

Universidade Federal de Campina Grande
Centro de Ciências e Tecnologia
Coordenação de Pós-Graduação em Informática

**Projeto e implementação do módulo TAOS-Graph
da ferramenta *i*TAOS para análise e modelagem
da tarefa**

Francisco Petrônio Alencar de Medeiros

Campina Grande - PB
Fevereiro de 2003

Universidade Federal de Campina Grande
Centro de Ciências e Tecnologia
Coordenação de Pós-Graduação em Informática

Projeto e implementação do módulo TAOS-Graph da ferramenta *i*TAOS para análise e modelagem da tarefa

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

Instituição: Universidade Federal de Campina Grande
Área de Concentração: Ciência da Computação
Linha de Pesquisa: Engenharia de Software

Francisco Petrônio Alencar de Medeiros
Orientador: Bernardo Lula Jr., Dr.

Campina Grande - PB
Fevereiro de 2003

Ficha Catalográfica

MEDEIROS, Francisco Petrônio Alencar de Medeiros

M488P

Projeto e Implementação do módulo TAOS-Graph da ferramenta iTAOS para análise e modelagem da tarefa

Dissertação (Mestrado), Universidade Federal de Campina Grande, Centro de Ciências e Tecnologia, Coordenação de Pós-Graduação em Informática, Campina Grande – Pb, Fevereiro de 2003.

106 p. Il.

Orientador: Bernardo Lula Júnior, Dr.

Palavras-Chaves:

1. Engenharia de Software
2. Interfaces Homem-Máquina
3. Análise e Modelagem de Tarefas
4. Formalismos TAOS (Task and Action Oriented System)

CDU – 519.683

**“PROJETO E IMPLEMENTAÇÃO DO MÓDULO TAOS-GRAPH DA
FERRAMENTA ITAOS PARA ANÁLISE E MODELAGEM DA TAREFA”**

FRANCISCO PETRÔNIO ALENCAR DE MEDEIROS

DISSERTAÇÃO APROVADA EM 26.02.2003



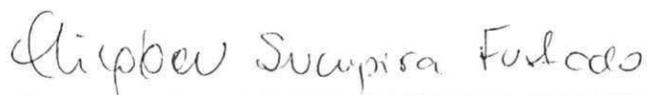
PROF. BERNARDO LULA JÚNIOR, Dr.
Orientador



PROFª FRANCILENE PROCÓPIO GARCIA, D.Sc
Examinadora



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



PROFª MARIA ELIZABETH SUCUPIRA FURTADO, Drª
Examinadora

CAMPINA GRANDE – PB

**Projeto e implementação do módulo TAOS-Graph da
ferramenta *i*TAOS para análise e modelagem da tarefa**

Francisco Petrônio Alencar de Medeiros

Bernardo Lula Jr., Dr., UFCG
Orientador

Francilene Procópio Garcia, Dsc., UFCG
Componente da Banca

Maria Elizabeth Sucupira Furtado, DsC., UNIFOR
Componente da Banca

Maria de Fátima Queiroz Vieira Turnell, Ph.D., UFCG
Componente da Banca

Campina Grande, fevereiro de 2003

*Este trabalho é dedicado a
minha família: Gabriel, Fátima,
Patrícia, Manoel Neto e Gabriela (Gaby).*

Agradecimentos

A Deus, por essa vida maravilhosa.

Aos meus pais, Gabriel e Fátima, pelo esforço constante em favor da minha formação.

Aos meus irmãos, Manoel Neto (Neneto), Patrícia (Teté) e Gabriela (Gaby), pela cumplicidade, força e amizade.

A toda minha família, pelo apoio recebido.

Ao meu orientador, Bernardo Lula, pelo grande aprendizado durante todo este período e pelas discussões periódicas na busca de um trabalho bem feito.

Ao amigo e colega de projeto Pedro, por enfrentarmos juntos os dias enfurecidos de Bernardo e por compartilharmos os momentos de sucesso.

A professora Fátima Turnell, ao professor Eustáquio e ao professor Marconi, pela colaboração nesse trabalho. Aos professores do DSC que despertaram meu lado acadêmico e que me forneceram grande conhecimento.

Ao meu professor e amigo Hamurabi, pelos conselhos na vida pessoal e profissional.

Aos meus grandes amigos Alisson, Alexandre, Philip e Claudivan, pelo incentivo nos momentos de stress e companhia nas horas de alegria.

Aos colegas e amigos de apartamento, Alisson, Eloi, Alex, Heronides (Lula) e Daniel (BA), por proporcionarem um astral sempre positivo no ambiente de estudo.

A todos meus amigos de Campina Grande e Patos.

A todos os funcionários do DSC, em especial a Aninha, Vera e Zeneide.

A Dona Inês e Dona Maria, pelos almoços e cafezinhos.

A todos que contribuíram diretamente ou indiretamente para que eu alcançasse mais essa etapa da minha vida.

Sumário

<i>Sumário</i>	<i>i</i>
<i>Lista de figuras</i>	<i>iv</i>
<i>Lista de tabelas</i>	<i>v</i>
<i>Lista de descritores</i>	<i>vi</i>
<i>RESUMO</i>	<i>vii</i>
<i>ABSTRACT</i>	<i>viii</i>
1. Introdução	1
1.1. <i>Independência do diálogo</i>	<i>1</i>
1.2. <i>Análise da tarefa</i>	<i>3</i>
1.3. <i>Metodologias de concepção baseadas na tarefa</i>	<i>4</i>
1.5. <i>TAOS – Task and Action Oriented System</i>	<i>5</i>
1.6. <i>Problemática</i>	<i>6</i>
1.7. <i>Objetivos do trabalho</i>	<i>7</i>
1.8. <i>Hipóteses</i>	<i>8</i>
1.9. <i>Metodologia de trabalho</i>	<i>8</i>
1.10. <i>Estrutura da dissertação</i>	<i>9</i>
2. Formalismo TAOS para análise e modelagem de tarefas no contexto do projeto de interfaces do usuário	11
2.1. <i>TAOS: Inteligência Artificial</i>	<i>11</i>
2.2. <i>Modelo TAOS</i>	<i>12</i>
2.2.1. <i>Conceitos estáticos</i>	<i>13</i>
2.2.2. <i>Conceitos dinâmicos</i>	<i>15</i>
2.3. <i>Validação de TAOS face MAD</i>	<i>17</i>
2.4. <i>TAOS: Modelagem da tarefa</i>	<i>18</i>
2.4. <i>Considerações finais</i>	<i>19</i>
3. MEDITE – Uma metodologia baseada na tarefa e orientada a modelos para concepção de interfaces ergonômicas	21
3.1. <i>Descrição de MEDITE</i>	<i>21</i>
3.1.1. <i>Análise e modelagem da tarefa</i>	<i>23</i>
3.1.2. <i>Especificação conceitual parcial da interação</i>	<i>23</i>
3.1.3. <i>Definição dos atributos</i>	<i>24</i>
3.1.4. <i>Geração do protótipo</i>	<i>24</i>

3.2. Modelo MAD*	25
3.2.1. Conceitos definidos por MAD*	25
3.3. Modelo EDITOR	27
3.3.1. Agentes do modelo EDITOR	28
3.4. Base de regras ergonômicas	30
3.5. Suporte computacional (Ferramentas)	31
3.6. Considerações finais	31
4. Análise e modelagem da tarefa “Utilizar a ferramenta iTAOS para analisar e modelar tarefas”	33
4.1. Análise e modelagem da tarefa na metodologia MEDITE	33
4.2. Análise e Modelagem da Tarefa	34
4.2.1. Aquisição de dados sobre a tarefa e sobre o usuário	34
4.2.2. Descrição da tarefa utilizando TAOS	35
4.3. Avaliação da modelagem	38
4.4. Considerações finais	39
5. Especificação conceitual da interação	40
5.1. Especificação conceitual da interação na metodologia MEDITE	40
5.2. Especificação conceitual parcial da interação	41
5.3. Especificação conceitual completa da interação	44
5.4. Avaliação da Especificação	47
5.5. Integração dos módulos TAOS-Graph e TAME	48
5.6. Considerações finais	48
6. Protótipo da interface	50
6.1. Protótipo da interface na metodologia MEDITE	50
6.2. Framework JHotDraw	51
6.2.1. Processo de desenvolvimento típico Usando o JHotDraw	52
6.3. Classes que compõem o sistema	53
6.4. Apresentação de iTAOS	54
6.5. Considerações finais	58
7. Inspeção da usabilidade do protótipo da interface da ferramenta iTAOS	59
7.1. O padrão ISO 9241	59
7.2. Metodologia utilizada na inspeção da usabilidade	61
7.3. Problemas identificados no produto a partir da inspeção de usabilidade	63
7.4. Taxas de adoção com base no processo de inspeção	65

7.5. <i>Considerações finais</i>	66
8. Conclusão	67
8.1. <i>Discussão dos resultados</i>	68
8.2. <i>Contribuições</i>	70
8.3. <i>Propostas de continuidade</i>	72
Referências Bibliográficas	73
Apêndice 1 – Formulário DePerUse	79
Apêndice 2 – Perfil do usuário	81
Apêndice 3 – Lista de verificação da aplicabilidade e adoção do ISO 9241-14	83
Apêndice 4 – Lista de verificação da aplicabilidade e adoção do ISO 9241-16	93
Apêndice 5 – Lista de verificação da aplicabilidade e adoção do ISO 9241-17	98

Lista de figuras

FIGURA 1. META-MODELO TAOS.....	12
FIGURA 2. METODOLOGIA MEDITE.....	23
FIGURA 3. CONCEITOS DEFINIDOS POR MAD*	25
FIGURA 4. ORGANIZAÇÃO DOS AGENTES DE UM EDITOR.....	29
FIGURA 5. ANÁLISE E MODELAGEM DA TAREFA NA METODOLOGIA MEDITE.....	33
FIGURA 6. TAREFA “MODELAR_TAREFAS_UTILIZANDO_ITAOS”	35
FIGURA 7. TAREFA “EDITAR_AGENTE”	36
FIGURA 8. ESPECIFICAÇÃO CONCEITUAL DA INTERAÇÃO NA METODOLOGIA MEDITE.....	40
FIGURA 9. ÁRVORE EDITOR PARCIAL DA TAREFA "MODELAR TAREFA"	42
FIGURA 11. ÁRVORE EDITOR PARCIAL DA TAREFA "EDITAR AGENTE"	43
FIGURA 12. ÁRVORE EDITOR COMPLETA DA TAREFA "MODELAR TAREFA"	45
FIGURA 13. ÁRVORE EDITOR COMPLETA DA TAREFA "REALIZAR MODELAGEM"	46
FIGURA 14. ÁRVORE EDITOR COMPLETA DA TAREFA "EDITAR AGENTE"	47
FIGURA 15. PROTÓTIPO DA INTERFACE NA METODOLOGIA MEDITE.....	50
FIGURA 16. ESTRUTURA EM PACOTES DO FRAMEWORK JHOTDRAW.....	51
FIGURA 17. DIAGRAMA DAS PRINCIPAIS CLASSES DO MÓDULO TAOS-GRAPH DA FERRAMENTA ITAOS.....	53
FIGURA 18. TELA INICIAL DA FERRAMENTA ITAOS	54
FIGURA 19. TELA QUE MOSTRA O MENU FILE ABERTO	55
FIGURA 20. TELA QUE MOSTRA UM MODELO ABERTO	55
FIGURA 22. TELA QUE MOSTRA O DESCRITOR DE UMA AÇÃO ABERTO.....	56
FIGURA 22. TELA QUE MOSTRA O DESCRITOR DE UM AGENTE ABERTO	57
FIGURA 23. JANELA QUE APRESENTA AS AÇÕES DO MODELO ABERTO.....	58
FIGURA 24. FLUXOGRAMA EMPREGADO NA INSPEÇÃO DA USABILIDADE	62

Lista de tabelas

TABELA 1. PARTES INTEGRANTES DO PADRÃO ISO 9241	60
TABELA 2 – TAXAS DE ADOÇÃO DO ÍTAOS ÀS PARTES 14, 16 E 17 DO ISO 9241.....	65

Lista de descritores

DESCRITOR 1: CLASSE TAREFA <i>REALIZAR_MODELAGEM</i>	36
DESCRITOR 2. CLASSE SITUAÇÃO <i>SITUAÇÃO_INICIAL_DE_REALIZAR_MODELAGEM</i>	37
DESCRITOR 3. CLASSE SITUAÇÃO <i>SITUAÇÃO_OBJETIVO_DE_REALIZAR_MODELAGEM</i>	37
DESCRITOR 4. CLASSE MÉTODO <i>MÉTODO_REALIZAR_MODELAGEM</i>	37
DESCRITOR 5: CLASSE PLANO <i>EDITAR_NOME_AGENTE</i>	37
DESCRITOR 6. CLASSE SITUAÇÃO <i>SITUAÇÃO_INICIAL_DE_EDITAR_NOME_AGENTE</i>	38
DESCRITOR 7. CLASSE SITUAÇÃO <i>SITUAÇÃO_OBJETIVO_DE_EDITAR_NOME_AGENTE</i>	38
DESCRITOR 8: CLASSE AGENTE <i>PROJETISTA_DE_INTERFACE</i>	38
DESCRITOR 9: CLASSE INSTRUMENTO <i>AMBIENTE_DE_MODELAGEM</i>	38

RESUMO

Esse trabalho apresenta o processo de construção e implementação do módulo TAOS-Graph da ferramenta iTAOS. iTAOS é uma ferramenta gráfica que implementa o formalismo TAOS (Task and Action Oriented System) concebida para acompanhar o projetista de interfaces durante a fase de análise e descrição da tarefa dentro de um processo de desenvolvimento de interfaces, verificando a completude e consistência da representação. TAOS-Graph foi desenvolvido utilizando a metodologia MEDITE, uma metodologia guiada por modelos e baseada na tarefa para construção de interfaces ergonômicas. Os artefatos gerados ao final de cada etapa do processo de desenvolvimento de TAOS-Graph foram: a descrição TAOS da tarefa, a especificação conceitual da interação e o código da interface. Como recomenda a metodologia, foi realizada uma inspeção de conformidade da ferramenta iTAOS com as partes 14 (Menus), 16 (Manipulação direta) e 17 (Formulários) do padrão ISO 9241.

ABSTRACT

This work presents the process of construction and implementation of the TAOS-Graph module of the iTAOS tool. iTAOS is a graphical tool that implements the TAOS formalism (Task and Action Oriented System) and is responsible for accompanying the interface designer (iTAOS user) during domain task's description and analysis phases within the interface development process, verifying the completeness and the consistency of the representation. TAOS-Graph was developed using the methodology MEDITE, a methodology guided for models and based in the task for construction of ergonomic interfaces. The artefacts generated to the end of each stage of the development process of TAOS-Graph had been: description TAOS of the task, the conceptual specification of the interaction and the code of the interface. As recommends the methodology, iTAOS was carried through an inspection of conformity with the parts 14, 16 and 17 of the standard ISO 9241.

Capítulo 1

1. Introdução

O campo da interação homem-computador é o estudo do indivíduo (usuário), da tecnologia computacional (sistema) e do entendimento do trabalho (tarefa) que o indivíduo tenta realizar utilizando essa tecnologia. No processo de interação usuário-sistema, a interface é o combinado de software e hardware necessário para viabilizar e facilitar os processos de comunicação entre o usuário e o sistema computacional utilizado para realizar a tarefa (Souza et al., 1999).

Para o usuário, o sistema é apenas uma ferramenta e, como tal, exige, para ser utilizado, um conhecimento e habilidade não somente do domínio da tarefa, mas também do uso do sistema, envolvendo as funcionalidades e os meios disponíveis para utilizá-lo. Dessa forma, aprender a usar o sistema torna-se uma tarefa adicional para o usuário, e, portanto, deve exigir o menor esforço possível. Portanto, é de fundamental importância que o sistema seja adaptado às características do usuário, às tarefas a serem executadas e ao ambiente de trabalho, para que o esforço adicional de ter que aprender a usar o sistema seja o menor possível. É nesse sentido que vem se exigindo, cada vez mais, dos projetistas desses produtos e sistemas, uma preocupação com a qualidade da interface.

A meta é oferecer aos usuários sistemas com alto grau de usabilidade, ou seja, que satisfaçam os requisitos de funcionalidade e sejam fáceis de usar, entender e aprender.

1.1. Independência do diálogo

Nos dias de hoje, os postos nas equipes que desenvolvem sistemas computacionais são ocupados, principalmente, por profissionais de informática por formação e esses se

empenham, naturalmente, em definir, antes de tudo, as funções lógicas do sistema sem de fato se preocuparem com o uso deste, ou seja, a interação com o usuário do sistema.

Na realidade, a maioria das metodologias tradicionais de desenvolvimento de software não se preocupa com a interface do usuário, que passa a ser considerada apenas nas fases finais do processo, tornando-se assim nada mais que um “apêndice” do sistema.

As primeiras abordagens de concepção de sistemas interativos tinham uma tendência a “misturar” fortemente o código da interface com o código que implementa a funcionalidade do sistema. A dificuldade de modificação proibia uma abordagem iterativa, praticamente impedindo de se levar em conta as opiniões de especialistas em comunicação e os resultados da observação do usuário durante a concepção e o desenvolvimento da interface.

Assim, como no campo de Banco de Dados, com o princípio da independência dos dados, e como no campo da Inteligência Artificial, com o princípio da independência do conhecimento, a solução preconizada para uma abordagem iterativa do desenvolvimento da interface foi a separação modular entre o código da implementação (componente funcional) e o código da interface (componente interativo). O componente funcional (ou aplicação) define a funcionalidade do sistema, ou seja, as funções que a aplicação disponibiliza para a realização de suas tarefas. O componente de interação (ou interface) é o software responsável pelo diálogo entre o usuário e o sistema. Essa decomposição deve permitir se conceber, desenvolver e modificar os módulos/componentes independentemente um do outro (Ehrich e Hartson, 1981; Draper e Norman, 1985; Dodani et al., 1985). O *princípio de independência do diálogo*, termo consagrado para caracterizar essa decomposição, tem se tornado um princípio essencial a respeitar no processo de concepção e desenvolvimento de sistemas interativos, pelas razões já expostas, ou sejam:

- i.* A possibilidade de um desenvolvimento independente dos módulos;
- ii.* A possibilidade de um processo iterativo de concepção e desenvolvimento;
- iii.* A possibilidade de utilização de profissionais em comunicação (ergonomistas, artistas gráficos, psicólogos da cognição, lingüistas, etc.);
- iv.* A possibilidade de se construir diversas interfaces representando diversas maneiras de comunicação, com uma mesma aplicação;

- v. A possibilidade de reutilizar toda ou parte de uma interface para outra aplicação;

1.2. Análise da tarefa

A separação modular preconizada pelo princípio da independência do diálogo abriu caminho para o uso da ergonomia no processo de concepção e desenvolvimento de interfaces de qualidade.

Em Ergonomia, um princípio fundamental consagrado no projeto de ferramentas é a necessidade de conhecimento do usuário e do trabalho a ser realizado (Cybis, 1996; Sebillote, 1995). O trabalho é visto segundo dois componentes básicos: a tarefa e a atividade (Hammouche, 1995):

Tarefa: é descrita por um objetivo (o objetivo do usuário), uma lista de sub-tarefas (os procedimentos) necessárias para atingir o objetivo, assim como suas restrições estruturais e temporais. A definição de tarefa é recursiva, visto que uma sub-tarefa é também uma tarefa, distinguindo-as apenas pelo nível de abstração. A recursividade pára quando a lista de sub-tarefas só agrupa ações.

Atividade: Um conjunto de ações *efetuadas por um ou mais operadores humanos em vista de realizar um ou mais objetivos*. A atividade pode ser vista como a instanciação da tarefa, ou simplesmente, a tarefa realizada.

A análise da tarefa descreve o conjunto de tarefas com o objetivo de entender o funcionamento cognitivo dos operadores (Sebillote, 1995) e é realizado a partir de entrevistas dirigidas, observações, experimentos ou outros métodos habituais de pesquisa, buscando evidenciar as características do processo de realização. O resultado da análise é um documento (relatório) contendo uma descrição detalhada e hierarquizada do trabalho segundo o ponto de vista do usuário.

Do ponto de vista da concepção de sistemas, o resultado da análise pode ser aplicado para apoiar as ações do projetista em pelo menos três momentos: na especificação do sistema (funcionalidades), no projeto da interface e na elaboração de manuais de treinamento (Heemann, 1997).

No entanto, a utilização efetiva da análise da tarefa no processo de concepção e desenvolvimento de interfaces não se deu que após a introdução de formalismos apropriados e que pudessem ser entendidos tanto pelos ergonomistas quanto pelos projetistas de interfaces e/ou sistemas. Estes formalismos têm surgido como tentativas de integrar a análise e a descrição da tarefa ao processo de concepção de interfaces (Limbourg, Pribeanu e Vanderdonckt, 2001), melhorando, com isso, a qualidade das descrições e maximizando a completude e a coerência (Gamboa e Scapin, 1997).

Diversos formalismos foram propostos com essa finalidade e destacamos os seguintes pelos seus graus de incidência na literatura a respeito: MAD (Méthode Analytique de Description) (Scapin e Pierre-Golkbreich, 1989), TKS (Task Knowledge Structure) (Johnson at al., 1988), ETAG (Tauber, 1988; Tauber, 1990; Haan, Van der Veer e Van Vliet, 1992; Haan, 2000) e MAD* (Hammouche, 1995, Gamboa, 1998).

O Grupo de Interfaces Homem-Máquina (GIHM) da Universidade Federal de Campina Grande (UFCG) tem proposto o formalismo TAOS para análise da tarefa no âmbito da concepção e desenvolvimento de interfaces de sistemas computacionais.

1.3. Metodologias de concepção baseadas na tarefa

A partir da necessidade de se desenvolver aplicações utilizáveis e acessíveis a um grande número de usuários, diversas metodologias de concepção e desenvolvimento de interfaces foram propostas. A maioria dessas metodologias tem como elemento base a representação da tarefa que o usuário desempenha ou deve desempenhar com a utilização do sistema correspondente. Elas guiam, a partir da descrição da tarefa, do perfil do usuário e de princípios ergonômicos, a construção de um protótipo da interface levando em conta os objetivos do usuário. As metodologias que se baseiam no modelo da tarefa do usuário constituem um expressivo apoio à concepção de interfaces ergonômicas (Morkopoulos e Gikas, 1997). Neste contexto, a análise e modelagem da tarefa não têm o objetivo de avaliar, mas descrever uma tarefa precisamente com o intuito de entender a “lógica do usuário”.

Dentre as metodologias que se enquadram nesse contexto, citamos: TRIDENT (Bodart at al, 1994; Bodart at al, 1995), ADEPT (Johnson at al., 1993), ERGO-START (Hammouche, 1995), MACIA (Furtado, 1997), ALACIE (Gamboa, 1998) e MCI (Sousa, 1999).

O Grupo de Interfaces Homem-Máquina da UFCG tem proposto a metodologia MEDITE (Guerrero, 2001; Guerrero e Lula, 2001a; Guerrero e Lula, 2001b; Guerrero e Lula, 2002), orientada a modelos e baseada na tarefa, para a concepção de interfaces ergonômicas.

1.4. Ferramentas para análise e modelagem da tarefa

Embora modelagem da tarefa e projeto baseado na tarefa estejam sendo largamente considerados nos projetos de interfaces homem-computador, sua adoção tem sido limitada pela carência de ferramentas que suportem o desenvolvimento, a análise interativa e o uso de modelos da tarefa.

As ferramentas inicialmente desenvolvidas para análise da tarefa de concepção de interfaces homem-máquina, por exemplo, Adept (Wilson et al., 1993) e IMAD* (Gamboa e Scapin, 1997) são rudimentares, com características limitadas e usadas somente pelos grupos de pesquisas que as desenvolveram.

Mais recentemente, algumas ferramentas mais elaboradas, com características mais interessantes foram desenvolvidas e disponibilizadas na Internet (Euterpe (Van Welie, M. e Van Der Veer, 1998)).

1.5. TAOS – Task and Action Oriented System

TAOS é um formalismo de aquisição e representação do conhecimento baseado na modelagem do domínio e foi desenvolvido por J.H. de Medeiros (Medeiros, 1995; Medeiros e Rousselot, 1995a, Medeiros e Rousselot, 1995b). TAOS se insere entre os trabalhos teóricos e experimentais desenvolvidos nesses últimos anos em Inteligência Artificial, no campo da construção de sistemas baseados em conhecimento (SBC) fundado na modelagem do domínio. Esses trabalhos têm mostrado as vantagens de metodologias baseadas na construção de modelos desde a fase de aquisição de conhecimento até a fase de projeto do SBC (Silva, 1999).

Kafure, em (Kafure, 2000), mostra que TAOS tem o mesmo poder descritivo que MAD para a análise da tarefa no contexto de concepção de interfaces. Assim como MAD, TAOS permite descrever os objetivos do usuário, a decomposição de tarefas em sub-tarefas,

as relações estruturais e temporais entre as sub-tarefas, suas condições de inicialização e de término, os objetos necessários para sua realização, entre outros.

O modelo de representação proposto pelo formalismo TAOS considera que podem existir em um domínio dois tipos de entidades ou conceitos: os *conceitos estáticos*, que conservam suas características durante um intervalo de tempo pré-estabelecido (objetos, métodos e situações) e os *conceitos dinâmicos* (processos, planos e ações), que representam uma evolução da situação observada num intervalo de tempo. Segundo Medeiros (1995), essa distinção é importante para a representação do conhecimento uma vez que a natureza das relações lógicas que existem entre conceitos estáticos (na maioria dos casos, apenas de generalização/particularização e decomposição) não é a mesma que existe entre os conceitos dinâmicos (envolve também relações de temporalidade). A representação orientada-objeto dos conceitos (estáticos e dinâmicos) é um grafo (árvore), cujos nós representam as classes e instâncias de classes, e cujos arcos representam relações entre as classes ou entre classes e instâncias.

TAOS foi concebido para ser implantado em dois módulos: módulo TAME (“*Task and Action Modeling Environment*”) e módulo TAOS-Graph. O módulo TAME guia a construção de um modelo para um domínio. Ele define uma linguagem e uma metodologia descendente-ascendente de construção do modelo do domínio. O módulo TAOS-Graph permite a visualização do processo de modelagem. Trata-se de um editor de entidades estáticas e dinâmicas estruturadas de acordo com os princípios preconizados pelo módulo TAME. Isso permite modelar o conhecimento do domínio fazendo uma verificação preliminar desse conhecimento, tanto do ponto de vista da sua completude quanto de sua coerência.

1.6. Problemática

O surgimento de formalismos para análise da tarefa foi um passo importante para melhorar a qualidade das descrições das tarefas, principalmente do que diz respeito à colaboração entre ergonômicos e projetistas que passaram a usar uma linguagem formal comum.

Porém, realizar a análise e modelagem de tarefas sem a disponibilidade de uma ferramenta específica que auxilie o projetista implica em problemas (Gamboa e Scapin, 1997), tais como:

- i.* Dificuldade em descrever a estrutura de árvores das tarefas e suas notações formais;
- ii.* Dificuldade de verificar a consistência e completude das tarefas modeladas;
- iii.* Impossibilidade de geração automática ou semi-automática do modelo da interação a partir do modelo da tarefa;

Esses problemas listados são sentidos com uma grande intensidade por projetistas que analisam e modelam tarefas utilizando os formalismos disponíveis citados anteriormente. Pela dificuldade em adquirir uma ferramenta de modelagem, a construção e manipulação das árvores do modelo da tarefa são feitas utilizando lápis e papel ou, no máximo, ferramentas que desenhavam e manipulavam figuras geométricas básicas, e não os conceitos de tarefas envolvidos. Por esse motivo, há uma grande dificuldade em verificar a consistência e completude da tarefa, principalmente pela necessidade de memorização dos atributos, objetos e termos relacionados com a tarefa que está sendo modelada.

Além disso, a não existência de uma representação digital apropriada da tarefa implica: *(i)* na impossibilidade de se ter ferramentas computacionais para auxiliar os projetistas nos passos seguintes do processo e *(ii)* em grande dificuldade na utilização de conhecimentos ergonômicos para a obtenção da representação da interação (modelo da interação).

1.7. Objetivos do trabalho

Este trabalho teve como objetivo principal projetar e implementar o módulo TAOS-Graph e integrá-lo ao módulo TAME gerando a ferramenta *i*TAOS (*i*nterface TAOS) de suporte a análise e modelagem da tarefa no contexto de concepção e desenvolvimento de interfaces do usuário. A ferramenta *i*TAOS deverá: *(i)* apoiar o projetista de interfaces na elaboração da representação da tarefa segundo o formalismo e metodologia TAOS, *(ii)* verificar a consistência e completude da representação e *(iii)* viabilizar a construção de outras

ferramentas computacionais de suporte à transformação da representação da tarefa na representação da interação segundo critérios ergonômicos.

Como objetivo periférico, mas não menos importante, contribuir para a promoção e a consolidação da idéia de que metodologias de concepção e desenvolvimento de interfaces que levem em conta, desde o início do processo, a análise da tarefa e o conhecimento ergonômico produzem interfaces do usuário com alto grau de usabilidade.

1.8. Hipóteses

As hipóteses que fundamentaram a escolha da metodologia de trabalho aplicada a construção de nosso objetivo principal foram as seguintes:

- i.* É possível projetar e implementar o módulo TAOS-Graph separado do processo de desenvolvimento do módulo TAME.
- ii.* O processo de desenvolvimento a ser utilizado no projeto do módulo TAOS-Graph deve partir da análise da tarefa “Utilizar a Ferramenta iTAOS para Analisar e Modelar Tarefas” de forma a capturar efetivamente o conhecimento sobre o usuário e sobre o seu processo de realização da tarefa.
- iii.* A metodologia MEDITE possibilita uma construção guiada por modelos do módulo TAOS-Graph de forma a obedecer aos princípios da ergonomia e aos objetivos do usuário da ferramenta iTAOS.

1.9. Metodologia de trabalho

A metodologia que adotamos no desenvolvimento deste trabalho consiste nas seguintes etapas:

- i.* Pesquisa bibliográfica sobre formalismos de descrição de tarefas, metodologias de concepção baseadas na tarefa e ferramentas para análise da tarefa.
- ii.* Estudo detalhado sobre TAOS e sobre a metodologia MEDITE (Guerrero, 2001; Guerrero e Lula, 2001; Guerrero e Lula, 2002a; Guerrero e Lula, 2002b).
- iii.* Levantamento do perfil do usuário.

- iv. Análise e modelagem da tarefa “Analisar e modelar tarefas utilizando o formalismo TAOS”.
- v. Produção da especificação conceitual da interação utilizando o modelo EDITOR (Lula, 1992).
- vi. Construção do protótipo da interface utilizando a linguagem de programação JAVA¹. Essa linguagem foi escolhida devido a sua portabilidade e à experiência por parte dos projetistas da ferramenta iTAOS.
- vii. Avaliação dos artefatos projetados. Esta avaliação foi feita interativamente e iterativamente à medida que o modelo era desenvolvido e ao final através de uma avaliação baseada na inspeção de padrões (Stewart, 2000).

1.10. Estrutura da dissertação

O trabalho está organizado em oito capítulos da seguinte forma: O segundo capítulo apresenta o formalismo TAOS e seu uso para analisar e modelar tarefas no contexto de interface homem-máquina.

O terceiro capítulo apresenta a metodologia MEDITE, uma metodologia baseada na tarefa para a concepção de interfaces ergonômicas e utilizada no projeto e implementação do módulo TAOS-Graph, componente interativo da ferramenta iTAOS.

O quarto capítulo apresenta a análise e modelagem da tarefa “Modelar tarefa utilizando a ferramenta iTAOS”. Nesta fase, os dados colhidos junto aos usuários sobre a tarefa analisada se deu através de técnicas como observações, entrevistas com os usuários e consultas a manuais.

O quinto capítulo apresenta a utilização de regras ergonômicas da base de regras formuladas em MEDITE na criação da especificação conceitual da interação a partir do modelo da tarefa. Algumas das regras foram adaptadas e outras criadas, de forma que fossem úteis ao projeto.

¹ JAVATM é uma tecnologia Sun Microsystems

O sexto capítulo apresenta o protótipo da interface implementado com a linguagem de programação JAVA.

O sétimo capítulo apresenta a avaliação aplicada ao produto final. Trata-se de uma avaliação baseada em padrões ergonômicos.

No oitavo capítulo, finalmente, discute-se os resultados alcançados, sugestões de continuidade e trabalhos futuros.

Capítulo 2

2. Formalismo TAOS para análise e modelagem de tarefas no contexto do projeto de interfaces do usuário

Neste capítulo descrevemos o formalismo TAOS: o seu surgimento dentro do campo da Inteligência Artificial, o modelo o qual ele define e sua utilização como formalismo para análise e modelagem de tarefas no contexto de interfaces homem-máquina.

2.1. TAOS: Inteligência Artificial

A representação e a manipulação do conhecimento é um problema comum encontrado nos campos da Psicologia, da Ergonomia e da Informática, particularmente da Inteligência Artificial (IA). Portanto, grande parte do esforço em IA tem se concentrado em buscar ou aperfeiçoar formalismos para a representação do conhecimento.

TAOS é um formalismo para aquisição e representação do conhecimento baseado na modelagem do domínio (Kessel, Medeiros e Rousselot, 1995a; Medeiros e Rousselot, 1995b; Medeiros, 1995; Medeiros, Kafure e Lula, 2000), foi concebido por J. H. de Medeiros e validado inicialmente no domínio da biologia molecular. TAOS define um modelo teórico e experimental que se fundamenta no conhecimento que um especialista possui sobre um determinado assunto. Esse conhecimento é extraído através de técnicas de aquisição do conhecimento, como entrevistas, questionários, protocolos etc, e representado através de uma descrição hierárquica formal baseada no conceito de frames (Minsky, 1975).

O problema da análise e modelagem da tarefa do usuário é semelhante ao problema do desenvolvimento de um sistema baseado em conhecimento (SBC): aquisição/extração do conhecimento do usuário (especialista), representação do conhecimento (base de

conhecimento) e verificação da representação (inferência). Portanto, resolver o problema da análise e descrição de tarefas significa obter o conhecimento que o usuário tem da tarefa e representá-la de forma adequada através de um formalismo.

2.2. Modelo TAOS

O modelo de representação proposto por TAOS² considera que os conceitos podem ser de dois tipos: conceitos estáticos e conceitos dinâmicos. Um conceito é considerado estático se ele conserva suas características durante um intervalo de tempo pré-estabelecido. Os conceitos objeto (*Object*), agente (*Agent*), instrumento (*Instrument*), situação (*Situation*) e método (*Method*) são considerados estáticos. Um conceito é considerado dinâmico se ele representa a evolução de uma situação observada num intervalo de tempo pré-estabelecido. Os conceitos processo (*Process*), plano (*Plan*) e ação (*Action*) são considerados dinâmicos.

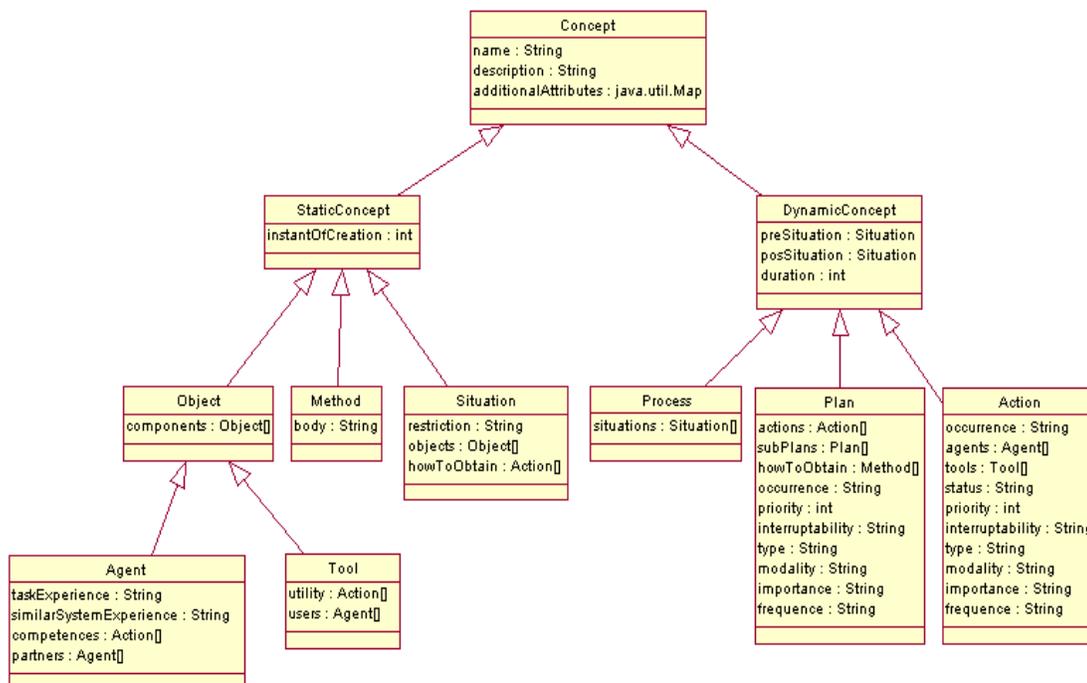


Figura 1. Meta-modelo TAOS

² Os nomes das classes e de seus atributos são apresentados em inglês como definidos no meta-modelo (Figura 1).

A figura 1 ilustra a taxonomia dos conceitos definidos por TAOS (Meta-modelo). A representação gráfica dos conceitos é uma árvore, cujos nós representam as classes e cujos arcos representam as relações entre as classes.

A distinção entre conceitos estáticos e dinâmicos é importante para a representação do conhecimento uma vez que a natureza das relações lógicas que existem entre conceitos estáticos não é a mesma que existe entre conceitos dinâmicos. Enquanto que as relações existentes entre os conceitos estáticos envolvem quase sempre relações de generalização e decomposição, as relações entre conceitos dinâmicos envolvem também relações de temporalidade.

A classe de mais alto nível, denominada **Concept**, define a entidade de maior grau de abstração, possui como atributos *name* e *description* que são do tipo cadeia de caracteres (String), além de atributos opcionais, tais como : *additionalAttributes*, que é representado por uma tupla (nome do atributo, tipo do atributo).

2.2.1. Conceitos estáticos

A classe **StaticConcept** é uma sub-classe da classe **Concept**. A classe **StaticConcept** se decompõe em três outras classes: **Object**, **Method** e **Situation**, sendo que a classe **Object** se decompõe em mais duas, que são **Tool** e **Agent**.

A classe **StaticConcept** possui além dos atributos herdados da classe **Concept**, o atributo *instantOfCreation*, que representa o instante de criação de um determinado conceito estático.

A classe **Object** possui como novo atributo em relação a sua superclasse **StaticConcept** o atributo *Components*. Este atributo representa a decomposição do objeto em suas partes constituintes e é representada por uma lista (vetor) de **Objects**.

A classe **Tool** define um tipo de objeto que é utilizado pelos agentes para executar suas ações (ferramenta). Essa classe, além dos atributos herdados da classe **Object**, possui os seguintes atributos:

- i. *Utility* : Ações que podem ser realizadas utilizando a ferramenta (instrumento). É representado por uma lista (vetor) de ações.

- ii. *Users* : Agentes que utilizam a ferramenta para execução de uma ação. É representado por uma lista (vetor) de agentes.

A classe *Agent* define um tipo de objeto habilitado a executar uma ação. Além dos atributos herdados da classe *Object*, a classe *Agent* possui os seguintes atributos :

- i. *Competences* : O agente realiza ações que ele conhece e para as quais ele é competente. Este atributo é representado por uma lista (vetor) de ações.
- ii. *Partners* : O agente pode tomar decisões a partir de informações que ele recebe de outros agentes. Este atributo é representado por uma lista (vetor) de outros agentes.

A classe *Method* define a estratégia para executar um plano (sub-planos e ações). Além dos atributos herdados da classe *StaticConcept*, a classe *Method* possui o atributo *body*. Este atributo define uma expressão composta ou simples para descrever a decomposição do plano em sub-planos e ações, segundo a gramática abaixo :

```

<expression> ::= <opr> (<list-expression>)
<list-expression> ::= <action>,<list-expression>
<list-expression> ::= <plan>,<list-expression>
<list-expression> ::= <expression>,<list-expression>
<list-expression> ::= <action>,<simple-expression>
<list-expression> ::= <plan>,<simple-expression>
<list-expression> ::= <expression>,<simple-expression>
<simple-expression> ::= <action>|<plan>|<expression>
<opr> ::= SEQ|OR|XOR|AND|SIM|PAR
<plan> ::= object of the type Plan3
<action> ::= object of the type Action

```

Os operadores SEQ, OR, XOR, AND, SIM, PAR permitem estabelecer relações temporais (precedência) e/ou lógicas entre os sub-planos e/ou ações que compõem um plano. Abaixo temos a descrição de cada um dos operadores citados:

- i. OR: indica que pelo menos um dos sub-planos e/ou ações de um plano deve ser executado.
- ii. XOR: indica que apenas um dos sub-planos e/ou ações de um plano deve ser executado.

³ Os conceitos dinâmicos (Action, Plan e Process) serão descritos na próxima seção.

- iii. AND: indica que todos os sub-planos e/ou ações de um plano devem ser executados não importando a ordem de execução das mesmas.
- iv. SEQ: Os sub-planos e/ou ações de um plano devem ser executados em seqüência.
- v. SIM: indica que os sub-planos e/ou ações de um plano podem ser executados simultaneamente.
- vi. PAR: indica que os sub-planos e/ou ações de um plano são executados paralelamente, não necessariamente ao mesmo tempo.

A classe **Situation** define os objetos e as restrições sobre esses objetos antes e depois da realização de uma ação ou plano e as maneiras de chegar a esse estado. Além dos atributos herdados da classe **StaticConcept**, a classe **Situation** possui os atributos *Objects*, *Restrictions* e *howToObtain* descritos a seguir :

- i. *Objects*: Objetos do domínio da tarefa que se está modelando. Este atributo é representado por uma lista (vetor) de objetos.
- ii. *Restrictions*: É uma expressão lógica sobre os objetos referenciados na lista de objetos do atributo *Objects*.
- iii. *howToObtain*: É uma lista de planos ou ações que indicam as diversas maneiras de se atingir a situação.

2.2.2. Conceitos dinâmicos

A classe **DynamicConcept** é sub-classe da classe **Concept** e se decompõe em três outras classes, são elas: **Process**, **Action** and **Plan**. **DynamicConcept** possui além dos atributos herdados da classe **Concept**, os atributos *preSituation*, *posSituation* e *duration*, descritos a seguir:

- i. *preSituation*: Este atributo é do tipo **Situation** e define a situação inicial necessária a realização de um Plano, Ação ou Processo.
- ii. *posSituation*: Este atributo é do tipo **Situation** e define a situação final resultante da realização de um Plano, Ação ou Processo.

iii. *duration*: tempo de duração.

A classe **Process** define um grupo de situações observadas em instantes diferentes (dispostas em ordem cronológica parcial ou total) e permite registrar a execução de um plano. A classe **Process** possui os atributos herdados da classe **DynamicConcept** e o atributo *situations*, que é representado por uma lista (vetor) de **Situations**.

A classe **Plan** define uma tarefa de alto nível ou de nível intermediário. Um plano se decompõe em sub-planos e ações, podendo ser decomposto em vários níveis de abstração. A classe **Plan** possui onze novos atributos, além dos herdados da classe **DynamicConcept**. Esses novos atributos são descritos a seguir:

- i. *actions*: É representado por uma lista (vetor) de **Actions**. Essas ações serão executadas baseado nos operadores definidos na classe **Method**.
- ii. *sub-plans*: É representado por uma lista (vetor) de **Plans**. Esses sub-planos serão executadas baseado nos operadores definidos na classe **Method**.
- iii. *howToObtain*: Representado por uma lista de **Methods**. Indica como o plano será executado.
- iv. *number*: número que identifica o plano (sub-plano) na árvore de tarefas em que ela se encontra.
- v. *ocurrence*: Determina a quantidade de vezes que o plano será executado. Pode assumir um dos seguintes valores:
 - (0,0) – O plano não é executado.
 - (0,1) – Executado nenhuma vez ou uma vez.
 - (0,n) – Executado zero ou mais vezes.
 - (1,1) – Executado uma única vez.
 - (1,n) – Executado pelo menos uma vez.

A classe **Action** descreve uma tarefa elementar. Na hierarquia de planos e ações, as ações seriam as folhas da árvore, isto é, não podem ser mais decompostas. Além dos atributos herdados da classe **DynamicConcepts**, a classe **Action** possui os seguintes atributos:

- i. number*: número que identifica a ação na árvore de tarefas em que ela se encontra.
- ii. occurrence*: Determina a quantidade de vezes que a ação será executada. Pode assumir um dos seguintes valores:
 - (0,0) – O plano não é executado.
 - (0,1) – Executado nenhuma vez ou uma vez.
 - (0,n) – Executado zero ou mais vezes.
 - (1,1) – Executado uma única vez.
 - (1,n) – Executado pelo menos uma vez.
- iii. agents*: Lista (vetor) de objetos do tipo **Agent**. Agentes que executam a ação.
- iv. tools*: Lista (vetor) de objetos do tipo **Tool**. Indica as ferramentas utilizadas pelos agentes para execução da ação.
- v. status*: Indica o estado da ação, podendo assumir os valores: em execução, finalizada, pausada, interrompida ou saltada.

2.3. Validação de TAOS face MAD

O formalismo TAOS, concebido inicialmente para aquisição e representação do conhecimento no domínio da biologia molecular, foi validado para a análise e modelagem da tarefa no contexto da concepção de interfaces homem-máquina (Kafure, 2000), sendo capaz de evidenciar os requisitos exigidos para esta análise, entre os quais:

- i.* Os objetivos da tarefa, ou seja, o estado do mundo que se deseja atingir com a execução da tarefa.
- ii.* A decomposição hierárquica da tarefa em sub-tarefas e ações segundo Sebillote (1998), isto é, a explicitação do plano de ação do usuário para atingir os objetivos a que se propõe.
- iii.* Os objetos envolvidos, ou seja, informações que são utilizadas e produzidas em cada etapa da análise.

- iv. Os procedimentos utilizados, ou seja, as diferentes maneiras ou estratégias de se realizar a tarefa.
- v. As condições necessárias para a realização da tarefa, que dizem respeito às restrições sobre o estado do mundo que devem ser verificadas para que a tarefa possa ser iniciada e finalizada.
- vi. As incoerências e incompletude da descrição, ou seja, a construção incremental do modelo é guiada através de uma metodologia descendente-ascendente, ou seja, deve ser construída uma hierarquia de tarefas e ações, tendo a tarefa mais geral a ser executada no topo e as ações que permitem realizar a execução da tarefa nas folhas da árvore.

2.4. TAOS: Modelagem da tarefa

Hammouche (1995) definiu a nova versão do MAD, um modelo analítico de descrição de tarefas do usuário orientado à especificação de interface chamado MAD*. Além dos atributos existentes em MAD, MAD* define novos atributos orientados à especificação da interface e à simulação de um modelo, que foram agregados ao formalismo TAOS nas classes *Agent*, *Plan* e *Action*.

Na classe **Agent** foram agregados os atributos *taskExperience* e *similarSystemExperience*:

- i. *taskExperience* : Experiência que o agente tem acerca da ação que está realizando. Este atributo, que é do tipo cadeia de caracteres (String), assume o valor alta, média ou baixa.
- iii. *similarSystemExperience* : Experiência com sistemas similares ao que se está projetando (descrevendo). Este atributo, que é do tipo cadeia de caracteres (String), assume o valor alta, média ou baixa.

Nas classes *Action* e *Plan* foram agregados os seguintes atributos :

- i. *priority*: Esse atributo é representado por um inteiro e indica a prioridade da ação.

- ii. *interruptability*: Determina se a ação que está sendo executada pode ser interrompida ou não, e caso possa, como será o reinício da execução dessa ação. Esse atributo pode assumir um dos três valores descritos abaixo:
- Não Interrompível: A ação não pode ser interrompida.
 - Interrompível com reinício a partir do começo – se a ação for interrompida, o fluxo de sua execução deve ser retomada do início.
 - Interrompível com reinício em curso – se a ação for interrompida, seu fluxo de execução deve ser retomada de onde parou.
- iii. *type*: Esse atributo ajuda na especificação das formas de interação. Indica o tipo da ação e pode assumir um dos seguintes valores: mental, sensório-mental ou verbal.
- iv. *modality*: Esse atributo também ajuda na especificação das formas de interação. Pode assumir um dos seguintes valores: automática, manual ou interativa.
- v. *importance*: Indica a importância da ação e assume um dos seguintes valores: alta, média ou baixa.
- vi. *frequency*: Indica a frequência em que a ação é executada. Pode assumir um dos seguintes valores: alta, média ou baixa.

Como TAOS foi validado para análise e modelagem da tarefa no contexto de interfaces homem-máquina, os conceitos *Plan* e *Sub-plan* referentes à classe ***Plan*** do modelo TAOS e que são oriundos dos sistemas baseados em conhecimento (SBC), passaram a ser nomeadas como *Task* (*tarefas*) *Sub-Task* (*sub-tarefas*), de acordo com a nomenclatura dos projetistas de interfaces.

2.4. Considerações finais

Neste capítulo foi apresentado o formalismo TAOS. Foram descritos sua estrutura e seus conceitos. TAOS foi concebido para aquisição e representação do conhecimento e validado para análise e modelagem da tarefa no contexto de interfaces homem-máquina.

A descrição hierárquica da tarefa em sub-tarefas e ações (conceitos dinâmicos) utilizando o formalismo TAOS é representada graficamente na forma de uma árvore, cujos

nodos representam as tarefas, sub-tarefas e ações e os arcos representam as conexões com as tarefas superiores. Os frames são representados em TAOS por descritores (tabelas) que descrevem os atributos relativos aos conceitos estáticos. Um extrato de uma descrição TAOS pode ser encontrado no capítulo 4.

TAOS foi concebido para ser implementado em dois módulos: o módulo TAME e o módulo TAOS-Graph. O módulo TAME guia a construção de um modelo para um domínio baseando-se numa metodologia ascendente-descendente. O módulo TAOS-Graph permite a visualização gráfica do processo de modelagem. Trata-se de um editor de entidades estáticas e dinâmicas estruturadas de acordo com os princípios preconizados pelo módulo TAME.

O formalismo TAOS vem sendo utilizado como modelo de descrição de tarefas no âmbito da Universidade Federal de Campina Grande (UFCG). Porém, realizar a análise e modelagem de tarefas sem a disponibilidade de uma ferramenta específica que auxilie o projetista implica em problemas (Gamboa e Scapin, 1997), tais como:

- i.* Dificuldade em descrever a estrutura de árvores das tarefas e suas notações formais;
- ii.* Dificuldade de verificar a consistência e completude das tarefas modeladas;
- iii.* Impossibilidade de geração automática ou semi-automática do modelo da interação a partir do modelo da tarefa;

Pelas dificuldades citadas acima, foi projetada e implementada a ferramenta iTAOS, uma ferramenta para análise e modelagem de tarefas segundo o formalismo TAOS. A ferramenta iTAOS foi concebida seguindo o princípio da independência do diálogo, ou seja, o módulo TAME (componente funcional) foi projetado independentemente do módulo TAOS-Graph (componente interativo), seguindo processos distintos. O módulo TAOS-Graph, resultado desse trabalho, foi desenvolvido utilizando a metodologia MEDITE (Guerrero, 2001), que será apresentada no próximo capítulo.

Capítulo 3

3. MEDITE – Uma metodologia baseada na tarefa e orientada a modelos para concepção de interfaces ergonômicas

MEDITE (Guerrero, 2001, Guerrero e Lula, 2001a, Guerrero e Lula, 2001b, Guerrero e Lula, 2002) é uma metodologia baseada na tarefa e orientada a modelos que utiliza o conhecimento ergonômico logo no início do processo de concepção de interfaces de forma a produzir interfaces que reflitam os objetivos, as características e as necessidades do usuário.

MEDITE (MAD* + EEDITOR + ERGONOMIA) guia o projetista, especialmente aquele que não tem conhecimento sobre Ergonomia, segundo modelos bem definidos durante todo o processo de concepção de interfaces.

Neste capítulo descrevemos o ciclo de desenvolvimento definido por MEDITE, os modelos da tarefa, da interface e da arquitetura utilizados na metodologia, a base de regras ergonômicas proposta inicialmente e o suporte computacional (ferramentas) para cada etapa do processo de concepção.

3.1. Descrição de MEDITE

A maioria das metodologias tradicionais de desenvolvimento de software não se preocupa com a interface do usuário, que passa a ser considerada apenas nas fases finais do processo, tornando-se assim nada mais que um “apêndice” do sistema. Devido à necessidade de desenvolver aplicações utilizáveis, acessíveis a um grande número de usuários, foram surgindo várias metodologias e ferramentas de ajuda à concepção de interfaces para o usuário.

As metodologias baseadas na tarefa têm se mostrado um caminho efetivo no apoio à concepção de interfaces ergonômicas. Partindo da análise e modelagem da tarefa, do perfil do usuário e de princípios ergonômicos, é possível se construir interfaces que levam em conta os reais objetivos do usuário.

Entre essas várias metodologias que surgiram podemos citar TRIDENT (Bodart at al, 1994; Bodart at al, 1995), ADEPT (Johnson at al., 1993; Markopoulos, Pycock e Wilson, 1992; Wilson at al., 1993), ERGO-START (Hammouche, 1995), ALACIE (Gamboa, 1998) e MCI (Sousa, 1999). Essas metodologias, em geral, fazem uso de modelos (modelo da tarefa, modelo do usuário, modelo da interação, modelo arquitetural), de conhecimento ergonômico e de ferramentas computacionais para apoiar a concepção da interface.

No entanto, algumas dificuldades e limitações são sentidas na utilização das metodologias citadas acima, como a dificuldade de escolha e de utilização de regras ergonômicas em relação às etapas do processo, dificuldades na passagem do modelo da tarefa para o modelo da interação, dificuldade da aplicação da metodologia na ausência de ferramentas e dificuldades em modificar o código da interface (Guerrero, 2001).

Com o intuito de minimizar esses problemas e dificuldades no âmbito da UFCG foi desenvolvida uma metodologia baseada na tarefa e orientada a modelos denominada MEDITE. Essa metodologia parte da descrição da tarefa. Cada etapa do processo segue um modelo e o conhecimento ergonômico é integrado de forma a ser usado com facilidade através de uma base de regras que auxilia o projetista na passagem do modelo da tarefa para o modelo da interação. Devido a modelos bem definidos para cada artefato gerado, MEDITE pode ser utilizada independentemente de ferramentas computacionais.

MEDITE define o processo de construção de interfaces ergonômicas em 5 etapas: análise e modelagem da tarefa, especificação conceitual da interação, refinamento da especificação conceitual, geração do protótipo e avaliação distribuída em todas as outras etapas do processo. A Figura 2 ilustra os processos através de círculos e por meio de retângulos os produtos (artefatos) gerados e as ferramentas conceituais (modelos) utilizadas em cada etapa do processo.

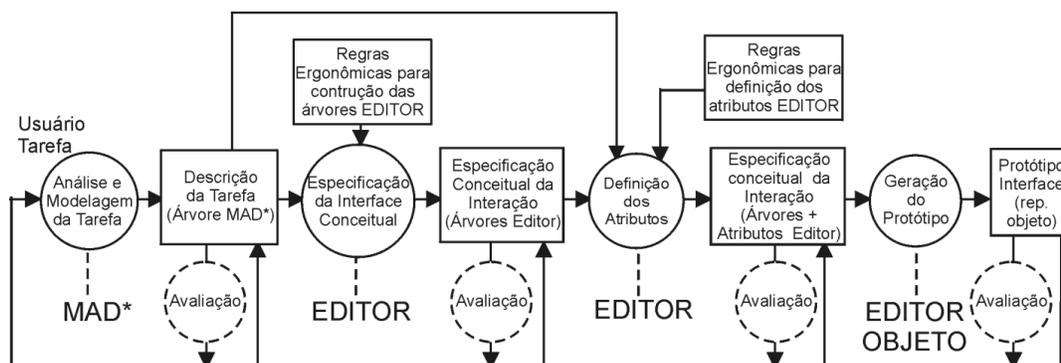


Figura 2. Metodologia MEDITE

3.1.1. Análise e modelagem da tarefa

Esta etapa pretende descrever os objetivos dos usuários e os meios utilizados no processo de realização de uma determinada tarefa. A tarefa deve ser descrita em termos de objetivos, procedimentos, objetos, decomposição em sub-tarefas, restrições, etc. O modelo utilizado na descrição da tarefa é o MAD*.

Esta etapa tem como entrada os dados sobre o usuário e sobre o domínio da tarefa. O produto gerado no final desta etapa é a descrição MAD* da tarefa (árvore MAD* e seus descritores). A avaliação desta etapa é realizada iterativamente e interativamente com o usuário, até que o usuário consiga visualizar o fluxo de realização da sua tarefa explícita na árvore e nos descritores do modelo. A consistência e completude da tarefa devem ser verificadas pelo projetista, bem como a escolha e modificações das tarefas de acordo com o sistema a ser desenvolvido.

3.1.2. Especificação conceitual parcial da interação

Esta etapa consiste em produzir a especificação conceitual parcial da interação. Determina a estrutura e o estilo dos aspectos da interação de acordo com o modelo EDITOR (Lula, 1992). Essa especificação permite uma visão inicial (esboço) da apresentação e do encadeamento do diálogo da interface em construção.

Esta etapa tem como entrada a descrição MAD* da tarefa gerada na etapa anterior e utiliza o conhecimento ergonômico armazenado na forma de regras de produção para

transformar a descrição MAD* da tarefa na especificação conceitual da interação segundo o modelo EDITOR.

A avaliação nesta etapa consiste em verificar se a especificação conceitual gerada através das regras ergonômicas está de acordo com uma primeira apreciação do usuário.

3.1.3. Definição dos atributos

Esta etapa consiste na definição dos atributos das árvores EDITOR, ou seja, o complemento e refinamento da especificação gerada na etapa anterior com a definição dos atributos de *Apresentação* (localização, formato, tamanho de fonte, etc.), de *Abstração* (em relação ao domínio da aplicação) e de *Controle*, definindo o encadeamento do diálogo entre as janelas da interface.

As entradas desta etapa são a especificação gerada na etapa anterior e a descrição MAD* da tarefa. O produto gerado é a especificação conceitual completa da interação de acordo com o modelo EDITOR e com o conhecimento ergonômico para definição dos atributos.

Nesta etapa, a especificação conceitual completa já permite uma melhor visualização das janelas da interface que se está projetando, que deve ser levada novamente ao usuário de forma a validar essa especificação.

3.1.4. Geração do protótipo

Esta etapa consiste da codificação da interface a partir da especificação conceitual completa definida na etapa anterior. Esse código é estruturado de acordo com o modelo de arquitetura definido no modelo EDITOR.

A avaliação do protótipo deve ser realizada junto ao usuário através de alguma(s) técnica(s) de avaliação, tais como: testes de usabilidade, inspeção por padrão, avaliação heurística, conformidade com recomendações, exploração cognitiva ou uma abordagem híbrida. Dependendo do resultado da avaliação, o projetista deve retornar a etapas anteriores, baseado no problema que for encontrado.

3.2. Modelo MAD*

O modelo MAD* é um modelo analítico de descrição de tarefas do usuário orientado à especificação de interface. Ele descreve e modela as tarefas de forma hierárquica e possui uma sintaxe gráfica de descrição (Hammouche, 1995). A figura 3 apresenta os quatro conceitos definidos por MAD*.



Figura 3. Conceitos definidos por MAD*

3.2.1. Conceitos definidos por MAD*

Unidade-tarefa

O conceito de unidade-tarefa é definido como uma estrutura de conhecimento que é: (i) unitário e uniforme, (ii) autônomo, isto é, permite analisar as tarefas individualmente e (iii) um instrumento de medida da complexidade de uma árvore de tarefas. Este conceito permite representar tanto tarefas elementares como tarefas compostas.

Em MAD* distinguem-se três níveis de unidade-tarefa: alto, intermediário e baixo. Uma unidade-tarefa de alto nível descreve um objetivo da tarefa. Uma unidade-tarefa de nível intermediário representa uma estratégia de procedimentos de execução da tarefa e uma unidade-tarefa de baixo nível corresponde a uma ação da tarefa.

Uma unidade-tarefa de alto nível pode agrupar diversas unidades tarefas de nível intermediário, assim como uma unidade tarefa de nível intermediário pode agrupar diversas unidades tarefas de baixo nível (ações).

Corpo da unidade tarefa

O corpo da unidade tarefa trata das condições de entrada e de saída e do corpo da tarefa. As condições de entrada e de saída possuem como principal atributo *pré-condições* e *pós-condições* respectivamente. As pré-condições e pós-condições exprimem as restrições sobre os objetos do estado inicial e final respectivamente.

O corpo da tarefa agrupa grande parte dos atributos MAD*, são eles: nome da tarefa (tipo alfanumérico), número da tarefa (tipo numérico), objetivo (texto explicando o objetivo da tarefa), modo da tarefa (facultativa, obrigatória), ator da tarefa (usuário, sistema, desconhecido), tempo (início da execução, fim da execução e a duração da execução), prioridade da tarefa, interruptibilidade da tarefa e estado de execução.

Decomposição da unidade tarefa

MAD* define dois tipos de relações de decomposição: estruturais e temporais. As relações estruturais permitem definir o ordenamento da tarefa e as relações temporais permitem definir a sincronização das tarefas. Descreveremos a seguir os construtores das relações estruturais e temporais:

Relações estruturais:

E – executar todas as sub-tarefas.

OU – executar pelo menos uma das sub-tarefas.

ALT – executar somente uma das sub-tarefas.

Relações temporais:

SEQ – executar as tarefas seqüencialmente.

PAR – executar as tarefas paralelamente.

SIM – executar as tarefas ao mesmo tempo.

Atributos orientados à especificação da interface

O quarto conceito de MAD* concerne os atributos que tem por objetivo integrar as recomendações orientadas à tarefa com aquelas encontradas nas regras ergonômicas,

permitindo orientar as primeiras decisões de concepção de uma futura interface. Os atributos orientados a concepção da interface são adicionados ao nível do corpo ou da decomposição da unidade-tarefa. São eles:

- i.* Restrições do diálogo: permitem refinar o sequenciamento das sub-tarefas de uma tarefa, notadamente pelas relações E e PAR.
- ii.* Tipo e modalidade da tarefa: Os valores que o atributo tipo pode receber são *mental*, *sensorio-motor*, *cognitivo*, *verbal* etc., enquanto que o atributo modalidade da tarefa pode assumir os valores *manual*, *automático* e *interativo*.
- iii.* Papel do operador: Competência e experiência do operador para realizar a tarefa. Os seguintes atributos o definem:
- iv.* Centralidade da tarefa: Este atributo tenta explicitar a maior importância de uma tarefa em detrimento a outras. Assume o valor *importante*, *mediano* ou *secundário*. Os dois atributos descritos a seguir o definem:
 - Frequência: {*elevada*, *mediana*, *fraca*}.
 - Entidades importantes: {sub-conjunto dos objetos da tarefa}.

3.3. Modelo EDITOR

O modelo EDITOR é um modelo multi-agente utilizado como modelo unificado da arquitetura e de interação na metodologia MEDITE. Ele utiliza o modelo PAC (Coutaz, 1987; Coutaz, 1990) como modelo de arquitetura e adota o conceito de edição como modelo conceitual de interação (Smith, Barth e Young, 1987).

EDITOR é um modelo de arquitetura que alia os aspectos remarcáveis do modelo de Seeheim (Pfaff, 1985) às vantagens de um modelo multi-agente (Coutaz, 1990). Um sistema interativo é um agente PAC composto, ou seja, se decompõe em agentes intermediários até os agentes PAC elementares, estruturando a arquitetura de um sistema interativo como uma hierarquia de agentes PAC. Todo agente PAC possui três facetas: *Apresentação*, *Abstração* e *Controle*. As facetas são distribuídas a todos os níveis de abstração.

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 as regras que permitam ao projetista identificar os seus componentes a partir da análise do domínio e/ou da tarefa.

Quanto ao modelo conceitual da interação, o modelo EDITOR adota o conceito de edição como paradigma central. Pode ser considerado um processo de edição 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 esses estados.

O modelo EDITOR une as vantagens arquiteturais do modelo PAC com as vantagens de um modelo conceitual de interação, permitindo que seus elementos sejam identificados a partir da descrição da tarefa através de regras claras e objetivas. EDITOR encontra no paradigma orientado a objeto um modelo computacional adequado para sua representação e realização.

3.3.1. Agentes do modelo EDITOR

O modelo EDITOR define uma interface (editor) composta por três agentes do tipo PAC, são eles: *Editor*, *Visão* e *Objeto de Interação*, onde cada um deles apresenta as facetas *Apresentação*, *Abstração* e *Controle* obrigatoriamente, distribuídas a todos os níveis da hierarquia.

A faceta *Apresentação* define a forma como o agente é percebido, sendo associado a um objeto gráfico de uma biblioteca virtual gráfica. A faceta *Abstração* define um método funcional da aplicação associado ao agente. A faceta *Controle* define o diálogo entre as facetas *Apresentação* e *Abstração*, mantendo a coerência entre elas. Vejamos a organização hierárquica dos agentes de um Editor na Figura 4:

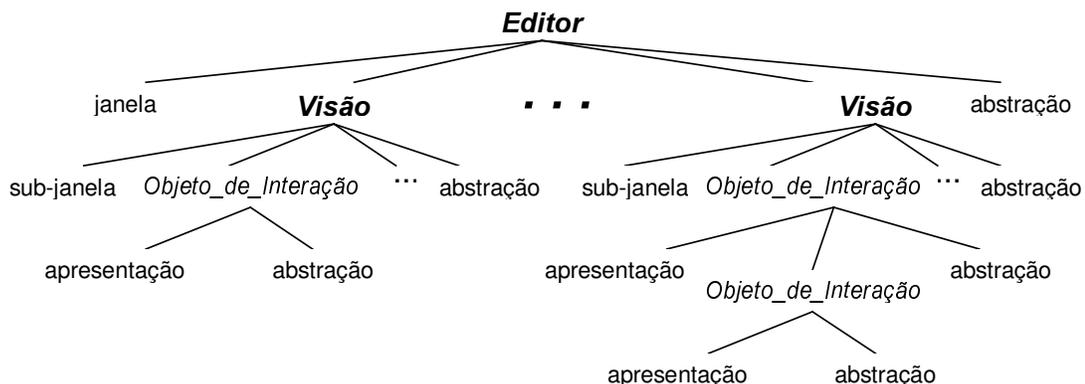


Figura 4. Organização dos agentes de um Editor

Um agente *Editor* é um agente PAC composto que define um conjunto de visões sobre a aplicação (agentes *Visão*) apresentadas e organizadas em uma janela na tela. 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 os eventos trocados entre a faceta *Apresentação* e *Abstração*.

Um agente *Visão* é um agente PAC composto que define um conjunto de objetos de interação (agentes *Objetos de Interação*) agrupados. Sua faceta *Apresentação* é definida a partir de um objeto sub-janela ou barra de menu da biblioteca virtual e também pelos eventos recebidos do usuário. A faceta *Abstração* permite designar um objeto da aplicação e de definir as modalidades de comunicação com ele. A faceta *Controle* gerencia os eventos trocados entre a faceta *Apresentação* e *Abstração*, como também entre os *Objetos 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. Sua faceta *Apresentação* é definida a partir de um objeto específico de uma biblioteca virtual e também pelos eventos que o objeto de interação pode receber do usuário. 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. Os agentes *Objetos de Interação* podem ser simples e complexos, como visto na Figura 4.

Como podemos observar, uma especificação conceitual da interação segundo o modelo EDITOR é uma árvore que espelha a composição gráfica da interface a ser gerada, permitindo uma visualização (esboço) antecipada da sua apresentação.

3.4. Base de regras ergonômicas

Trataremos nesta seção da base de regras ergonômicas de auxílio à concepção de interfaces utilizada na metodologia MEDITE. São regras de produção do tipo (SE - ENTÃO) extraídas de guias de recomendações, normas, padrões e guias de estilos ergonômicos com o objetivo de auxiliar o projetista durante a passagem da descrição da tarefa para a especificação da interação.

A base de regras contém dois tipos de regras: (i) regras para a construção da árvore EDITOR e (ii) regras para definição dos atributos. As regras do tipo (i) relacionam elementos do modelo da tarefa (MAD*) aos elementos do modelo da interação (EDITOR). Estas regras são referentes à especificação conceitual inicial da interação, enquanto que as regras do tipo (ii) definem os atributos das facetas de cada agente: *Apresentação* (localização, formato, tipo e tamanho da fonte); *Abstração* (em relação ao domínio da aplicação); e *Controle* que define o diálogo.

Exemplo de uma regra do tipo (i):

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

Exemplo de uma regra do tipo (ii):

SE a tarefa for de alto nível (MAD* ou TAOS) 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.

As recomendações ergonômicas estão sujeitas a controvérsias, podendo ser contraditórias em determinados contextos. Porém, as regras oriundas dessas recomendações ajudam enormemente aos projetistas, especialmente aqueles (maioria) que não possuem conhecimento em Ergonomia. A base de regras ergonômicas é uma base de conhecimento e, portanto pode ser refinada ao longo do tempo e de seu uso.

3.5. Suporte computacional (Ferramentas)

A maioria das metodologias baseada na tarefa para o projeto de interfaces do usuário apresenta dificuldades de utilização quando não há a disponibilidade ou mesmo a existência de ferramentas que auxiliem a construção das especificações, podendo este fato inviabilizar sua utilização.

MEDITE possui como uma de suas características a independência de ferramentas computacionais, ou seja, todo o processo de concepção de uma interface pode ser feito independente de ferramentas, visto que todas as etapas do processo de concepção produzem artefatos de acordo com modelos bem definidos.

Existem ferramentas disponíveis que poderiam ser incorporados à metodologia. Na primeira etapa, a ferramenta MAD*-toolkit poderia ser utilizada para gerar a descrição MAD* da tarefa. Na segunda, terceira e quarta etapas de MEDITE, onde a especificação da interação, seu refinamento e a construção do protótipo são realizados respectivamente, a ferramenta EDITOR OBJETO poderia ser utilizada, especificando a interação da interface através das árvores EDITOR e realizando a implementação objeto dessas árvores, visto que elas são representadas por uma estrutura orientada a objetos.

3.6. Considerações finais

A metodologia MEDITE vem sendo utilizada na concepção de interfaces homem-máquina no âmbito da Universidade Federal de Campina Grande. Como dito no capítulo introdutório, realizar a análise e modelagem de tarefas, primeira etapa de MEDITE, sem a disponibilidade de uma ferramenta específica que auxilie o projetista implica em problemas, tais como: dificuldade em descrever a estrutura de árvores da tarefa, dificuldade em verificar a consistência e completude da tarefa descrita e a impossibilidade de geração automática ou semi-automática da especificação da interação.

Devido à dificuldade de aquisição de uma ferramenta para descrição de tarefas, como, por exemplo, MAD*-toolkit, o GIHM trabalhou na geração de uma ferramenta específica para este propósito. A ferramenta iTAOS, em parte fruto deste trabalho, implementa o formalismo TAOS para análise e modelagem da tarefa. Devido à sua equivalência com o modelo MAD

(Kafure, 2000), acrescentado dos atributos orientados à especificação da interface e à simulação do modelo de MAD*, podemos substituir MAD* por TAOS como modelo da tarefa na metodologia MEDITE.

Embora MEDITE pressuponha MAD*, qualquer modelo da tarefa que tenha equivalência com MAD* pode ser utilizado. O grupo de desenvolvimento de MEDITE é quem está investindo em iTAOS, o que levará a uma substituição espontânea de MAD* por TAOS.

Algumas vantagens em relação a outros métodos de concepção de interfaces nos levaram a adotar MEDITE como processo de concepção do componente interativo (TAOS-Graph) da ferramenta iTAOS para análise e modelagem de tarefas, entre elas:

- Facilidade em selecionar e aplicar regras ergonômicas por estarem classificadas de acordo com as etapas do processo;
- Facilidade em passar da descrição da tarefa à especificação da interação por utilizar um modelo da interação explícito e simples;
- Ser independente de ferramentas, devido ao uso de modelos conceituais bem definidos;
- A especificação da interação segundo o modelo EDITOR proporciona uma descrição gráfica clara e intuitiva da interface ao projetista.

Nos próximos quatro capítulos descreveremos a utilização da metodologia MEDITE para a concepção da interface (TAOS-Graph) da ferramenta iTAOS.

4.2. Análise e Modelagem da Tarefa

Nesta seção descrevemos o processo de aquisição dos dados sobre a tarefa e sobre o usuário e posteriormente a descrição dessa tarefa utilizando o formalismo TAOS. A descrição completa da tarefa “*Utilizar a ferramenta iTAOS para analisar e modelar tarefas*” pode ser encontrada em Medeiros, Cordeiro e Lula (2002a).

4.2.1. Aquisição de dados sobre a tarefa e sobre o usuário

O processo de coleta de dados sobre a tarefa *Utilizar a ferramenta iTAOS para analisar e modelar tarefas* e sobre o usuário foi realizado utilizando entrevistas semidirigidas, a técnica *trace analysis* e questionários.

A técnica de entrevista semidirigida utiliza o procedimento de perguntas e respostas (Graesser, 1978), ou “why and how” (Sebillote, 1991). Baseado nas questões do tipo *Why and How* que iam sendo levantadas e nas respostas dadas e analisadas, foi possível identificar quais tarefas eram de nível alto ou intermediário (planos) ou de baixo nível (ações), possibilitando que a tarefa fosse descrita hierarquicamente (Sebillote, 1988).

Em paralelo com as entrevistas realizadas, foi utilizada a técnica *trace analysis*, que consiste de um processo de coleta de dados através da observação de usuários realizando a tarefa e de consultas a documentos. Essa técnica foi utilizada com o intuito de complementar a descrição da tarefa e para isso foram realizadas consultas a manuais e livros sobre: análise de tarefas, TAOS e ferramentas para análise de tarefas, permitindo que novos elementos fossem agregados a descrição e que alguns elementos já levantados fossem checados.

Finalmente, utilizamos o questionário DePerUse (Queiroz, 2001) para obtenção de dados sobre o usuário da tarefa *Utilizar a ferramenta iTAOS para analisar e modelar tarefas* (ver Apêndice 1). Este questionário foi aplicado a 22 estudantes de graduação ou pós-graduação que já analisaram e modelaram tarefas utilizando o formalismo MAD ou o formalismo TAOS. Do questionário aplicado e das entrevistas semidirigidas realizadas, traçamos o perfil do usuário (Turnell, 1998) (ver Apêndice 2). O perfil traçado do usuário

aborda as seguintes questões: grau de instrução, habilidades para realização da tarefa, motivações, faixa etária, experiência com a tarefa ou com sistemas similares, etc.

4.2.2. Descrição da tarefa utilizando TAOS

Apresentamos abaixo parte da descrição TAOS da tarefa *Utilizar a ferramenta iTAOS para analisar e modelar tarefas*. Esta descrição foi oriunda dos dados sobre a tarefa e sobre o usuário levantados através das técnicas descritas anteriormente. A descrição completa da tarefa pode ser encontrada em Medeiros, Cordeiro e Lula (2002a).

Na figura 5 a seguir, apresentamos um trecho da árvore de modelagem que representa os níveis mais altos da hierarquia, ou seja, os objetivos gerais da tarefa *Utilizar a ferramenta iTAOS para analisar e modelar tarefas*. Essa descrição TAOS da tarefa expressou fielmente os dados obtidos das entrevistas, dos questionários e da técnica trace analysis, o que significa afirmar que o usuário que deseja modelar tarefas utilizando a ferramenta iTAOS tem que executar pelo menos uma das seguintes sub-tarefas: ativar ambiente de modelagem, realizar a modelagem da tarefa, verificar o modelo criado, simular o modelo criado, alternar entre os modelos disponíveis e abertos, solicitar ajuda e desativar ambiente de modelagem, segundo sua própria vontade, como podemos observar na figura 6.

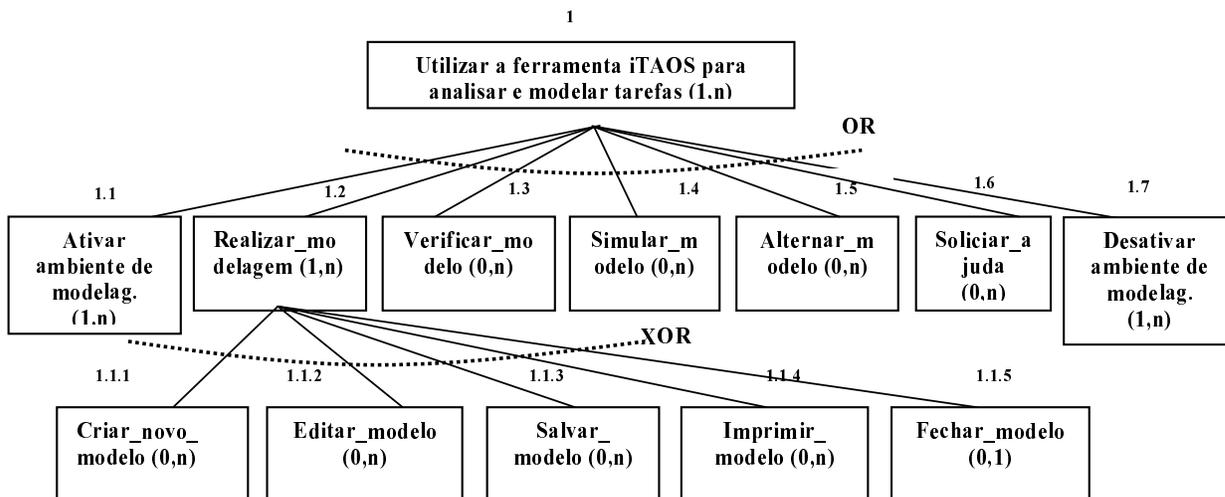


Figura 6. Tarefa “Modelar_tarefas_utilizando_iTAOS”

Na figura 7 a seguir, apresentamos um trecho da árvore que representa a sub-tarefa *Editar Agente*. O usuário que deseja editar um agente deve executar uma ou mais das seguintes ações: Editar nome do agente, Editar descrição do agente, Editar competência,

Editar experiência com a tarefa, Editar experiência com um sistema similar, Editar meios de interação.

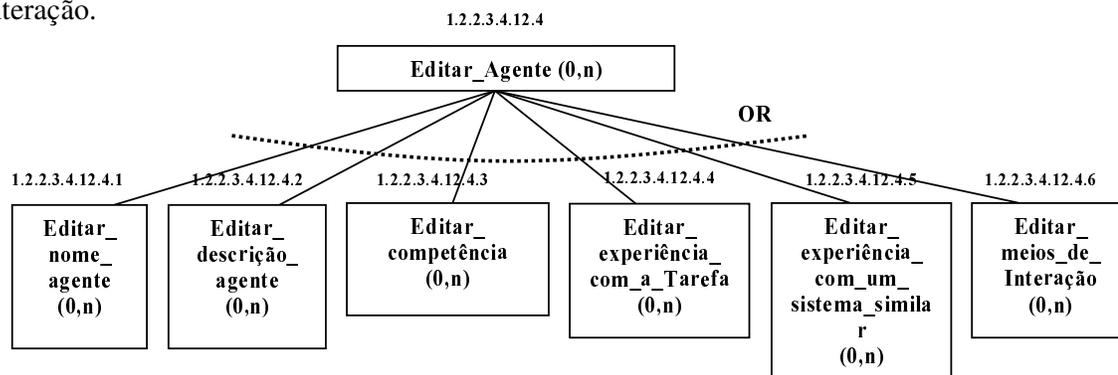


Figura 7. Tarefa “Editar_Agente”

Após a apresentação de dois trechos da árvore TAOS da descrição *Utilizar a ferramenta iTAOS para analisar e modelar tarefas*, mostraremos os descritores relativos a tarefa *Realizar_modelagem* e a ação *Editar_nome_agente* respectivamente. A escolha dessa tarefa e dessa ação ocorreu de forma aleatória. Os descritores definem os elementos e atributos das tarefas e ações da descrição gerada.

Além dos atributos nome, número e ocorrência já encontrados em cada tarefa e em cada ação na própria árvore TAOS de modelagem, as tarefas e ações possuem atributos, objetos e procedimentos que são armazenados nos descritores. A seguir mostramos os descritores relativos a tarefa *Realizar_modelagem*:

Classe	Task
Name	Realizar_modelagem
Description	Realizar a modelagem, criar as árvores e descritores
Pre situation	Situação_Inicial_de_Realizar_Modelagem
Pos situation	Situação_Objetivo_de_Realizar_Modelagem
Ocurrence	(0,n)
Actions	[Salvar_Modelo, Imprimir_Modelo, Fechar_Modelo]
Sub-plans	[Criar_um_Novo_Modelo, Editar_Modelo_Existente]
How to realize	Método_Realizar_Modelagem
Priority	() 1 () 2 (x) 3
Interruptibility	() Não interrompível () Interruptível com reprise no início (x) Interruptível com reprise em curso
ATRIBUTOS ORIENTADOS A ESPECIFICAÇÃO DA INTERFACE	
Type of the task	(x) sensório-motor (x) mental (x) verbal () Outro
Modality of the task	() manual () automático (x) interativo
Importance of the task	(x) importante () mediana () secundária
Frequence of the task	() elevada (x) mediana () fraca

Descritor 1: Classe Tarefa *Realizar_modelagem*

Classe	Situação
Name	Situação_Inicial_de_Realizar_Modelagem
Description	Situação que deve ser satisfeita para o início da realização de uma modelagem da tarefa
Objects	[Ambiente_de_Modelagem, Mouse, Teclado, Projetista_de_Interface]
Restriction	ativado(Ambiente_de_Modelagem), disponível(Mouse), disponível(Teclado), disponível(Projetista_de_Interface)
Como Atingir	Ativar_Ambiente_de_Modelagem

Descritor 2. Classe Situação Situação_Inicial_de_Realizar_Modelagem

Classe	Situação
Name	Situação_Objetoivo_de_Realizar_Modelagem
Description	Situação depois da tarefa realizar modelagem
Objects	Modelo, Mouse, Teclado, Projetista_de_Interface
Restriction	OR(XOR(criado(Modelo),editado(Modelo)),salvo(Modelo),impresso(Modelo), Fechado(Modelo)), disponível(Mouse), disponível(Teclado), Disponível(Projetista_de_Interface))
Hot to obtain	Realizar_Modelagem

Descritor 3. Classe Situação Situação_Objetoivo_de_Realizar_Modelagem

Classe	Método
Name	Método_Realizar_Modelagem
Description	Estratégia para Modelar modelagem
Body	OR(XOR(Criar_um_Modelo_Existente, Editar_um_Modelo_Existente), Salvar_Modelo, Imprimir_Modelo, Fechar_Modelo)

Descritor 4. Classe Método Método_Realizar_Modelagem

Descritores da ação *Editar_nome_agente*.

Classe	Ação
Name	Editar_nome_agente
Description	Editar o nome de um agente
Pre situation	Situação_Objetoivo_de_Selecionar_Link_para_Agente
Pos situation	Situação_Objetoivo_de_Editar_Agente
Ocurrence	(0,n)
Agent	Projetista de interface
Tool	Ambiente de modelagem
Priority	(x) 1 () 2 () 3
Interruptibility	() Não interrompível () Interruptível com reprise no início (x) Interruptível com reprise em curso
ATRIBUTOS ORIENTADOS A ESPECIFICAÇÃO DA INTERFACE	
Type of the task	(x) sensorio-motor () mental () verbal Outro:
Modality of the task	(x) manual () automático () interativo
Importance of the task	(x) importante () mediana () secundária
Frequence of the task	(x) elevada () mediana () fraca

Descritor 5: Classe Ação Editar_nome_agente

Classe	Situação
Name	Situação_Inicial_de_Editar_Nome_Agente
Description	Pós situação da ação Selecionar Link para Agente
Objects	[atributos_agente]

Restriction	Exibido(atributos_agente)
How to obtain	-

Descritor 6. Classe Situação *Situação_Inicial_de_Editar_Nome_Agente*

Classe	Situação
Name	Situação_Objetivo_de_Editar_Nome_Agente
Description	Pré Situação do plano Editar Agente
Objects	[agente]
Restriction	Editado(agente)
How to obtain	-

Descritor 7. Classe Situação *Situação_Objetivo_de_Editar_Nome_Agente*

Classe	Agente
Name	Projetista_de_Interface
Description	Projetista de interface que modela tarefas
Competence	Todas as ações de plano Modelar tarefas
taskExperience	() iniciante (x) meio () especialista
similarSystemExperience	(x) elementar () médio () complexo

Descritor 8: Classe Agente *Projetista_de_Interface*

Classe	Instrumento
Name	Ambiente_de_Modelagem
Description	Ambiente utilizado pelo projetista de interface para modelar a tarefa
User	Projetista_de_Interface
Utility	Todas as ações do plano Modelar tarefas

Descritor 9: Classe Instrumento *Ambiente_de_Modelagem*

4.3. Avaliação da modelagem

A avaliação da descrição TAOS da tarefa foi realizada junto aos usuários concomitantemente com a modelagem da tarefa. Por se tratar de um processo iterativo e iterativo, à medida que a coleta dos dados sobre o usuário e sobre a tarefa era realizada, a árvore e os descritores TAOS da tarefa eram construídos em conjunto com os usuários. Frequentemente, os usuários avaliavam a descrição que estava sendo construída e essa avaliação era então discutida com os projetistas ensejando eventualmente possíveis modificações.

A avaliação procurou verificar que as árvores e os descritores correspondiam à lógica de execução da tarefa segundo os usuários. Foi verificada também a consistência das pré e pós-condições e a completude do modelo, além de remover as tarefas não informatizáveis, sempre focando na utilização dessa descrição para a concepção da ferramenta iTAOS.

4.4. Considerações finais

Neste capítulo foi apresentada a descrição TAOS da tarefa *Utilizar a ferramenta iTAOS para analisar e modelar tarefas*. Essa descrição corresponde a primeira etapa da metodologia MEDITE, que foi utilizada para concepção do módulo interativo (TAOS-Graph) da ferramenta iTAOS.

A descrição da tarefa *Utilizar a ferramenta iTAOS para analisar e modelar tarefas* possuiu como entrada dados sobre o usuário e sobre a tarefa obtidos através de técnicas de coleta de dados como entrevistas semidirigidas e *trace analysis* e de questionários. Tratou-se de um processo interativo e iterativo pelo fato de que os usuários estiveram sempre presentes disponibilizando informações para que a tarefa fosse descrita segundo o formalismo TAOS. À medida que a árvore e os descritores TAOS da tarefa eram construídos, o usuário os avaliava. Essa avaliação foi realizada periodicamente e em paralelo com a modelagem.

A partir do questionário DePerUse aplicado a 22 pessoas, foram obtidas informações sobre o usuário que ajudou a levantar seu perfil. As informações contidas no perfil do usuário (Queiroz, 1998), juntamente com as entrevistas semidirigidas e observações, foram representados nos descritores *Agente* da descrição TAOS da tarefa. O descritor *Agente* possui os seguintes atributos: experiência com a tarefa, experiência com um sistema similar e os meios de interação para execução a tarefa.

No próximo capítulo apresentaremos o processo de concepção da especificação completa da interação. A especificação da interação possuiu como entrada a descrição da tarefa mostrada neste capítulo e utilizou o conhecimento ergonômico armazenado numa base de regras ergonômicas para transformação das árvores TAOS da tarefa nas árvores EDITOR da interação.

Capítulo 5

5. Especificação conceitual da interação

Neste capítulo descrevemos o processo de obtenção da especificação conceitual da interação a partir da descrição da tarefa *Utilizar a ferramenta iTAOS para analisar e modelar tarefas*. Trata-se do processo de construção das árvores EDITOR, nesta etapa cada agente do modelo EDITOR (*Editor, Visão e Objeto de Interação*) é definido.

Situaremos inicialmente a especificação conceitual dentro da metodologia MEDITE, depois descreveremos a especificação conceitual parcial da interação, que consiste da construção das árvores EDITOR a partir das árvores TAOS da tarefa e posteriormente apresentaremos a especificação completa da interação, onde os atributos relativos a *Apresentação, Abstração e Controle* são definidos e anexados as árvores. A especificação completa da interação pode ser encontrada em Medeiros, Cordeiro e Lula (2002b).

5.1. Especificação conceitual da interação na metodologia MEDITE

A especificação conceitual da interação corresponde à segunda e à terceira etapa da metodologia MEDITE como mostra a figura abaixo:

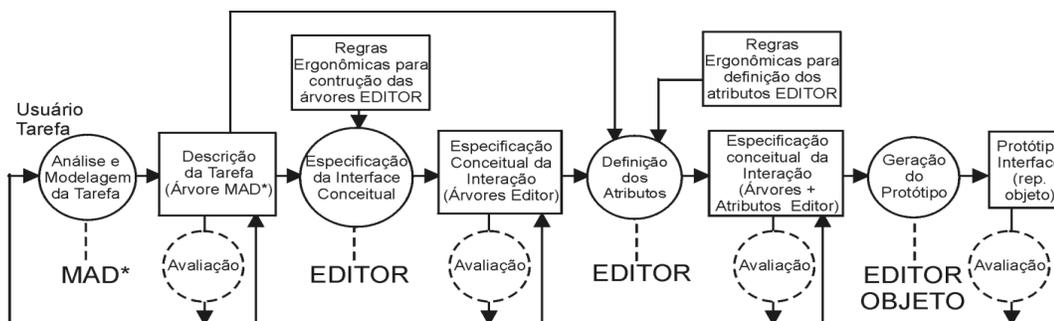


Figura 8. Especificação conceitual da interação na metodologia MEDITE

5.2. Especificação conceitual parcial da interação

Esta etapa teve como entrada a descrição TAOS da tarefa e possuiu como saída a especificação conceitual parcial da interação, ou seja, as árvores parciais EDITOR. Nesse momento ocorreu a primeira inserção do conhecimento ergonômico, representado sob a forma de regras de produção, no processo de transformação das árvores TAOS (descrição da tarefa) nas árvores EDITOR (especificação da interação).

Descrevemos o processo de construção de algumas árvores EDITOR explorando a transformação das árvores TAOS nas árvores EDITOR com o auxílio das regras ergonômicas. Apresentamos a seguir alguns trechos das árvores EDITOR geradas nesta etapa. No texto, essas árvores EDITOR serão sempre precedidas pela árvore TAOS da tarefa equivalente e das regras ergonômicas utilizadas para fazer a transformação.

Mostramos, inicialmente, a transformação dos dois primeiros níveis da árvore TAOS da tarefa *Utilizar a ferramenta iTAOS para analisar e modelar tarefas* (ver figura 5)⁴ na árvore EDITOR correspondente (especificação conceitual parcial da interação). Partindo da árvore TAOS e com o auxílio de regras ergonômicas foi construída a árvore EDITOR da figura 9. As seguintes regras foram utilizadas:

Regra 1: SE a tarefa for de alto nível (TAOS) ENTÃO definir um *Editor* (Modelo Editor) cuja *Apresentação* deve ser do tipo *Janela*.

Regra 2: SE as tarefas forem ligadas pelo construtor OU (TAOS) ENTÃO definir para esse conjunto de tarefas uma *Visão* (Modelo Editor) cuja *Apresentação* deve ser do tipo *Barra de Menu* e cada tarefa deve ser um Objeto de Interação (Modelo Editor) do tipo *item-menu* dessa *Visão*.

Regra 3. 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).

Pelo fato da tarefa *Utilizar a ferramenta iTAOS para analisar e modelar tarefas* ser de alto nível, foi definido um editor *Editor Modelar tarefa* cuja *faceta Apresentação* é do tipo *Janela* (Regra 1). Como as sub-tarefas (sub-planos e ações) da tarefa *Utilizar a ferramenta iTAOS para analisar e modelar tarefas* são ligados por um construtor *OU*, foi criada uma

⁴ Na figura 5 foram apresentados três níveis da árvore TAOS da tarefa, porém mostramos a transformação na árvore EDITOR de somente dois desses níveis.

visão *Funcionalidades* cuja faceta *Apresentação* é do tipo barra de menu e cada sub-tarefa é um objeto de interação do tipo item-menu (Regra 2). Aplicando a regra 3 acima foi definida uma visão que expressa a orientação do usuário.

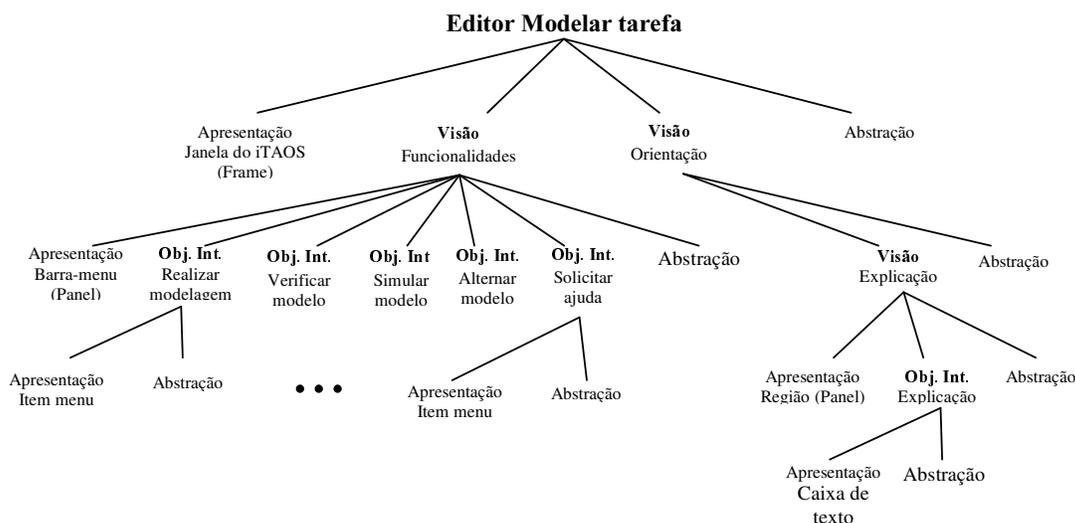


Figura 9. Árvore EDITOR parcial da tarefa "Modelar tarefa"

Apresentamos, a seguir, o processo de construção da árvore EDITOR correspondente a descrição TAOS da tarefa *Realizar Modelagem*, ou seja, o segundo e o terceiro nível da árvore TAOS apresentada na figura 5. A seguinte regras foi utilizada no processo de transformação:

Regra 4: 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).

De acordo com a regra acima, pelo fato da tarefa *Realizar modelagem* ser considerada importante e de frequência elevada, optamos por utilizar um menu hierárquico com as opções: Criar um novo modelo, Editar modelo existente, Salvar modelo, Imprimir modelo e Fechar modelo.

Vejamos na figura 10 a especificação conceitual parcial da interação da tarefa *Realizar modelagem*:

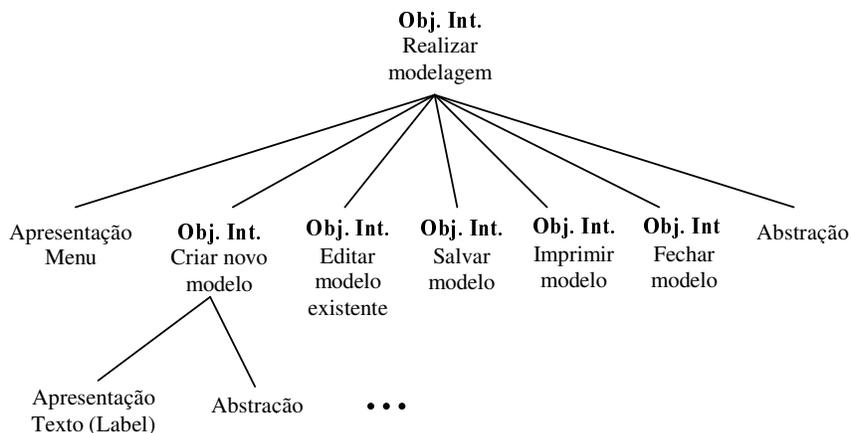


Figura 10. Árvore EDITOR parcial da tarefa "Realizar modelagem"

As regras ergonômicas disponíveis na primeira versão da base de regras ergonômicas não foram suficientes para que fosse realizada a transformação de toda a árvore TAOS da tarefa nas árvores EDITOR correspondentes. Por esse motivo, algumas regras foram modificadas e outras criadas baseando-se na nossa experiência com outras ferramentas equivalentes já existentes. A regra abaixo foi adaptada de uma regra já existente:

Regra 5: SE as tarefas forem ligadas pelo construtor OU e essas tarefas forem do tipo Editar ... (TAOS) ENTÃO definir para esse conjunto de tarefas uma Visão (Modelo Editor) cuja Apresentação deve ser do tipo Formulário e cada tarefa deve ser um Objeto de Interação (Modelo Editor) do tipo caixa de texto dessa Visão.



Figura 11. Árvore EDITOR parcial da tarefa "Editar agente"

A figura 11 apresenta a árvore EDITOR correspondente à tarefa *Editar agente* que foi obtida a partir do trecho da árvore TAOS mostrado no capítulo anterior na figura 6 utilizando a regra ergonômica acima.

O produto gerado nesta etapa é, portanto, o conjunto de árvores (parciais) EDITOR. Porém, as árvores ainda não estão completas, falta definir os atributos dessas árvores. Apresentaremos na próxima seção algumas árvores completas acrescentando os atributos referentes às facetas *Apresentação* (localização, formato, tamanho de fonte, etc.), *Abstração* (em relação ao domínio da aplicação) e *Controle*, definindo o encadeamento do diálogo entre as janelas da interface.

5.3. Especificação conceitual completa da interação

Esta etapa teve como entrada a especificação conceitual parcial da interação (árvores EDITOR parciais) e teve como saída a especificação conceitual completa da interação, ou seja, as árvores EDITOR completas. Nesta etapa, ocorreu a segunda inserção do conhecimento ergonômico, representado sob a forma de regras de produção, na definição dos atributos das facetas *Apresentação* (localização, formato, tamanho de fonte, etc.), *Abstração* (com relação ao domínio da aplicação) e *Controle*.

Apresentamos ,a seguir, novamente, as três árvores EDITOR apresentadas na seção anterior acrescidas dos atributos das facetas *Apresentação*, *Abstração* e *Controle*. Mostramos como foram definidos estes atributos através do conhecimento ergonômico representado pelas regras ergonômicas. Baseado nas regras ergonômicas abaixo, os atributos do *Editor Modelar tarefa* foram definidos:

Regra 6: SE a tarefa for de alto nível (TAOS) ENTÃO o tamanho, formato e localização da janela (atributos EDITOR) será o pré-definido pelo tipo de plataforma utilizado.

Regra 7: 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).

Regra 8: 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).

Regra 9: 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).

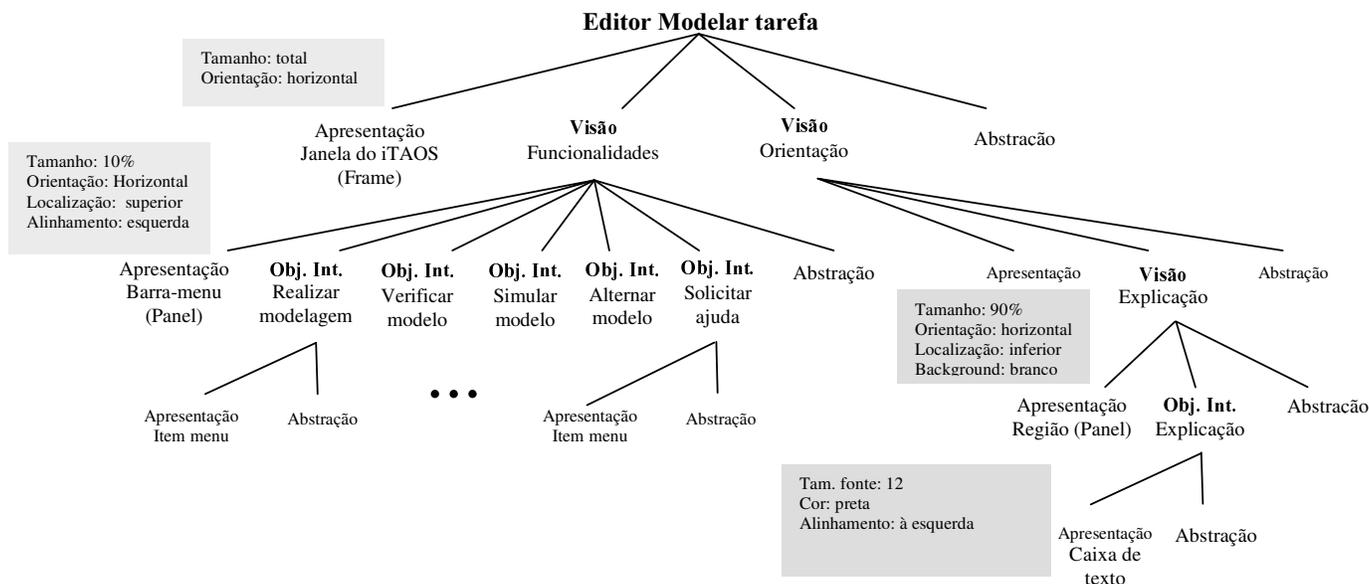


Figura 12. Árvore EDITOR completa da tarefa "Modelar tarefa"

O *Editor Modelar tarefa* possui como *Apresentação* uma janela com tamanho, formato e localização pré-definido pela plataforma utilizada, segundo a regra 6. Este editor define duas visões: a visão *Funcionalidades* e a visão *Orientação*. Como o número de opções a escolher na visão *Funcionalidades* é pequeno, foi definida uma barra de menu horizontal. Estas opções da barra de menu estão dispostas da esquerda para direita e no canto superior esquerdo como dizem as regras ergonômicas 7 e 8. A *Apresentação* do objeto de interação *Explicação* é uma caixa de texto com os atributos definidos segundo a regra ergonômica 9.

Mostramos a seguir um extrato da árvore EDITOR completa que representa o objeto de interação *Realizar modelagem*. A faceta *Abstração* dos objetos de interação que representam os itens de menu do menu Realizar modelagem acessam os métodos do módulo funcional da ferramenta iTAOS (módulo TAME).

Extrato da árvore EDITOR completa da tarefa *Realizar Modelagem*:

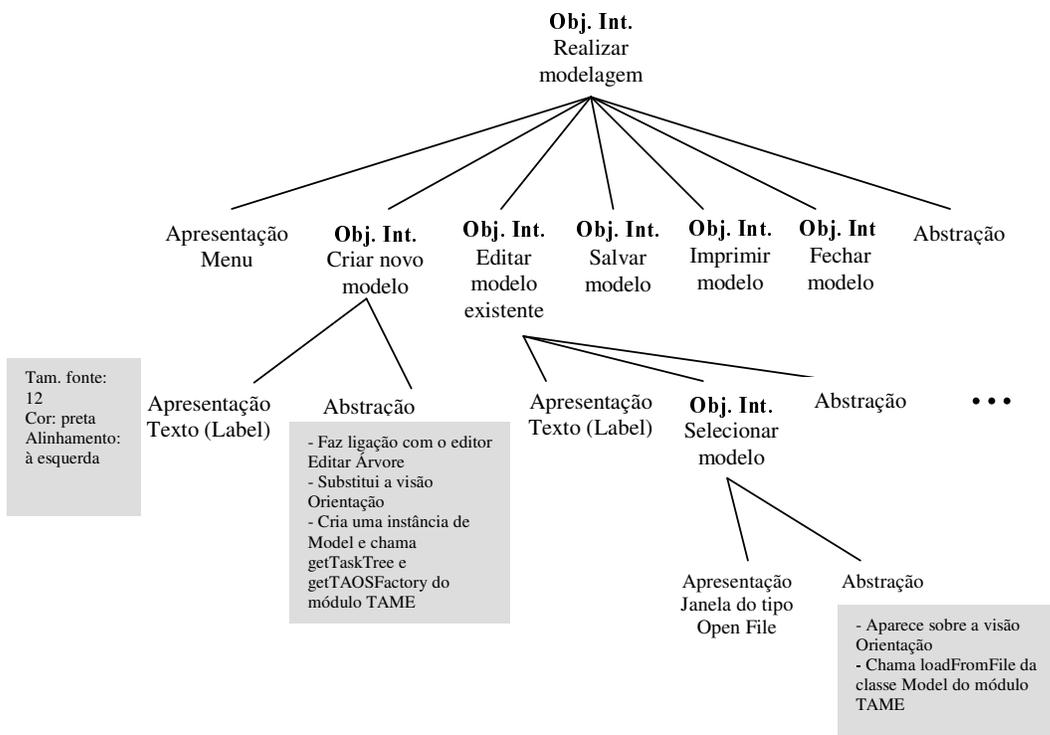


Figura 13. Árvore EDITOR completa da tarefa "Realizar modelagem"

Apresentamos na figura 14 a árvore EDITOR completa que representa a tarefa Editar agente. Utilizamos as seguintes regras ergonômicas para definição dos atributos dessa árvore:

Regra 11: 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).

Regra 12: 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).

Regra 13: 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).

3.3.1, uma árvore EDITOR espelha a composição gráfica da interface a ser gerada. Essa especificação era então levada ao usuário para apreciação, num processo paralelo de construção e avaliação da interação do sistema. Ao longo desse processo foram sugeridas algumas modificações que foram analisadas pelos projetistas e executadas em determinadas situações.

5.5. Integração dos módulos TAOS-Graph e TAME

Como citado anteriormente, os módulos TAME e TAOS-Graph foram desenvolvidos separadamente, baseando-se no princípio da independência do diálogo. Porém, ao final, estes módulos tiveram que ser integrados, formando assim a ferramenta iTAOS. Essa integração se deu através de uma API (Application Interface Programming) disponibilizada no módulo TAME, ou seja, no componente funcional.

Não houve a necessidade de que o módulo TAME estivesse finalizado para que o módulo TAOS-Graph começasse a ser construído. Uma primeira versão funcional do módulo TAME foi construída e disponibilizada então a API, que foi sendo utilizada para a construção do TAOS-Graph. O módulo TAOS-Graph evoluiu ao passo que o módulo TAME evoluía.

Os atributos da faceta *Abstração* da especificação conceitual da interação definem os métodos que foram invocados da API disponibilizada no módulo TAME. Alguns desses métodos foram apresentados nos extratos das árvores EDITOR apresentadas nesse capítulo.

5.6. Considerações finais

Neste capítulo foi apresentado o processo de obtenção da especificação da interação da tarefa *Utilizar a ferramenta iTAOS para analisar e modelar tarefas*. Essa especificação corresponde à segunda e à terceira etapa da metodologia MEDITE, que foi utilizada para concepção do módulo interativo (TAOS-Graph) da ferramenta iTAOS.

A especificação da interação foi realizada em dois momentos distintos: no primeiro momento foi realizada a especificação conceitual parcial da interação, que consistiu da construção das árvores EDITOR a partir da árvore TAOS da tarefa utilizando o conhecimento ergonômico disponível numa base de regras ergonômicas (GUERRERO, 2001) e num segundo momento essas árvores EDITOR foram complementadas com os atributos das

facetas *Apresentação*, *Abstração* e *Controle*, o que correspondeu a especificação completa da interação.

Logo após uma primeira versão da especificação conceitual da interação, ou seja, após a construção das árvores parciais EDITOR, começou-se a avaliação da especificação junto ao usuário, num processo interativo e incremental, ou seja, a especificação da interação era incrementada a cada interação com os usuários. Essa avaliação se estendeu até que tivéssemos a versão completa da especificação da interação validada pelo usuário.

No próximo capítulo, descreveremos o processo de construção do protótipo da interface da ferramenta iTAOS, ou seja, a implementação em JAVA. A construção do protótipo corresponde a última fase da metodologia MEDITE, que foi utilizada no projeto e implementação do módulo TAOS Graph de iTAOS.

Capítulo 6

6. Protótipo da interface

Neste capítulo, descrevemos o processo de construção do protótipo da interface da ferramenta iTAOS. Esta etapa possui como entrada a especificação conceitual completa da interação, ou seja, as árvores EDITOR e possui como saída um protótipo da interface da ferramenta iTAOS. Trata-se do processo de implementação orientada a objetos da interface utilizando a linguagem de programação JAVA.

Apresentamos inicialmente onde a construção do protótipo da interface se situa dentro da metodologia MEDITE. Na construção desse protótipo foi utilizado o framework JHotDraw (Kaiser, 2001) de apoio a construção de interfaces com o usuário em JAVA, o qual descrevemos na seção 6.2. Logo após, ilustramos como o framework foi estendido para a criação da interface da ferramenta iTAOS. Finalmente, apresentamos algumas telas da interface, comparando-as com as árvores geradas na especificação da interação.

6.1. Protótipo da interface na metodologia MEDITE

O protótipo da interface corresponde quarta etapa da metodologia MEDITE como mostra a figura abaixo:

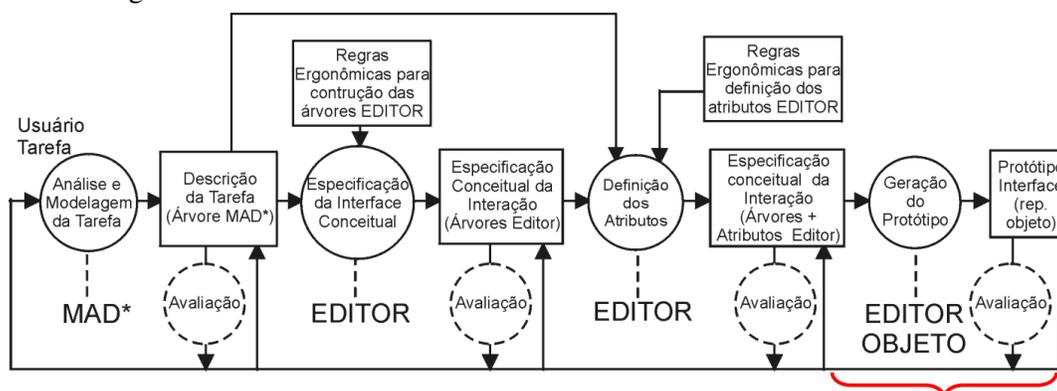


Figura 15. Protótipo da interface na metodologia MEDITE

6.2. Framework JHotDraw

O framework JHotDraw define um esqueleto básico para a construção de um editor com diferentes visões, figuras gráficas, ferramentas para manipulação dessas figuras, etc. O framework pode ser customizado usando herança e combinando componentes. O JHotDraw é estruturado em pacotes, como mostrado na figura 16 abaixo:

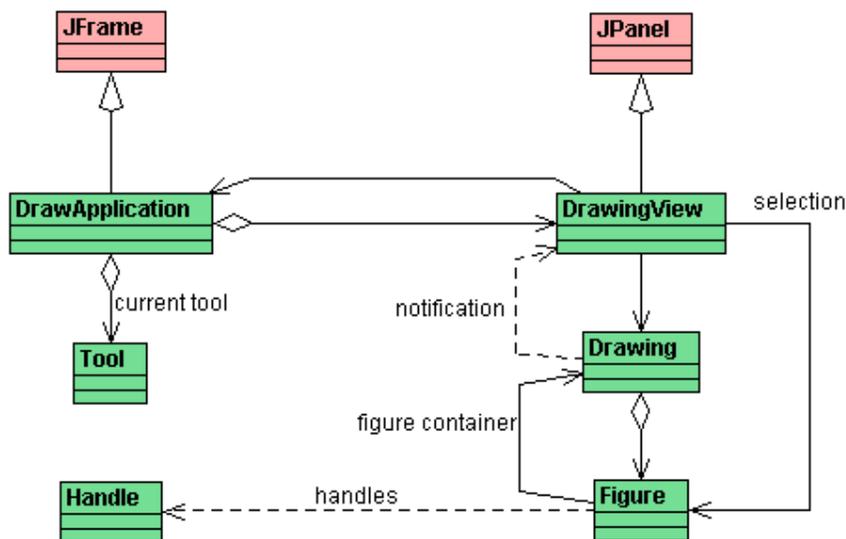


Figura 16. Estrutura em pacotes do framework JHotDraw

Qualquer aplicação que use o JHotDraw possui uma janela dedicada ao desenho. Essa janela editor possui uma ou mais janelas internas, cada uma associada a uma visão diferente (*DrawingView*). Cada visão possui uma área que aceita entradas oriundas do usuário (*Drawing*). Se algo é mudado no *Drawing*, esta mudança é propagada para a visão (*DrawingView*) que é responsável por atualizar as figuras. O *Drawing* consiste de figuras (*Figure*), que podem conter outras figuras. Cada figura possui um *Handle*, que define os pontos de acesso e determinam como interagir com a figura. Na visão, pode-se selecionar várias figuras e trabalhar com elas ao mesmo tempo. A janela editor possui uma ferramenta (*Tool*) ativada por vez, que opera no *Drawing* associada à visão (*DrawingView*) sob um determinado aspecto.

6.2.1. Processo de desenvolvimento típico Usando o JHotDraw

A lista apresentada a seguir envolve as tarefas envolvidas com o desenvolvimento de uma aplicação utilizando o JHotDraw:

- i.* Criar suas próprias figuras gráficas e símbolos da sua aplicação: Algumas figuras já estão definidas como *AbstractFigure*, *CompositeFigure* e *AttributeFigure*. Pode-se redefinir os métodos nas subclasses para conseguir o comportamento desejado de cada uma das figuras. Tipicamente, uma figura gráfica deve corresponder aos objetos usados na aplicação.
- ii.* Desenvolver suas próprias ferramentas de criação de figuras e manipulá-las de acordo com os requisitos da aplicação: JHotDraw oferece algumas classes que devem ser especializadas como *CreationTool*, *ConnectionTool*, *SelectionTool* e *TextTool*. Especializar tais classes e redefinir métodos como *mouseUp* e *nouseDown* permite que seja definida a interação com o usuário.
- iii.* Criar a atual interface com o usuário (GUI) e integrá-la a aplicação: JHotDraw também inclui o esqueleto de uma aplicação básica. Podem ser redefinidos os métodos da sua aplicação, tais como *createMenus*, *createFileMenu*. Uma completa interface com o usuário é criada quando se instancia a aplicação e chama *open()*.
- iv.* Compilar a aplicação usando a JVM (javac): É importante que sejam incorporados todos os pacotes necessários pelo JHotDraw.

Algumas dessas tarefas envolvem aplicar designs patterns⁵ (Gamma et al., 2001), como é o caso de Composite, State, Template method, Factory method, e Strategy.

Embora aprender a usar o JHotDraw ter requerido um esforço razoável, o uso deste reduziu o tempo de desenvolvimento e melhorou a qualidade do software.

⁵ Conhecidos como padrões de projeto em português.

6.3. Classes que compõem o sistema

Mostraremos algumas classes especializadas do framework JHotDraw para construção da interface da ferramenta iTAOS. As classes em azul pertencem ao framework JHotDraw e as classes em amarelo foram criadas:

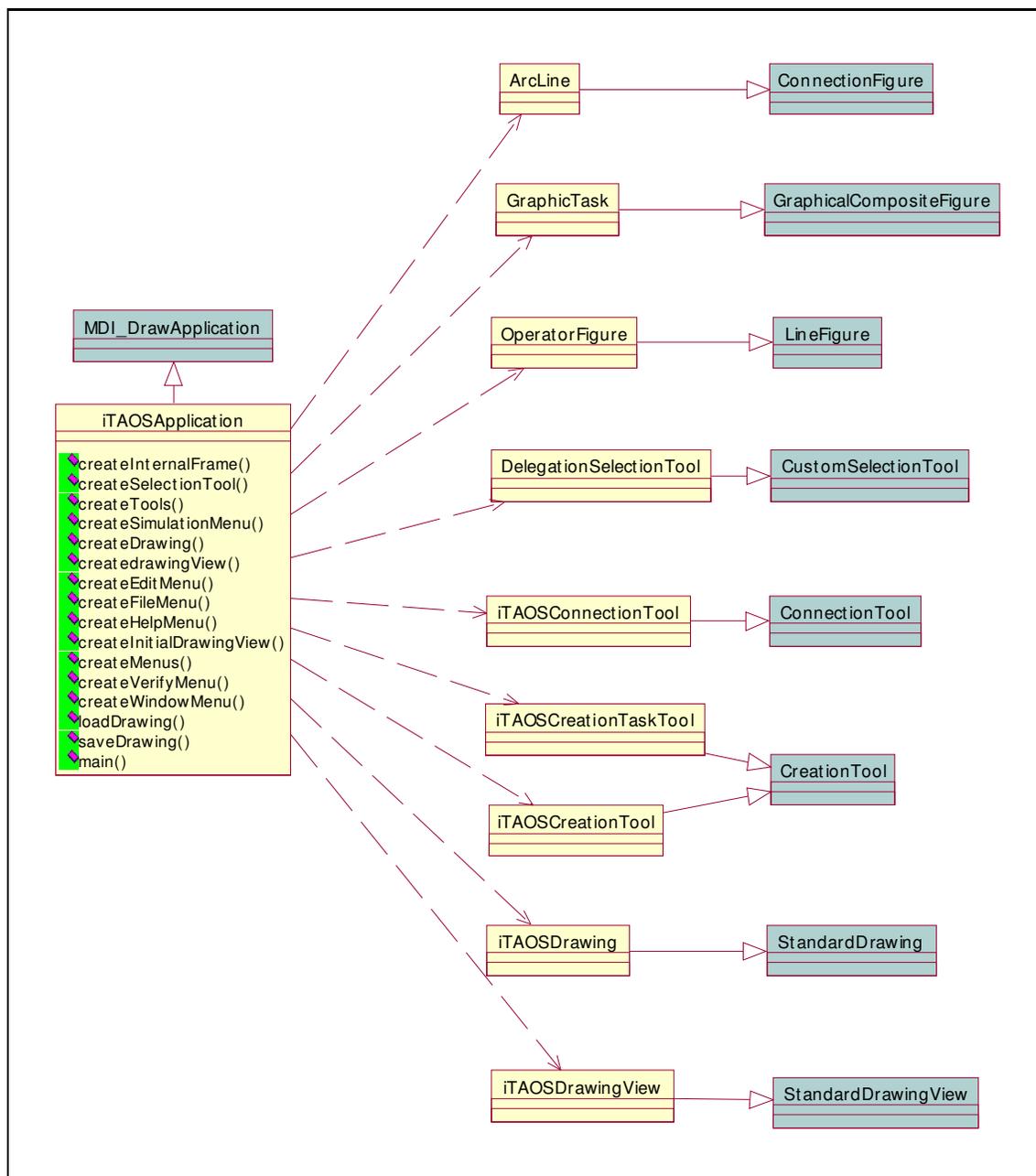


Figura 17. Diagrama das principais classes do módulo TAOS-Graph da ferramenta iTAOS

Ilustramos as dez principais classes das trinta do módulo interativo TAOS-Graph de iTAOS. Em todas essas classes há chamadas aos métodos do módulo TAME, especificadas através da API.

6.4. Apresentação de iTAOS

Nesta seção apresentamos algumas telas da interface do iTAOS. A figura 18 se refere à tela inicial da ferramenta. Segundo o modelo da interação, esta tela deve ter duas visões, sendo a primeira representada por uma barra de menu horizontal.



Figura 18. Tela inicial da ferramenta iTAOS

Posteriormente, apresentamos duas telas, a primeira, na figura 19, mostra o menu vertical com as opções: Criar um novo modelo, abrir um modelo existente, salvar o modelo, imprimir o modelo e fechar o modelo, como definido no modelo da interação. A segunda tela, na figura 20, apresenta um modelo aberto e pronto para ser manipulado diretamente pelo usuário:



Figura 19. Tela que mostra o menu File aberto

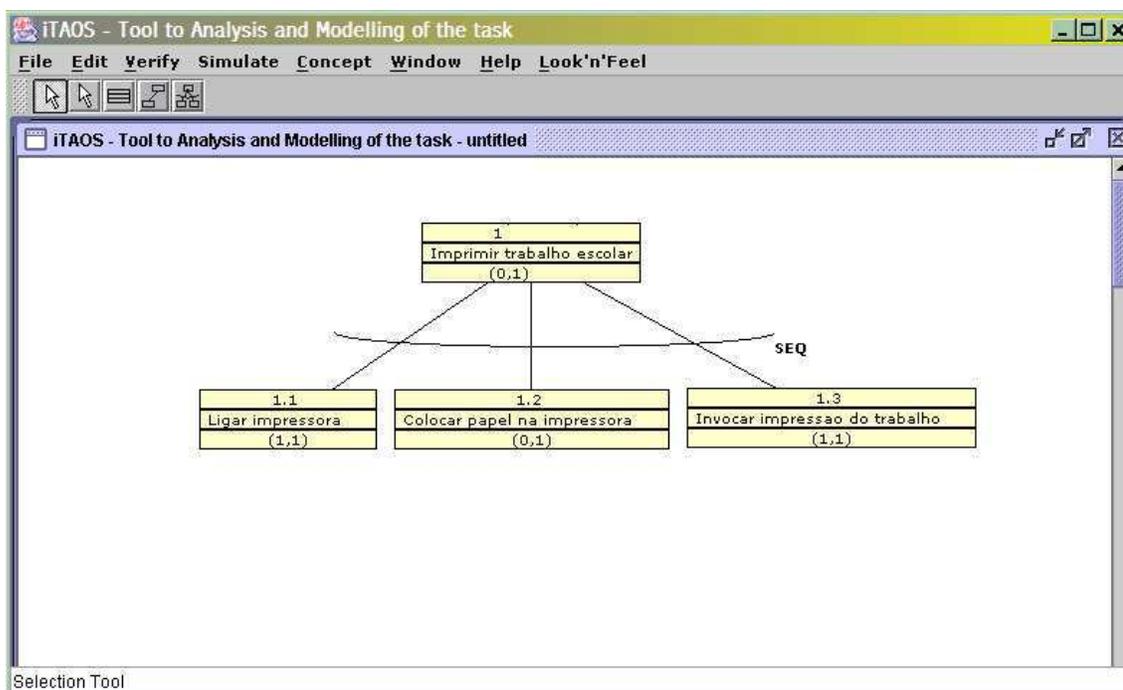


Figura 20. Tela que mostra um modelo aberto

Na figura 21, apresentaremos o pop-up menu que foi aberto ao se clicar com o botão direito sobre uma tarefa ou ação. Este pop-up menu foi definido na faceta Abstração do Objeto de interação denominado Árvore da especificação da interação.

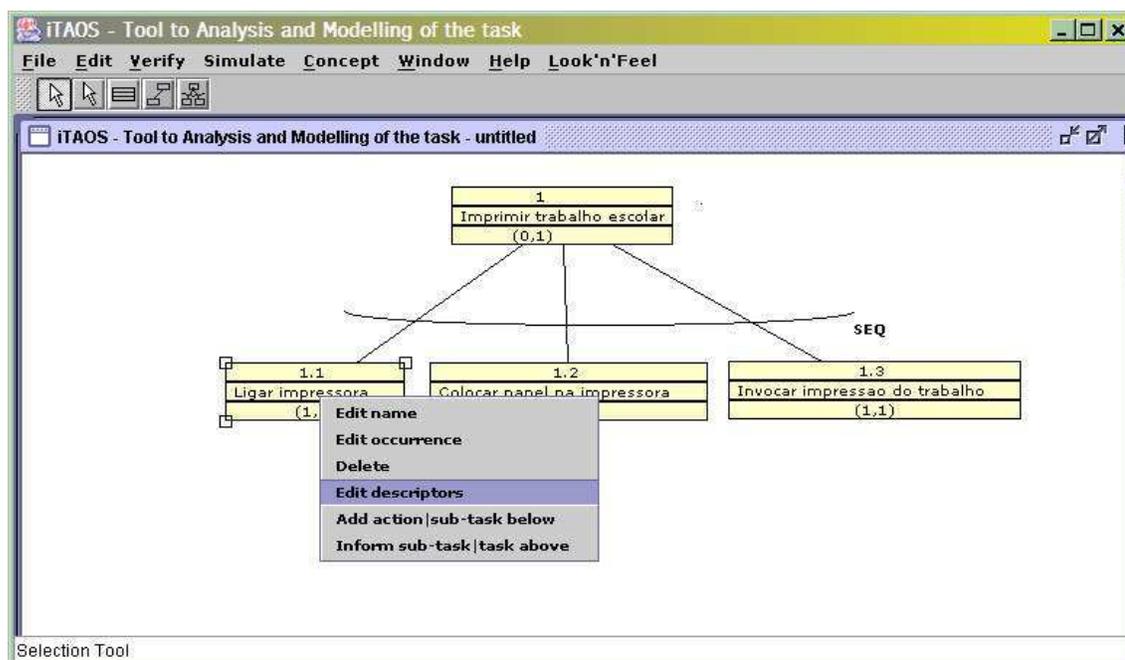


Figura 21. Tela que mostra o pop-up menu da ação

The screenshot shows the iTAOS application window with the 'Action Descriptor' dialog box open. The dialog has a menu bar (File, Edit, Verify) and a toolbar. It contains the following fields and controls:

- Name:** Ligar impressora
- Ocurrence:** (1,1)
- Description:** [Text field]
- Pre-situation:** Write or select a situation [Dropdown]
- Pos-situation:** Write or select a situation [Dropdown]
- List of agents:** [List box]
- Add, open or remove an agent:** Write or select an agent [Dropdown]
- List of tools:** [List box]
- Add, open or remove a tool:** Write or select a tool [Dropdown]

Buttons for OK, OPEN, and REMOVE are present next to the dropdown menus.

Figura 22. Tela que mostra o descritor de uma ação aberto

Na figura 22, apresentamos a tela com o descritor da ação *Ligar impressora* aberto. Esse descritor foi aberto ao clicar no item de menu *Edit descriptor* do pop-up menu. Este descritor, tipo formulário, também foi definido na especificação da interação.

Logo após ao descritor da ação *Ligar impressora*, apresentamos o descritor de um agente aberto. Este descritor foi aberto a partir do descritor de uma ação ou a partir do menu Concepts (Entidades) da barra de menu.

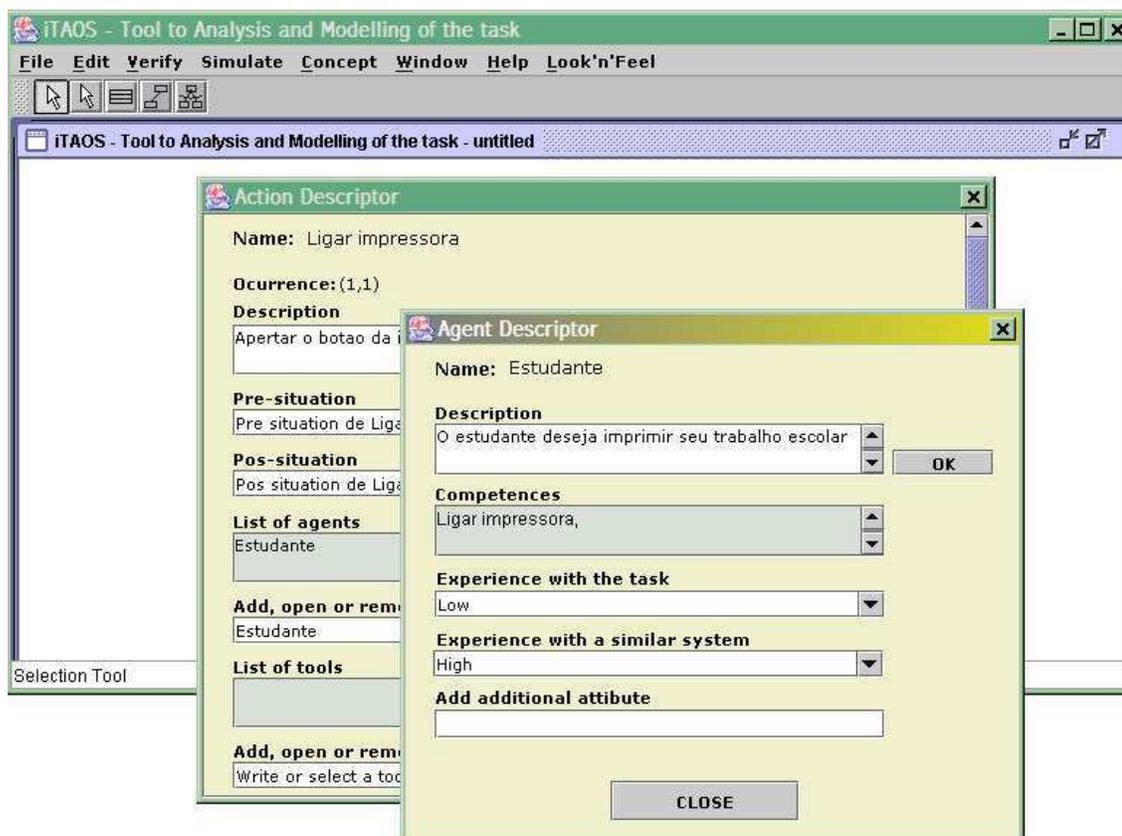


Figura 22. Tela que mostra o descritor de um agente aberto

Na figura 23, apresentamos uma tela onde o usuário pode visualizar todas as ações do modelo aberto e abrir uma delas. Janelas equivalentes a essa existem para todos os conceitos dinâmicos e estáticos de TAOS.

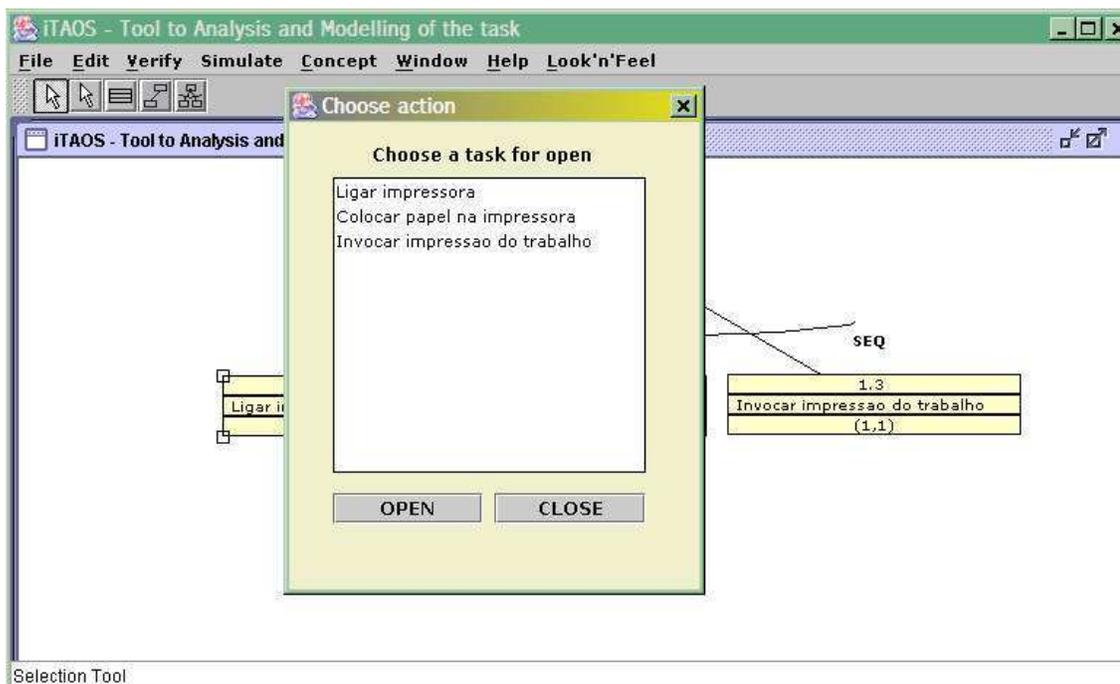


Figura 23. Janela que apresenta as ações do modelo aberto

6.5. Considerações finais

Neste capítulo foi apresentado o processo de construção do protótipo da interface da ferramenta iTAOS, que corresponde a quarta etapa da metodologia MEDITE foi utilizada para concepção do módulo interativo (TAOS-Graph) da ferramenta iTAOS.

A implementação do protótipo foi realizada utilizando o paradigma orientado a objetos, visto que os agentes PAC da especificação da interação são representados também segundo esse paradigma. Utilizamos a linguagem de programação JAVA e o framework JHotDraw na construção do protótipo da interface.

A validação do protótipo foi realizada segundo o método de inspeção de conformidade do protótipo da interface da ferramenta iTAOS com as partes 14, 16 e 17 do padrão ISO 9241. No próximo capítulo apresentamos a metodologia utilizada para realização da inspeção da usabilidade e os resultados obtidos.

Capítulo 7

7. Inspeção da usabilidade do protótipo da interface da ferramenta iTAOS

Apresentamos neste capítulo o processo de inspeção de conformidade do protótipo da interface da ferramenta iTAOS com as partes 14 (diálogos do tipo menu), 16 (diálogos do tipo manipulação direta) e 17 (diálogos do tipo formulário) do padrão internacional ISO 9241.

Inicialmente é feita uma breve descrição do padrão ISO 9241, apresentando posteriormente a metodologia utilizada na inspeção, os problemas identificados a partir da inspeção de conformidade com as partes 14, 16 e 17 do padrão e finalmente é discutido um parecer sobre o produto com base nas taxas de adoção das recomendações.

7.1. O padrão ISO 9241

A preocupação com a ergonomia de terminais cresceu no final da década de 70, sobretudo no tocante à possível fadiga visual e deterioração da visão dos usuários devido ao uso prolongado de terminais de vídeo, especialmente aqueles com baixa qualidade de apresentação visual. (Queiroz, 2001)

O padrão ISO 9241 (*International Organization for Standardization*) é uma norma internacional que se destina aos profissionais encarregados de garantir um trabalho de escritório seguro e efetivo com os computadores. As recomendações que constam na ISO 9241 foram definidas por evidência empírica e a partir da revisão da literatura existente na época, sendo então generalizadas e formuladas em termos de requisitos para o uso de projetistas e avaliadores de interfaces (Cybis, 2000).

Inicialmente, foi decidida a elaboração de um padrão que cobrisse uma gama de tópicos em ergonomia que o comitê de ergonomia da ISO acreditava precisar de atenção. Passaram quase sete anos antes da publicação das primeiras partes do padrão ISO 9241 (Stewart, 2000), um padrão de dezessete partes, cada uma tratando de diferentes aspectos do trabalho em escritórios informatizados, que levou mais de quinze anos para ser concluído. Na tabela abaixo listamos as dezessete partes que compõem o padrão ISO 9241.

Tabela 1. Partes integrantes do padrão ISO 9241

ISO 9241	
Ergonomic requirements for Office work with visual display terminals (VTDs)	
Parte	Denominação
1:1997	General Introduction
2:1992	Guidance on task requirements
3:1992	Visual display requirements
4:1998	Keyboard requirements
5:1998	Workstation layout and postural requirements
6:1999	Environmental requirements
7:1998	Display requirements with reflections
8:1997	Requirements for displayed colours
9:2000	Requirements for non-keyboard input devices
10:1996	Dialogue principles
11:1998	Guidance on usability specification and measures
12:1998	Presentation of information
13:1998	User guidance
14:1997	Menu dialogues
15:1997	Command dialogues
16:1999	Direct manipulation dialogues
17:1998	Form filling dialogues

As nove primeiras partes do padrão ISO 9241 tratam de aspectos relacionados com o hardware envolvido no processo interativo, enquanto que as outras oito partes são relativas ao software. As partes 14 a 17 referem-se a estilos de diálogo; por menu, por linguagem de comandos, por manipulação direta e por preenchimento de campos. As normas fornecem uma

estrutura de recomendações referentes à pertinência destes estilos de diálogo, sobre como realizá-los em seus diferentes aspectos e como avaliá-los.

Descrevemos abaixo as partes 14 (diálogos do tipo menu), 16 (diálogos do tipo manipulação direta) e 17 (diálogos do tipo formulários) do padrão ISO 9241, que foram utilizadas na realização da inspeção de conformidade da interface da ferramenta iTAOS:

- i.* Diálogos por menu (ISO, 1997): Apresentação de recomendações para o projeto ergonômico de diálogos via menus, englobando estrutura de menus, navegação, seleção e execução de opções e apresentação de menus por várias técnicas, e.g. janelas, painéis, botões, campos.
- ii.* Diálogos por manipulação direta (ISO, 1999): Apresentação de recomendações para o projeto ergonômico de diálogos via manipulação direta, incluindo a manipulação de objetos e o projeto de metáforas, objetos e atributos.
- iii.* Diálogos por preenchimento de formulários (ISO, 1998): Apresentação de recomendações para o projeto ergonômico de diálogos via preenchimento de formulários, cobrindo considerações estruturais, de entrada, saída e navegação.

7.2. Metodologia utilizada na inspeção da usabilidade

A utilização do padrão internacional ISO 9241 é destinada tanto para projetistas que desejam analisar a aplicabilidade de recomendações ergonômicas num produto quanto para avaliadores que desejam verificar a conformidade de um produto específico ao padrão ISO mediante a adoção ou não das recomendações existentes.

Ao final de cada parte da ISO 9241 há checklists (denominado Anexo A), que apresentam procedimentos para análise da aplicabilidade e/ou da adoção das recomendações. A conformidade ao padrão ISO 9241 é definida a partir dos resultados de duas análises, a de aplicabilidade do(s) quesito(s) do checklist e a da adoção do quesito pelo sistema avaliado. O resultado final da avaliação deve conter as tarefas (quesitos) avaliadas, as recomendações aplicáveis e as recomendações adotadas.

Nosso objetivo ao usar as recomendações da ISO foi de avaliar a interface da ferramenta iTAOS já existente, limitando-se as partes 14, 16 e 17 do padrão internacional ISO 9241. O procedimento utilizado foi o sugerido na norma ISO 9241:

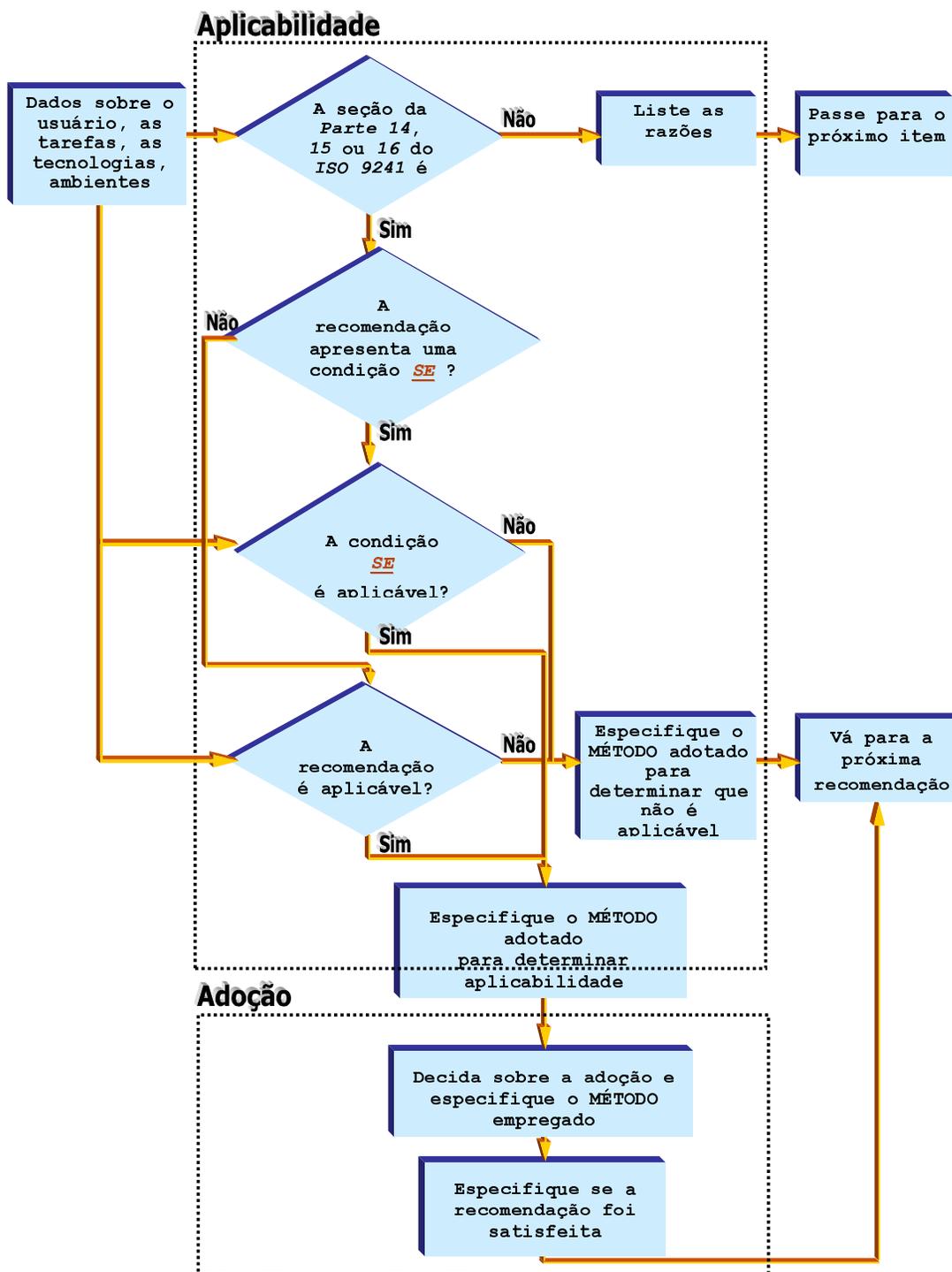


Figura 24. Fluxograma empregado na inspeção da usabilidade

7.3. Problemas identificados no produto a partir da inspeção de usabilidade

O *Anexo A* de cada uma das partes do padrão ISO 9241 apresenta ao final um checklist sintetizando as recomendações contidas em cada parte do padrão, servindo de auxílio tanto para projetistas, com respeito à aplicabilidade das recomendações no contexto do projeto, quanto para avaliadores, num processo de verificação das recomendações num produto final.

Nas listas de inspeção há campos em branco, que devem ser preenchidos conforme a evolução do processo de inspeção e há também campos não preenchíveis pelo projetista/avaliador, destacados dos demais pelo preenchimento em cor cinza, o que indica que não deve ser assinalado, por não se aplicarem ao processo de inspeção ou por não serem pertinentes ao contexto individual da declaração à qual estão associados.

Os checklists preenchidos referentes às partes 14, 16 e 17 do padrão ISO 9241 encontram-se nos apêndices 3, 4 e 5 respectivamente. Abaixo, apresentaremos os problemas identificados a partir da inspeção da conformidade da ferramenta iTAOS com o padrão ISO 9241.

Problemas identificados a partir da inspeção da conformidade do produto com a ISO 9241-14 - Diálogos por menu

Não foram encontrados problemas de conformidade da ferramenta iTAOS com as recomendações da parte 14 do padrão ISO 9241. Todas as recomendações que são aplicáveis a ferramenta iTAOS foram adotadas.

Problemas identificados a partir da inspeção da conformidade do produto com a ISO 9241-16- Diálogos por manipulação direta

Na declaração 6.1.4 (Manipulação direta de saída) recomenda-se que se apropriado para a tarefa, o resultado da manipulação direta deveria ser apresentado de tal maneira que o usuário pudesse modificá-lo. Por exemplo, o usuário cria um documento de texto, este documento possui um nome default “untitled document” que já aparece selecionado para que o usuário o modifique. No iTAOS, ao se inserir uma ação na tela, ela aparece com o nome default “name of the task”, porém ela não aparece selecionada para que o usuário a modifique.

A segunda falha explicitada através da inspeção de conformidade à ISO 9241-16 é relativa à recomendação 8.2.1 (Rearranjo do conteúdo visualizado de uma janela de acordo com a seleção do usuário), recomenda-se que o alcance dos objetos selecionados por manipulação direta possa ir além do tamanho da janela, fazendo com que a janela deslize até encontrar o objeto desejado. No iTAOS essa recomendação não foi atendida.

Problemas identificados a partir da inspeção da conformidade do produto com a ISO 9241-17 - Diálogos por preenchimento de formulários

Na declaração 6.1.4 (Manipulação direta de saída) recomenda-se que se os campos são alfanuméricos e alinhados verticalmente em colunas e se o tamanho do rótulo difere de forma significativa, os rótulos devem ser justificados a direita e os campos devem ser justificados a esquerda. No iTAOS, os rótulos e os campos estão alinhados a esquerda, não atendendo à recomendação.

A segunda falha foi constatada na declaração 5.3.3 (Campos modificáveis versus campos não modificáveis), que recomenda que os usuários sejam capazes de identificar facilmente os campos que podem ser modificados dos que não podem (campos só de leitura). No iTAOS, mesmo os campos que são somente de seleção, estão passíveis de edição, não condizendo com o que recomenda a declaração 5.3.3.

A declaração 6.4.2 (Identificando e localizando erros) recomenda que se na validação do campo forem detectados erros e se é apropriado para a tarefa, estes campos devem ser indicados e o cursor deve ser colocado no lugar onde ocorreu o primeiro erro, para que seja fácil para o usuário modificá-lo. No iTAOS, o cursor não volta para o campo onde ocorreu o erro.

A declaração 6.5.2 (Validação de múltiplos campos) recomenda que as dependências entre campos no formulário, ou entre campos e outros incidentes do mesmo formulário, sejam validadas, por exemplo: se um campo A não foi preenchido, então o campo B não pode ser editado. No iTAOS, este tipo de verificação não está sendo checado.

A última falha detectada concerne à recomendação 8.6.4 (Retornando ao formulário inicial), aplicável a um conjunto de formulários hierárquicos. Deve estar disponível para o

usuário um atalho para que ele retorne ao formulário inicial. Este tipo de atalho não está disponível nos formulários da ferramenta iTAOS.

7.4. Taxas de adoção com base no processo de inspeção

A *ISO* recomenda, após a realização da inspeção de conformidade de um produto a um de seus padrões internacionais ou a algumas de suas partes, que os resultados da inspeção sejam sumariados a partir da computação de um indicador denominado *taxa de adoção*⁶, definido como a razão percentual do número de recomendações julgadas satisfatoriamente adotadas pelo produto (i.e., o número de células assinaladas na coluna **P** da lista de inspeção) pelo número de recomendações julgadas aplicáveis ao contexto do projeto (i.e., o número de células assinaladas na coluna **S** da lista de inspeção).

Tabela 2 – Taxas de adoção do iTAOS às partes 14, 16 e 17 do ISO 9241.

PARTE DO ISO 9241	#P	#S	TA (%)
14	62	62	100,0
16	39	41	95,1
17	44	49	89,7

Além das taxas de adoção do *iTAOS* às partes 14, 16 e 17 do padrão internacional *ISO 9241*, apresentadas na coluna **TA** (taxa de adoção) da Tabela 2, são também reportados os números de células assinaladas nas colunas **P** e **S** das listas de inspeção, apresentados, respectivamente, nas colunas **#P** e **#S**, conforme recomendado na sub-seção **A.7.2 (Summary data)** de cada parte considerada.

De acordo com a Tabela 2, verifica-se que o modo de interação oferecido pelo *iTAOS* mais conforme (**TA = 100%**) às recomendações da *ISO* é o que envolve diálogos via *menus*, seguido do modo de interação via *manipulação direta* (**TA = 95,1%**). Em contrapartida, o modo de interação fundamentado em diálogos via *preenchimento de formulários* é o que adota o menor número de recomendações da *ISO*

⁶ Adoção das recomendações aplicáveis ao contexto do projeto cuja implementação originou o produto inspecionado.

aplicáveis aos contextos de uso do universo amostral considerado, apresentando, portanto, a taxa de adoção mais baixa (TA = 85,7%).

7.5. Considerações finais

Apresentamos neste capítulo o processo de inspeção de conformidade do protótipo da interface da ferramenta iTAOS com as partes 14 (diálogos do tipo menu), 16 (diálogos do tipo manipulação direta) e 17 (diálogos do tipo formulários) do padrão internacional ISO 9241. Segundo as recomendações da ISO, algumas foram detectadas falhas nos diálogos com o usuário.

Como apresentado anteriormente, a metodologia MEDITE utiliza o conhecimento ergonômico na concepção da interação. Esse conhecimento é estruturado na forma de regras ergonômicas e aplicado em dois momentos: no primeiro, as regras são utilizadas na estruturação das telas ou janelas do sistema (esboço inicial), enquanto que no segundo momento, as regras auxiliam na definição dos atributos da apresentação, abstração e controle do sistema.

Por se tratar de uma primeira versão da base de regras ergonômicas, existem poucas regras para definição dos atributos dos diálogos via manipulação direta e preenchimento de formulários. Por esse motivo, algumas falhas foram observadas na inspeção de conformidade com as partes 16 e 17 da norma ISO 9241.

As falhas levantadas foram todas corrigidas após a realização da inspeção. Porém, não foi realizada uma nova inspeção de conformidade com a norma ISO 9241.

Capítulo 8

8. Conclusão

Apresentamos nesse trabalho o processo de construção e implementação do módulo TAOS Graph da ferramenta iTAOS, uma ferramenta de suporte à análise e modelagem da tarefa. O módulo TAOS Graph refere-se ao componente interativo da ferramenta, ou seja, um editor de entidades estáticas e dinâmicas estruturadas de acordo com os princípios preconizados pelo módulo TAME (componente funcional).

O módulo TAOS-Graph foi projetado utilizando a metodologia MEDITE, uma metodologia baseada na tarefa e orientada a modelos que utiliza o conhecimento ergonômico na passagem do modelo da tarefa para a especificação conceitual da interação de forma a produzir interfaces que reflitam os objetivos, as características e as necessidades do usuário.

Inicialmente a tarefa foi descrita utilizando o modelo TAOS, logo após foi feita a passagem da descrição da tarefa para a especificação conceitual da interação com o auxílio da base de regras ergonômicas disponível em MEDITE e finalmente foi construído o protótipo da interface utilizando a linguagem JAVA.

Por se tratar de um processo interativo e incremental, a avaliação de cada etapa da metodologia adotada foi sendo realizada em paralelo com a descrição dos modelos. Após a construção do protótipo da interface, foi realizada uma inspeção de conformidade do protótipo da interface gerada com as partes 14, 16 e 17 do padrão internacional ISO 9241, que correspondem respectivamente a diálogos via menus, diálogos via manipulação direta e diálogos via preenchimento de formulários.

Num primeiro momento, realizaríamos um teste de usabilidade da interface da ferramenta iTAOS, porém tornou-se inviável devido ao tempo que dispúnhamos para realização de todo o trabalho, ou seja, o projeto, implementação e avaliação do módulo

TAOS-Graph do iTAOS. Por esse motivo, optamos por uma avaliação baseada em padrões (checklists), que é um método de avaliação mais rápido e que possui as seguintes vantagens segundo Cybis (2000):

- i.* Facilidade na identificação de problemas de usabilidade, devido à especificidade das questões do checklist.
- ii.* Aumento da eficácia da avaliação, devido à redução da subjetividade normalmente associada a processos de avaliação.
- iii.* Devido o conhecimento ergonômico ser inserido dentro do próprio checklist, a avaliação pode ser realizada por projetistas, não exigindo especialistas em avaliação de interfaces homem-máquina.
- iv.* Redução dos custos de avaliação, pois é um método de avaliação rápido.

Alem da inspeção de conformidade da interface de iTAOS com as partes 14, 16 e 17 da norma ISO 9241, foi realizada uma avaliação heurística do produto iTAOS. Essa avaliação foi executada por um especialista em design com o objetivo de diagnosticar problemas ou barreiras que os usuários encontrariam durante a interação com a interface. Essa avaliação voltou-se para algumas características visuais da interface, como fontes, diagramação, cores e imagens. No decorrer da avaliação, um conjunto de modificações foi detectado, entre as quais: a fonte que estava sendo utilizada nos menus e em outros estilos de diálogo, a diagramação que estava sendo utilizada nas telas de formulários, as imagens utilizadas nos botões (ícones) e as cores das tarefas e ações do modelo, dando um melhor contraste com o background da tela. É importante ressaltar que esta avaliação heurística foi realizada antes da inspeção por padrões

8.1. Discussão dos resultados

Discutiremos as hipóteses consideradas para o desenvolvimento deste trabalho de acordo com os resultados obtidos:

- i.* **É possível projetar e implementar o módulo TAOS-Graph separado do processo de desenvolvimento do módulo TAME.**

Os módulos TAME e TAOS-Graph foram projetados seguindo processos de desenvolvimento distintos. Isso foi possível por termos assumido *a priori* o princípio da independência do diálogo e pela definição de uma interface de comunicação (API: Application Programming Interface) robusta que permitisse uma integração adequada dos módulos *a posteriori*. Essa API foi definida e disponibilizada no módulo TAME e seu acesso definido através dos atributos das facetas *Abstração* da especificação EDITOR da interação, conforme Capítulo 5.

- ii. O processo de desenvolvimento a ser utilizado no projeto do módulo TAOS-Graph deve partir da análise da tarefa “Utilizar a ferramenta iTAOS para analisar e modelar tarefas” de forma a capturar efetivamente o conhecimento sobre o usuário e sobre seu processo de realização da tarefa.**

O artefato produzido durante a fase de análise e modelagem da tarefa descreve efetivamente o perfil do usuário, obtido através do questionário DePerUse e representado através dos atributos da classe Agente (Agent) e o processo de realização da tarefa obtido através de entrevistas semidirigidas e da técnica trace analysis e representado através da sua decomposição em sub-tarefas e dos descritores conforme descrito no capítulo 4.

- iii. A metodologia MEDITE possibilita uma construção guiada por modelos do módulo TAOS-Graph de forma a obedecer aos princípios da ergonomia e aos objetivos do usuário da ferramenta iTAOS.**

Todas as fases de construção do módulo TAOS-Graph tiveram um modelo como paradigma. Na fase inicial, análise da tarefa, utilizou-se o modelo TAOS de representação da tarefa no lugar de MAD* como preconizado pela metodologia. Isto foi feito sem problemas visto que TAOS já havia sido validado em relação a MAD e estendido para se equiparar a MAD* em termos de atributos orientados a especificação de interfaces e simulação. A passagem da descrição da tarefa para a especificação da interação foi realizada tendo como guia o modelo EDITOR, seguindo regras ergonômicas e preservando os objetivos do usuário declarados na descrição da tarefa. Nessa fase, a maior dificuldade encontrada foi com respeito à limitação da base de

regras de MEDITE que carece de mais regras relativas a diálogos via manipulação direta e preenchimento de formulários.

Os módulos TAME e TAOS-Graph foram desenvolvidos utilizando metodologias distintas e atenderam ao princípio da independência do diálogo. Ambas as metodologias (Processo Unificado e MEDITE) partiram de um mesmo ponto (análise da tarefa) e seguiram independentemente até chegar a um ponto comum, a integração dos dois módulos formando a ferramenta iTAOS.

8.2. Contribuições

A principal contribuição desse trabalho é o módulo TAOS-Graph (módulo interativo), que juntamente com o módulo TAME formam a ferramenta para análise da tarefa iTAOS. A ferramenta iTAOS possui características importantes, entre elas:

- i.* Adota o conceito de decomposição hierárquica de tarefas, permitindo que os projetistas considerem vários níveis de abstração e mantenham um relacionamento entre eles.
- ii.* Consideram tarefas concorrentes, através dos operadores SIM e PAR.
- iii.* Habilidade de modelar tarefas cooperativas envolvendo múltiplos usuários, através da lista de agentes (conceito Agent) das ações (Action).
- iv.* Habilidade de modelar tarefas envolvendo as ferramentas (Tools) e/ou objetos necessários para sua realização.
- v.* A descrição TAOS da tarefa gerada na ferramenta iTAOS (arquivo XML) pode ser utilizada para geração automática ou semi-automática da especificação da interação.
- vi.* Verificação da consistência da descrição gerada, analisando os objetos da descrição e a relação das pré e pós condições com os operadores envolvidos na descrição.

- vii. Verificação da completude da descrição, ou seja, é verificada a falta do preenchimento de algum atributo (Task, Action, Situation, Agent e Tool). (Melhorar)
- viii. Ambiente multiplataforma, permitindo que projetistas que utilizem diferentes sistemas operacionais possam usufruir da ferramenta iTAOS.
- ix. Suporte a simulação da descrição da tarefa, através dos atributos prioridade, tempo de execução e interruptibilidade.

Uma outra contribuição importante diz respeito à adição de uma nova regra a base de regras ergonômicas da metodologia MEDITE. Duas regras extraídas da experiência do autor com outras ferramentas foram utilizadas para a construção da especificação conceitual da interação. A primeira regra diz respeito ao estilo de diálogo Formulários e a segunda a manipulação direta de um objeto complexo que pode ser dinamicamente criado e/ou alterado. A primeira se encaixa nas regras para a construção da árvore EDITOR e a segunda na regras para definição dos atributos das árvores EDITOR.

Regra 1: SE as sub-tarefas forem ligadas pelo construtor OR (TAOS) e as sub-tarefas possuírem no atributo Nome a palavra Editar, Escolher ou equivalente, ENTÃO definir para esse conjunto de tarefas uma Visão (EDITOR) cuja Apresentação deve ser do tipo Formulário e cada sub-tarefa deve ser um Objeto_de_Interação (EDITOR) do tipo item-de-formulário dessa Visão.

Regra 2. Se o objeto de interação (EDITOR) for relacionado com um objeto da tarefa (TAOS) que pode ser dinamicamente criado ou alterado durante a interação, ENTÃO os atributos da sua faceta Apresentação devem descrever as ações de manipulação desse objeto e os atributos de sua faceta Abstração devem descrever os efeitos visuais e funcionais envolvendo o objeto.

Além das contribuições citadas anteriormente, a construção da ferramenta iTAOS procura contribuir com a promoção e consolidação da idéia de que metodologias de concepção e desenvolvimento de interfaces que levam em conta, desde o início do processo, a análise da tarefa e o conhecimento ergonômico produzem interfaces do usuário com um alto grau de usabilidade.

8.3. Propostas de continuidade

Daremos algumas sugestões de trabalhos futuros:

- i.* Validação da interface da ferramenta iTAOS pelos usuários. Essa validação vai ocorrer com o uso da ferramenta no âmbito da Universidade Federal de Campina Grande e até fora dela por projetistas de interfaces na descrição de suas tarefas, ou através de um teste de usabilidade.
- ii.* Acréscimo de regras à base de regras ergonômicas da metodologia MEDITE. O conhecimento ergonômico utilizado por MEDITE é limitado, portanto seria de grande contribuição que novas regras referentes a outros tipos de diálogos fossem agregados à metodologia.
- iii.* Desenvolvimento de ferramentas específicas para as outras etapas de uma metodologia para o projeto de interfaces, de forma que tenhamos uma metodologia totalmente suportada por ferramentas computacionais e que estas ferramentas se comuniquem.
- iv.* Possibilitar que usuários da ferramenta iTAOS possam simular seus modelos. iTAOS foi projetada e implementada de forma que essa nova funcionalidade pudesse ser agregada a ferramenta posteriormente.

Referências Bibliográficas

- BODART, F. 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.
- BODART, F. 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.
- COUTAZ, J. *The Construction of User Interfaces and the Object Paradigm*, The European Conference on Object Oriented Programming, ECOOP'87, Paris, França, 1987.
- COUTAZ, J. *Interfaces Homme-Ordinateur – Conception et Réalisation*, Bordas, Paris, 1990.
- CYBIS, W. A., *Ergonomia e Usabilidade de Software – Abordagem Ergonômica para IHC*. Florianópolis, 1996.
- CYBIS, W. A., *Abordagem ergonômica para IHC*, apostila de curso, Laboratório de Utilizabilidade INE/UFSC, Florianópolis, Brasil, 2000.
- DODANI at. al., *Separation of Powers*, Byte, mars 1989.
- DRAPER, S. W. e NORMAN, D. A., *Software engineering for user interfaces*, IEEE Trans. Softw. Eng. SE-11, 1985, 252-258.
- EHRICH, R. W. e HARTSON, H. R., *DMS - An environment for dialogue management*, Proc. of COMPCON81 (Washington, D.C.), IEEE, New York, septembre 1981, p. 121.
- 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.
- GAMMA, E. at al., *Design Patterns – Elements of Reusable Object-Oriented Software*, Addison-Wesley, 2001.

- GAMBOA, F.; SCAPIN, D. L. *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, Springer Verlag.
- 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.
- GRAESSER, A. C. How to catch a fish: The memory and representation of common procedures. *Discourse Processes*, 1978, 1, 79-89.
- GUERRERO, C.V.S. MEDITE, *Uma metodologia orientada a modelos para concepção de interfaces ergonômicas*. Dissertação de Mestrado - COPIN, Universidade Federal da Paraíba, Campina Grande, PB, 2001.
- GUERRERO, C. V. S.; LULA, B. Jr. *MEDITE: uma Abordagem Baseada em Modelos para a Concepção de Interfaces Ergonômicas*, IX Encuentro Chileno de Computación 2001- Jornadas Chilenas de Computación 2001, Atas (CD-ROM) Seção: Interfaz humano-computador y computación gráfica / Trabalho nº 4 (10 páginas), Punta Arenas - CHILE, novembro de 2001.
- 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 páginas), UFPB/DSC/CCT Campina Grande, PB, outubro de 2001.
- GUERRERO, C. V. S.; LULA, B. Jr. *Model-Guided and Task-Based Approach to UI Design Centered in a Unified Interaction and Architectural Model*, CADUI'2002 - 4th International Conference on Computer-Aided Design of User Interfaces, p. 107 - 119, Valenciennes, France, May 2002.
- HAAN, G. *ETAG - A Formal Model of Competence Knowledge for User Interface Design*, Ph.D. Thesis, Vrije Universiteit, Amsterdam, October, 2000.
- HAMMOUCHE, H. *De la modélisation des tâches utilisateurs au prototype de l'interface homme-machine*, Thèse de Docteur, Université Paris VI, Décembre, 1995.
- HEEMANN, V., *Curso de Ergonomia em Sistemas de Informação*, Universidade Federal da Paraíba, Campina Grande, Julho de 1997.

- 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.
- HUDSON, S., JOHN, B., KNUDSEN, K., BYRNE, M., *A Tool for Creating Performance Models from Users Interface Demonstrations*, Proceedings UIST'99, pp. 93-102.
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, *Projeto de norma internacional, Menu dialogues - ISO 9241 parte 14, final draft*, Genebra, Suíça, 1997.
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, *Projeto de norma internacional, Direct manipulation dialogues - ISO 9241 parte 16, final draft*, Genebra, Suíça, 1999.
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, *Projeto de norma internacional, Form filling dialogues - ISO 9241 parte 17, final draft*, Genebra, Suíça, 1998.
- JOHNSON, P. at al. *Task-Related Knowledge Structures: Analysis, Modelling and Application*, Queen Mary College, University of London 1988.
- JOHNSON, P. at al., J. ADEPT – *Advanced Design Environment for Prototyping with Task Models*, INTERCHI'93 Conference Proceedings, Amsterdam: ACM, 1993.
- 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.
- KAISER, W. *Become a programming Picasso with JHotDraw*, 2001.
- KESSEL, T.; MEDEIROS, H. e ROUSSELOT, F., *Some Useful Enhancements of Modeling Languages to become Modeling Languages*, International Workshop “Modeling Languages for Knowledge-Based Systems”, 01/95, Amsterdam, Pay Bas, 1995.
- LIMBOURG, Q.; PRIBEANU, C.; VANDERDONCKT, J., *Towards Uniformed Task Models in a Model-Based Approach*, GIST Technical Report, Glasgow, Scotland, June 2001.
- LULA, B. Jr. *Elaboration d'un Environnement de Génération Interactive d'Interfaces à Manipulation Directer pour le Language OBJLOG*, Thèse de Docteur, Universidade de

- Droit d'Economie et des Sciences d'Aix-Marseille III, Faculté des Sciences et Techniques de Saint – Jérôme, França, 1992.
- MARKOPOULOS P. e GIKAS S., *Formal Specification of a Task Model and Implications for Interface Design*, Cognitive Systems vol 4, pp. 4-3 Queen Mary and Westfield College, University of London, February 1997.
- MARKOPOULOS, P.; PYCOCK, J.; WILSON, S., *ADEPT - A task based design environment*, Queen Mary and Westfield College, UK, 1992.
- MEDEIROS, F. P. A., CORDEIRO, P. B. e LULA, B. J., *Projeto iTAOS – Modelagem da tarefa e Fase de planejamento*. Relatório Técnico RT- DSC-001/2002, (112 p.), UFPB/DSC/CCT Campina Grande, PB, Brazil, abril, 2002, disponível em www.dsc.ufcg.edu.br/~itaos.
- MEDEIROS, F. P. A.; CORDEIRO, P. B. e LULA, B. J., *Projeto iTAOS – Fase de Implementação (Primeira Iteração) e Especificação Conceitual da Interação*. Relatório Técnico RT- DSC-003/2002, (47 p.), UFPB/DSC/CCT Campina Grande, PB, Brazil, julho, 2002, disponível em www.dsc.ufcg.edu.br/~itaos.
- MEDEIROS, F. P. A., LULA, B. e CORDEIRO, P. B. *A Graphical Tool to Support Task Description using TAOS Formalism for UI Design* In: 7th ERCIM Workshop, 2002, Paris. 7th ERCIM Workshop. ERCIM (European Research Consortium for Informatics and Mathematics), 2002. p.45 – 51.
- MEDEIROS, F. P. A.; LULA, B., CORDEIRO; P. B. *iTAOS: a Graphical Tool to Support User's Task Description in UI Design Context* In: V Symposium on Human Factors in Computer Systems, 2002, Fortaleza. V Symposium on Human Factors in Computer Systems. Fortaleza: BNDE, 2002. v.01. p.376 – 379.
- MEDEIROS H., *L'Utilisation d'un Language Terminologique dans la Modélisation de Concepts Dynamiques*; Journées 1995, Projet ACNOS, LRPS-BETA-LAG-INRIA, 04-05, Sept 1995, Strasbourg, França, 1995.
- 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 oh he Chilean Computer Science Society, november 2000, Santiago, Chile, 2000.

- MEDEIROS, H. e ROUSSELOT, F., *Acquisition et Modélisation de Concepts Dynamiques: Le Système TAME*, Rapport ERIC R0102-96, Strasbourg, França, 1995.
- 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.
- MINSKY, M., *A framework for representing knowledge*. In Winston, P. (Ed.), *The Psychology of Computer Vision*, chap. 6, pp. 211--277. McGraw-Hill, New York, 1975
- PARIS, C.; TARBY,J.; VANDER, L.K., *A Flexive Enviroment for Building Task Models*. Proceedings of the IHM-HCI 2001, Lille, France.
- PFAFF, G. R. *User Interface Management Systems*, Eurographics Seminars, Springer-Verlag, 224 pages, 1985.
- 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.
- SMITH R. G.; BARTH, P. S. e 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.
- SCAPIN, D. L e PIERRET-GOLKBREICH, C., *Towards a Method for Task Description: MAD*, Unité de Recherche, INRIA, Rocquencourt, France, 1989.
- SEBILLOTE, S., *Hierarchical Planning as Method for Task Analisis: the example of office task analysis*, Behaviour and Information Technology, vol. 7, No. 3, pp. 275-293. INRIA, Rocquencourt, Le Chesnay-Cedex, France,1988.
- SEBILLOTE, S., *Décrire des tâches selon les objectifs des opérateurs. De l'interview à la formalisation [Task description model based on the operators' objectives. From interviews to formalization]*. Le travail Humain, 1991, 54, 193-223.
- 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.

- SILVA, J. C., *Aquisição de Conhecimentos e Manutenção para uma sociedade de Agentes Tutores Artificiais*, Dissertação de mestrado, Universidade Federal da Paraíba, COPIN, 1999.
- SOUZA, C. S. et al., *Projeto de Interfaces de Usuário: Perspectivas Conitivas e Semióticas*. In: XIX CONGRESSO NACIONAL DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 1999, Rio de Janeiro. Anais do XIX Congresso Nacional da Sociedade Brasileira de Computação. Rio de Janeiro: Edições Entrelugar, 1999. v.2. p.425-476.
- SOUSA, M. R. F., *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.
- STEWART, T., Standards Status ISO 9241, 2000.
- TAM, R.C.M.; MAULSBY, D.; PUERTA,A., U-TEL: A Tool for Eliciting User Task Models from Domain Experts, Proceedings IUI'98, ACM Press, 1998.
- 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.
- 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 Shackel, B. Proceedings Interact'90, pp. 163 - 168. Elseviers, North-Holland, Amsterdam, 1990.
- TURNELL,M.F.Q.V., *Notas de Aula – Conceitos e Projeto de Interfaces Usuário – Computador*, Laboratório de Interfaces Homem – Máquina, DEE – CCT/UFPB, 1998.
- VAN WELIE, M., VAN DER VEER, G. C., Eliens A., An Ontology for Task World Models, Proceedings DSV-IS'98, pp-57-70, Springer Verlag, 1998.
- 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.



DePerUSI[®]

DELINEAMENTO DO PERFIL DO USUÁRIO DE SISTEMAS INTERATIVOS



Grupo de Interfaces Homem-Máquina

Este questionário tem o propósito de coletar informações que possibilitem delinear seu perfil de usuário de sistemas computacionais e, em particular, do PRODUTO sob condições de teste. Por favor analise cada aspecto considerado, selecionando a opção que mais adequa à sua condição de usuário e fornecendo as demais informações solicitadas, quando se fizer necessário.

Quaisquer dúvidas no processo de preenchimento do questionário, consultar o suporte técnico nos telefones (021)83 310 1143 ou (021)83 310 1357.

DADOS PESSOAIS:				
ITEM		OPÇÕES		
1	Sua formação:	<input type="checkbox"/> ENSINO MÉDIO	<input type="checkbox"/> CURSO TÉCNICO	<input type="checkbox"/> GRADUAÇÃO 22
2	Você é do sexo:	<input type="checkbox"/> MASCULINO 10	<input type="checkbox"/> FEMININO 12	
3	Você é:	<input type="checkbox"/> DESTRO 18	<input type="checkbox"/> CANHOTO 4	<input type="checkbox"/> AMBIDESTRO
4	Você usa óculos ou lentes de contato?	<input type="checkbox"/> SIM 9	<input type="checkbox"/> NÃO 13	
5	Sua faixa etária é:	<input type="checkbox"/> 18 A 24 ANOS 20	<input type="checkbox"/> 25 A 35 ANOS 2	<input type="checkbox"/> 35 A 45 ANOS <input type="checkbox"/> ACIMA DE 45 ANOS (____ Anos)
6	Sua plataforma computacional é:	<input type="checkbox"/> WORKSTATION	<input type="checkbox"/> PC 21	<input type="checkbox"/> MAC <input type="checkbox"/> OUTRA 1
7	Você tem experiência prévia com sistemas computacionais?	<input type="checkbox"/> SIM 22	<input type="checkbox"/> NÃO 0	
8	Há quanto tempo v. usa sistemas computacionais?	<input type="checkbox"/> MENOS DE 3 MESES	<input type="checkbox"/> DE 3 MESES A 1 ANO	<input type="checkbox"/> MAIS DE 1 ANO 22 <input type="checkbox"/> NUNCA UTILIZEI
9	Com que frequência v. usa sistemas computacionais?	<input type="checkbox"/> DIARIAMENTE 21 <input type="checkbox"/> 1 VEZ POR QUINZENA	<input type="checkbox"/> MENOS DE 1 VEZ POR SEMANA <input type="checkbox"/> MENOS DE 1 VEZ POR QUINZENA	<input type="checkbox"/> 1 VEZ POR SEMANA 1 <input type="checkbox"/> 1 VEZ POR MÊS <input type="checkbox"/> MENOS DE 1 VEZ POR SEMANA <input type="checkbox"/> É A 1ª VEZ QUE USO O PRODUTO
DADOS RELACIONADOS COM O PRODUTO:				
ITEM		OPÇÕES		
10	Você tem experiência prévia com o PRODUTO? Se NÃO , descreva as suas expectativas quanto ao mesmo e depois vá para o item 19.	<input type="checkbox"/> SIM	<input type="checkbox"/> NÃO 22 EXPECTATIVAS:	
11	Há quanto tempo v. usa o PRODUTO?	<input type="checkbox"/> MENOS DE 3 MESES	<input type="checkbox"/> DE 3 MESES A 1 ANO	<input type="checkbox"/> MAIS DE 1 ANO
12	Com que frequência v. usa o PRODUTO?	<input type="checkbox"/> DIARIAMENTE <input type="checkbox"/> 1 VEZ POR QUINZENA	<input type="checkbox"/> MENOS DE 1 VEZ POR SEMANA <input type="checkbox"/> MENOS DE 1 VEZ POR	<input type="checkbox"/> 1 VEZ POR SEMANA <input type="checkbox"/> 1 VEZ POR MÊS <input type="checkbox"/> MENOS DE 1 VEZ POR SEMANA <input type="checkbox"/> É A 1ª VEZ QUE USO O PRODUTO

		QUINZENA				
13	Qual a versão do PRODUTO que v. utiliza atualmente?					
14	Como v. considera o processo de instalação do produto?	<input type="checkbox"/> MUITO FÁCIL	<input type="checkbox"/> FÁCIL	<input type="checkbox"/> NEM FÁCIL NEM DIFÍCIL	<input type="checkbox"/> DIFÍCIL	<input type="checkbox"/> MUITO DIFÍCIL
15	A natureza da principal atividade que v. desenvolve com o auxílio do PRODUTO é essencialmente de:	<input type="checkbox"/> PESQUISA E DESENVOLVIMENTO	<input type="checkbox"/> APLICAÇÃO IMEDIATA DE UTILITÁRIOS	<input type="checkbox"/> TREINAMENTO E APRENDIZAGEM		
16	No contexto de suas atividades v. utiliza o PRODUTO em:	<input type="checkbox"/> Construção de Modelos <input type="checkbox"/> Prestação de Serviços <input type="checkbox"/> Suporte a Disciplinas Teóricas (Graduação)	<input type="checkbox"/> Verificação de Modelos Existentes <input type="checkbox"/> Execução de Projetos <input type="checkbox"/> Suporte a Disciplinas Teóricas (Pós-graduação)	<input type="checkbox"/> Utilização em teses/Dissertações <input type="checkbox"/> Desenvolvimento de Produtos <input type="checkbox"/> Suporte Laboratorial (Graduação)	<input type="checkbox"/> Uso em Trabalhos de Iniciação Científica <input type="checkbox"/> Treinamento em Nível de Extensão <input type="checkbox"/> Suporte Laboratorial (Pós-graduação)	
17	Qual a forma de ajuda do PRODUTO que v. costuma utilizar mais freqüentemente?	<input type="checkbox"/> MANUAL <input type="checkbox"/> HELPDESK (INTERNET)	<input type="checkbox"/> HELPWIN <input type="checkbox"/> CONSULTAS ORAIS A OUTROS USUÁRIOS	<input type="checkbox"/> DEMOS <input type="checkbox"/> SUPORTE TÉCNICO (FONE/E-MAIL/FAX)		
18	Quais as Caixas de Ferramentas que v. costuma utilizar mais freqüentemente?	<input type="checkbox"/> CRIADOR DE BASES <input type="checkbox"/> EDITOR DE FORMULÁRIOS	<input type="checkbox"/> AMBIENTE DE APLICAÇÃO <input type="checkbox"/> ADMINSTRADOR DE SEGURANÇA	<input type="checkbox"/> ADMINISTRADOR DE BASES <input type="checkbox"/> EDITOR DE RELATÓRIOS		
DADOS COMPARATIVOS COM OUTROS PRODUTOS:						
ITEM		OPÇÕES				
19	Você tem experiência prévia com produtos similares? Se NÃO , encerre neste ponto o preenchimento do questionário. Se SIM , especifique o produto e continue o preenchimento do questionário.	<input type="checkbox"/> SIM 2	<input type="checkbox"/> NÃO 20			
		ESPECIFICAÇÃO :				
20	Se v. preencheu o item anterior, especifique o tempo de uso do produto.	<input type="checkbox"/> MENOS DE 3 MESES	<input type="checkbox"/> DE 3 MESES A 1 ANO	<input type="checkbox"/> MAIS DE 1 ANO		
21	Se v. tem experiência com um segundo produto similar, especifique-o.					
22	Se v. preencheu o item anterior, especifique o tempo de uso do produto.	<input type="checkbox"/> MENOS DE 3 MESES	<input type="checkbox"/> DE 3 MESES A 1 ANO	<input type="checkbox"/> MAIS DE 1 ANO		
23	Se v. tem experiência com um terceiro produto similar, especifique-o.					
24	Se v. preencheu o item anterior, especifique o tempo de uso do produto.	<input type="checkbox"/> MENOS DE 3 MESES	<input type="checkbox"/> DE 3 MESES A 1 ANO	<input type="checkbox"/> MAIS DE 1 ANO		

Apêndice 2 – Perfil do usuário

Características do usuário, escolhidas pelo projetista, de acordo com a relevância, para o projeto. Levantamento baseado em:

- ❑ Fatos
- ❑ Opiniões do usuário
- ❑ Dados medidos ou observados

PARTE I - CARACTERÍSTICAS

- Faixa etária: 18 a 60
- Sexo: Masculino ou Feminino
- Habilidades necessárias para executar a tarefa:
 - Habilidade de percepção: Tátil, Visual.
 - Habilidades motoras: Precisão, habilidade para manusear o mouse
- Grau de instrução: Segundo grau completo
- Frequência de execução das Tarefas: Uma vez a cada seis meses (por usuário)
- Objetivos: Modelar tarefas utilizando uma ferramenta de apoio
- Motivações: Facilitar a vida do usuário: Menor trabalho braçal, Verificação de consistência e completude da tarefa. Reduzir erros e Otimizar modelagem.

PARTE II - CONHECIMENTO CONCEITUAL

- Função: Projetista de interfaces e sistemas computacionais. Experiente.
- Tarefa: Analisar e modelar tarefas. Experiente.
- Método: Utilizando uma ferramenta de apoio. O usuário pode ser novico.
- Computador: Necessidade de bom conhecimento. Razoavelmente experiente.

Conhecimento Sintático

Nível de experiência

- | | |
|-----------------------------------------------|------------------|
| - Uso do teclado: | Experiente |
| - Uso de mouse | Experiente |
| - Uso de dispositivos especiais de interação: | Não é necessário |

⁷ Nível de experiência: novico, experiente, especialista, ...

- **Uso de terminologia específica:** Experiência com a terminologia do formalismo TAOS para análise e modelagem de tarefas ou outro formalismo. Experiência com alguma ferramenta de modelagem ou edição (ex: Dreamweaver, Front Page, Rational Rose ou até mesmo Microsoft Word - desenho).

PARTE I - CARACTERÍSTICAS

- **Aprendizado:** Por experiência adquirida para usar a ferramenta; Por tutoriais (livros) ou aulas para modelar tarefas.
- **Capacidade de solucionar problemas:** Razoável
- **Capacidade de reter o aprendizado:** Alta

Personalidade:

- **Nível de curiosidade:** Razoável
- **Nível de persistência:** Alto

- **Nível de inovação:**
 - Inovador**
 - Conservador**

 - Impulsivo**
 - Reflexivo**

	em termos auditivos e traduzidos por palavras únicas suficientemente espaçadas (no tempo), a fim de permitir a discriminação pelo usuário, e																
8.5.4	Capacidade de repetição Deverá ser oferecida.																
Legenda:																	
S = Sim (se aplicável)				A = Análise da Documentação do Sistema				AA = Avaliação Analítica				M = Mensuração					
N = Não (se não aplicável)				E = Evidência documentada				AE = Avaliação Empírica				P = Passou (atendeu à recomendação)					
O = Observação						DM = Método Diferente						F = Falhou (não atendeu à recomendação)					

Legenda**S** = Sim (se aplicável)**N** = Não (se não aplicável)**A** = Análise da documentação do sistema**E** = Evidência Documentada**O** = Observação**AA** = Avaliação Analítica**AE** = Avaliação Empírica**MD** = Método Diferente**M** = Mensuração**P** = Passou (atendeu à recomendação)**F** = Falhou (não atendeu à recomendação)

6.1.3	Valores defaults a) Campo contém valores default onde quer que possível e apropriado para a tarefa, e b) Campos default de texto editáveis																							
6.1.4	Alternando entre dispositivos de entrada Se apropriado para a tarefa, deve ser minimizado o trabalho de alternar entre dispositivos de entrada.																							
6.1.5	Dispositivos de apontamento Se um dispositivo de apontamento pode ser usado para entrada num formulário, ele deveria ser usado também para navegação																							
6.2	Entrada de textos alfanuméricos																							
6.2.1	Justificação das entradas O sistema justifica a entrada, não o usuário.																							
6.2.2	Condução zero Se nenhuma condução é necessária para entrada numérica, o sistema deve provê-las.																							
6.2.3	Múltiplas linhas Se o campo contém múltiplas linhas de texto (i.e. sentenças ou parágrafos): a) tamanho da área de entrada – o tamanho da área de entrada claramente indicada, e b) auto envolver – capacidade de se auto envolver, e c) editar e navegar – convenções normais.																							
6.2.4	Campos exclusivos mutuamente Sugestão visual indica só um dos campos ser usados por vez																							
6.2.5	Regras de interdependência O uso de complexas regras de interdependências do tipo if/then entre campos de entrada deve ser evitada																							
6.2.6	Área do campo de entrada de texto Campos de texto grandes suficientes para acomodar a maioria das entradas sem rolagem																							
6.3	Entradas de escolha																							
6.3.1	Opções de entrada limitadas Mecanismo provê capacitar o usuário visualizar e selecionar as opções disponíveis																							
6.3.2	Sugestões visuais discriminadas Sugestões visuais discriminadas usadas para discriminar entre diferentes tipos lógicos de escolhas de entrada na																							

6.4.2	Identificando e localizando erros																								
	a) Se a validação indicar que o campo foi preenchido com erros, o cursor é colocado no primeiro campo do erro e ao usuário é permitido facilmente mover através dos campos com erro.																								
	b) Se existem dependências entre campos, e se é apropriado para a tarefa, erros potenciais são indicados pelo sistema.																								
6.4.3	Re-entrada de dados																								
	Se o campo contém erros, o usuário não é requerido re-entrar com o dado correto.																								
6.4.4	Áreas não disponíveis																								
	Áreas da tela não disponíveis para entrada do usuário, não acessável pelo usuário e visualmente codificado de acordo.																								
6.4.5	Transmissão fácil																								
	Transmissão dos campos de entrada realizada pelo significado de uma simples ação explícita.																								
6.4.6	Controle do usuário																								
	A menos que seja óbvio para o usuário, o formulário deve indicar como realizar as seguintes ações: sinalize a conclusão do formulário e apresente um formulário vazio para a entrada de dados novos; sinalize a conclusão do formulário e apresente a versão previamente terminada do formulário ou uma versão default do formulário; permita a saída do formulário sem mudar nenhuns dados no sistema (opção cancelar)																								
6.4.7	Salve temporariamente																								
	Se apropriado para a tarefa e as restrições do sistema permitam, uma função para salvar temporariamente deveria ser provida.																								
6.5	Validação do campo																								
6.5.1	Validação do campo simples																								
	Se a capacidade do sistema permite, os campos de entrada devem ser checados antes da aceitação.																								
6.5.2	Validação de múltiplos campos																								
	Se dependência entre campos no formulário, ou entre campos com outra incidência no mesmo formulário, validação adicional deve ser checada.																								
7	Feedback (Retorno ao usuário)																								

