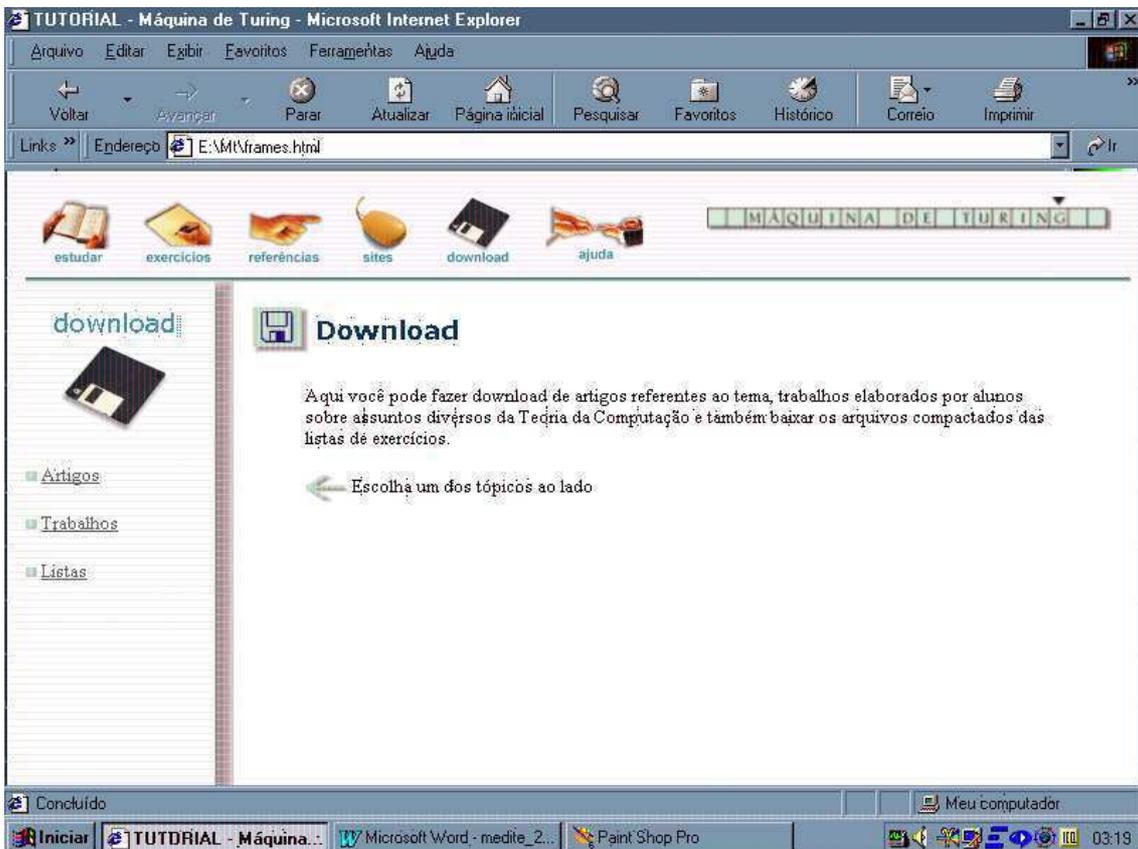


Figura 41: Protótipo janela "Fazer Download"

Regras utilizadas na passagem da descrição MAD* à descrição EDITOR: 2, 4, 5, 6, 9, 11, 12, 16, 17, 18, 20, 23, 24, 25, 26,27.

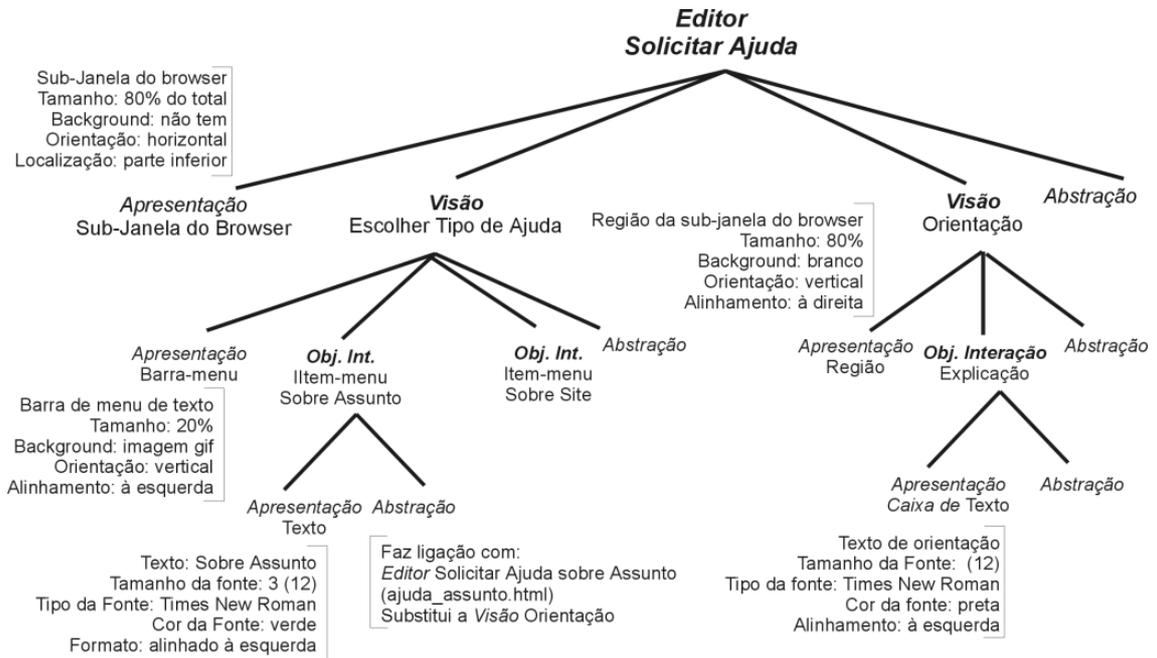


Figura 42: Árvore EDITOR completa da tarefa "Solicitar Ajuda"

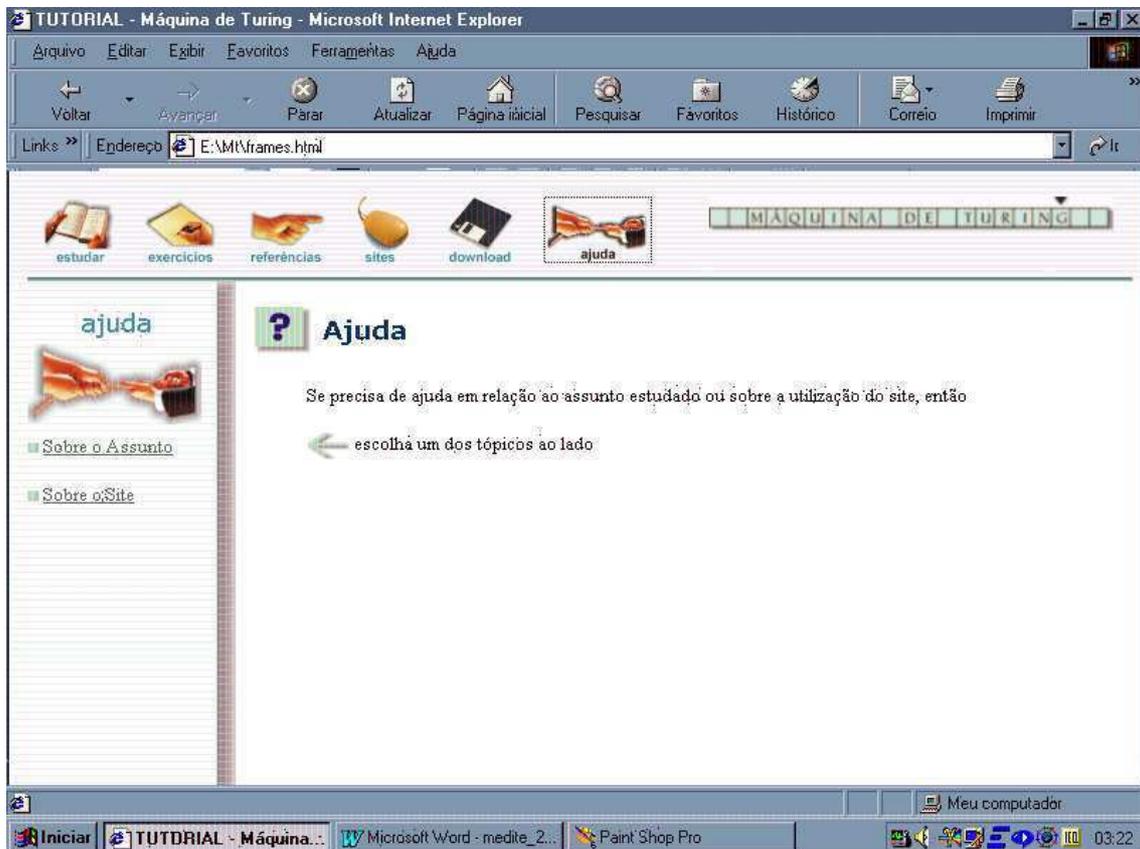


Figura 43: Protótipo janela "Solicitar Ajuda"

Regras utilizadas na passagem da descrição MAD* à descrição EDITOR: 2, 4, 5, 6, 9, 11, 12, 16, 17, 18, 20, 23, 24, 25, 26,27.

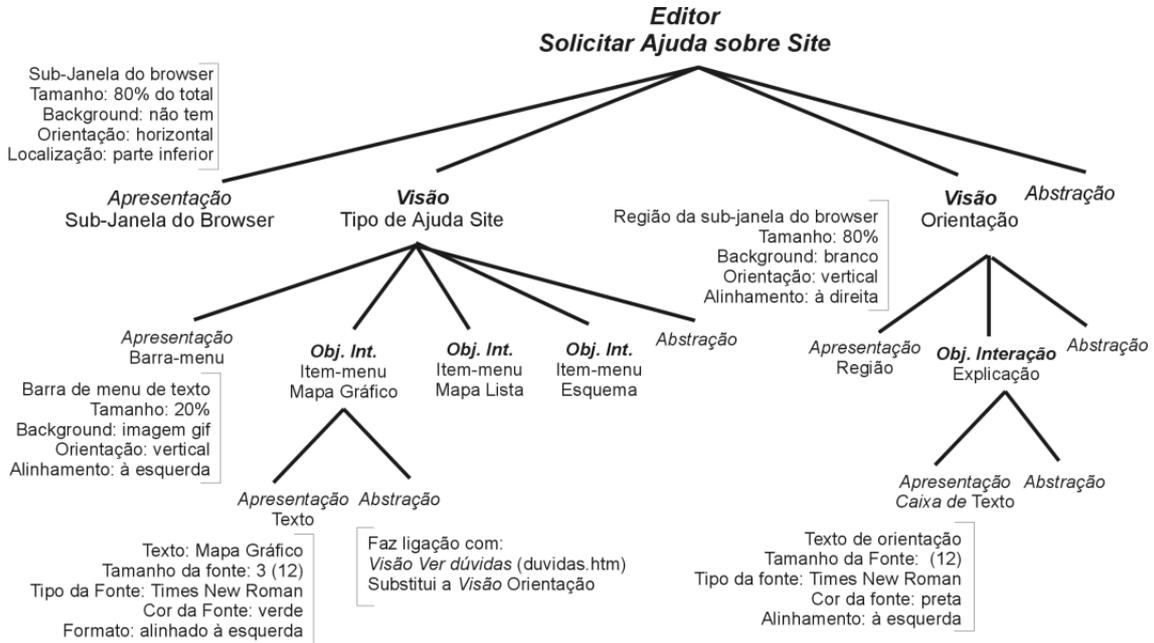


Figura 46: Árvore EDITOR completa da tarefa "Solicitar Ajuda sobre Site"



Figura 47: Protótipo janela "Solicitar Ajuda sobre Site"