



UNIVERSIDADE FEDERAL DA PARAÍBA
Centro de Ciências e Tecnologia – CCT
Departamento de Sistemas e Computação – DSC
Coordenação de Pós-Graduação em Informática - COPIN

**VALIDAÇÃO DO FORMALISMO TAOS PARA A
ANÁLISE DA TAREFA NO CONTEXTO DA CONCEPÇÃO
DE INTERFACES HOMEM-COMPUTADOR**

IVETTE KAFURE MUÑOZ

BERNARDO LULA JUNIOR, DR.

(Orientador)

Campina Grande – PB, Brasil

©Ivette Kafure Muñoz, fevereiro de 2000

IVETTE KAFURE MUÑOZ

**VALIDAÇÃO DO FORMALISMO TAOS PARA A
ANÁLISE DA TAREFA NO CONTEXTO DA CONCEPÇÃO
DE INTERFACES HOMEM-COMPUTADOR**

Dissertação de Mestrado submetida à Coordenação do
Curso de Pós-Graduação em Informática da Universi-
dade Federal da Paraíba – Campus II como parte dos
requisitos necessários para obtenção do grau de Mestre.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Design de Produtos de Informática

BERNARDO LULA JUNIOR, DR.

(Orientador)

Campina Grande – PB, Brasil

©Ivette Kafure Muñoz, fevereiro de 2000

Ficha Catalográfica

K11V

Kafure, Ivette

Validação do Formalismo TAOS para a Análise da Tarefa no Contexto da Concepção de Interfaces Homem-Computador

128 p. II

Dissertação de Mestrado, Universidade Federal da Paraíba, CCT, COPIN, Campina Grande, PB, 2000.

Orientador: Bernardo Lula Junior, Dr.

1. Interface Homem-Computador
2. Tarefa
3. TAOS
4. MAD

CDU = 519.683B

**VALIDAÇÃO DO FORMALISMO TAOS PARA A ANÁLISE DA
TAREFA NO CONTEXTO DA CONCEPÇÃO DE
INTERFACES HOMEM-COMPUTADOR**

IVETTE KAFURE MUÑOZ


DISSERTAÇÃO APROVADA EM 25.02.2000



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



**PROF. JOSÉ HAMURABI N. DE MEDEIROS, M.Sc
Examinador**



**PROF^ª MARIA DE FÁTIMA Q. V. TURNELL, Ph.D
Examinadora**



**PROF. FÁBIO PARAGUAÇU DUARTE DA COSTA, Dr.
Examinador**

CAMPINA GRANDE – PB

Dedico este Trabalho...

A los que aprenden del riesgo

A las cosas simples de la vida

Aquella sonrisa

En fin, a los que construyen en medio de las cenizas.

Aos que aprendem do risco

Às coisas simples da vida

Aquele sorriso

Em fim, aos que constróem no meio das cinzas.

SONETO A MI LABOR

A mis compañeros todos

Muy cerca de completar el medio Siglo,
de estar laborando en esta Empresa,
quiero decir sin ninguna ligereza
que este há sido mi mejor designio.

Ha transcurrido el tiempo tan ligero,
que yo mismo no sé ni cómo referirlo,
porque ha sido como fácil ejercicio,
que me ha suministrado todo lo que quiero.

La ausencia de algunos compañeros
deja muchas nostalgias y recuerdos
que cimbran lo profundo de la mente.

Y sin embargo persisto estar presente,
porque este medio de actividad es atrayente
como un faro que ilumina mis senderos.

Marino Kafure

Cali, Septiembre 1, 1994

Agradecimentos

OBRIGADA:

As pessoas que fizeram possível este intercâmbio entre Brasil e Colômbia: Patricia Kafure, Bernardo Lula, Claudete Mary Alves, Claudia Kafure, Marino Kafure, Marina Muñoz de Kafure, Marcus Costa Sampaio, Juan Francisco Díaz, Camilo Rueda, Marcelo A. Barros, Ana Lúcia Guimarães e Vera Lúcia de Oliveira Cruz.

A Bernardo por me orientar incansavelmente e me oferecer sua família como sendo a minha.

Aos professores Fátima e Hamurabi por seus conselhos e aportes neste trabalho.

A todo o pessoal da UFPB que me seria inumerável de descrever por me acolher e colaborar de uma ou outra forma.

A Gustavo pela força, motivação e carinho.

A meus amigos pela confiança e amizade.

A minha família por tanto AMOR.

A CAPES pelo apoio.

A Deus por TUDO.

Resumo

Apesar do progresso recente no domínio da concepção de interfaces homem-computador (IHC) de sistemas computacionais (SC), falta compatibilidade entre a representação cognitiva que o usuário tem de sua tarefa e a representação computacional (funcional) da mesma. Dada essa situação, este trabalho aborda três pontos importantes: o estado da arte no domínio da análise e da modelagem da tarefa e suas contribuições no projeto de IHC de SC, o estudo dos formalismos MAD (*“Méthode Analytique de Description”*) e TAOS (*“Task and Action Oriented System”*) e a validação do formalismo TAOS como ferramenta para a análise da tarefa no contexto de concepção de interfaces homem-computador.

Abstract

In spite of the recent advances in the human-computer interaction field, there is a lack of compatibility between the cognitive and functional representation of the user task. Therefore this work reviews the state of the art in terms of task analysis and closely examines two formalisms in the context of task analysis and description. The first formalism is MAD (*"Méthode Analytique de Description"*) proposed specifically to support the task analysis. The second formalism, developed for another context, TAOS (Task and Action Oriented System) was validated in this work as an adequate tool for user task analysis and representation in the interface design process.

Sumário

1. Introdução

1.1 Interação Homem-Computador	1
1.2 Interface Homem-Computador	2
1.3 Contribuições da Ergonomia Cognitiva	3
1.4 Análise da Tarefa	4
1.5 Modelagem da Tarefa	6
1.6 TAOS (<i>Task and Action Oriented System</i>)	7
1.7 Objetivo e Hipóteses	8
1.8 Metodologia e Organização do Trabalho	9

2. A Tarefa

2.1 Introdução	12
2.2 Teoria da Ação	13
2.3 Requisitos da Análise da Tarefa	18
2.4 Modelagem da Tarefa	21
2.5 TKS (<i>Task Knowledge Structure</i>).....	23
2.6 MAD (<i>Méthode Analytique des Description des tâches</i>)	24
2.7 Discussão	26
2.8 Conclusão	27

3. MAD e a Análise da Tarefa

3.1 Introdução	28
3.2 Metodologia MAD	29
3.3 Tarefa-Exemplo: Navegar na Web	31
3.4 Formalismo MAD	33
3.5 Representação MAD da Tarefa-Exemplo	36
3.6 Representação Orientada-Objeto de MAD	38
3.7 Satisfação dos Requisitos	41
3.8 Algumas Limitações do MAD	44
3.9 Conclusão	45

4. Formalismo TAOS	
4.1 Introdução	46
4.2 Modelo TAOS	46
4.3 Conceitos Estáticos	48
4.4 Conceitos Dinâmicos	51
4.5 A Representação TAOS da Tarefa-Exemplo	54
4.6 Conclusão	58
5. TAOS e a Análise da Tarefa	
5.1 Introdução	59
5.2 Satisfação dos Requisitos	59
5.3 Análise Comparativa MAD x TAOS	61
5.4 Conclusão	69
6. Conclusões, Contribuições e Trabalhos Futuros	
6.1 Conclusões	70
6.2 Contribuições	72
6.3 Sugestões e Trabalhos futuros	73
A. Modelagem da Tarefa-Exemplo Navegar na Web em MAD	
A.1 Notação MAD	74
A.2 Descrição da Árvore Hierárquica da Tarefa	74
A.3 Definição de Cada Objeto-Tarefa	78
B. Modelagem da Tarefa-Exemplo Navegar na Web em TAOS	
B.1 Notação TAOS	91
B.2 Descrição da Árvore Hierárquica da Tarefa	92
B.3 Definição de Cada Classe	95
Referências Bibliográficas	115
Bibliografia	125

Lista de Figuras

2.1	Tipo de Modelos	14
2.2	Elementos do Modelo TKS [Markopoulos 1997]	24
3.1	Decomposição Hierárquica da Tarefa Navegar na Web.....	38
3.2	Elementos do Modelo MAD	40
4.1	Hierarquia de Conceitos de TAOS	47
4.2	Tarefa Navegar na Web em TAOS.....	58
5.1	Utilização do Construtor ALT em MAD	66
A.1	Notação MAD	74
A.2	Decomposição da Tarefa NW	75
A.3	Sub-Tarefa Determinar Endereço de NW	75
A.4	Sub-Tarefa Editar Endereço de NW	75
A.5	Sub-Tarefa Manipular Endereço de NW	76
A.6	Sub-Tarefa Abrir Arquivo de NW	76
A.7	Sub-Tarefa Procurar Endereço de NW.....	76
A.8	Sub-Tarefa Salvar Arquivo de NW.....	77
A.9	Sub-Tarefa Imprimir Arquivo de NW.....	77
A.10	Sub-Tarefa Editar Arquivo de NW	77
A.11	Sub-Tarefa Ir Para de NW.....	78
A.12	Sub-Tarefa Solicitar Ajuda de NW	78

B.1	Notação MAD	91
B.2	Plano (Tarefa) Navegar na Web	92
B.3	Sub-Plano (Sub-Tarefa) Determinar Endereço de NW	92
B.4	Sub-Plano Manipular Arquivo de NW	92
B.5	Sub-Plano Editar Arquivo de NW	93
B.6	Sub-Plano Ir Para de NW	93
B.7	Sub-Plano Abrir Arquivo de NW	93
B.8	Sub-Plano Salvar Arquivo de NW	94
B.9	Sub-Plano Imprimir Arquivo de NW	94
B.10	Sub-Plano Procurar Endereço de NW	94

Capítulo 1 - Introdução

1.1 Interação Homem-Computador

Para um usuário de um sistema computacional (ou computador, como é popularmente entendido), a finalidade última do sistema é auxiliá-lo na realização de uma tarefa. Para ele, o sistema é apenas uma ferramenta e, como tal, exige, para ser utilizado, um conhecimento e habilidade não apenas no domínio da tarefa, mas também no domínio das funcionalidades e dos meios disponíveis no sistema para utilizá-las. Aprender a usar o sistema é uma tarefa adicional derivada da necessidade de aplicação do sistema para realização da tarefa original. Obviamente, para o usuário, essa tarefa ou esforço adicional de ter que aprender a usar o sistema deve ser o menor possível. Para isso, é fundamental que o sistema, ou a operação do sistema seja adaptada às características do(s) usuário(s), da tarefa e do ambiente de trabalho.

Tradicionalmente, o projeto de sistemas computacionais tem sido uma matéria que afeta exclusivamente aos profissionais da computação. Por outro lado, usuários e pessoas são estudados pela psicologia cognitiva enquanto tarefas e ambientes de trabalho pertencem aos campos da psicologia organizacional e da ergonomia ([Haan 1992]). No entanto, o projeto de sistemas computacionais que sejam fáceis de aprender e de usar devem incorporar contribuições de todas essas disciplinas.

A *Ergonomia Cognitiva*, como também é hoje chamado o campo de estudos e pesquisas sobre a interação homem-computador, é um campo

multidisciplinar que investiga o uso de teorias e métodos da psicologia e da ergonomia no projeto de sistemas computacionais adequados a seus usuários, ambientes e tarefas.

1.2 Interface Homem-Computador

De acordo com a Teoría da Ação de Donald Norman ([Norman 1983]), existe inevitavelmente um *fosso* ou *distância* entre a maneira pela qual o usuário concebe sua tarefa e a maneira de realiza-la com o apoio de um sistema computacional (ou computador). E o responsável por esse fosso é a interface de comunicação do usuário com o sistema, ou simplesmente, a *sua interface homem-computador*.

Para o usuário, a interface é o meio onde se opera a comunicação com o sistema utilizado para auxiliá-lo na sua tarefa: ela permite estabelecer um contato físico, perceptivo e conceitual entre ele e o sistema. Para o projetista, a interface é o agente do sistema (elemento de *software* e *hardware*) encarregado de implantar a *representação perceptível* do sistema (sua manifestação externa ou *imagem*) e de assegurar a conexão entre de um lado essa representação e os órgãos cognitivos, sensoriais e motores do usuário, e de outro lado, essa representação e as funcionalidades definidas pelo sistema. A interface tem, portanto, o papel de estabelecer a comunicação e de tornar compreensível para cada um dos intervenientes (usuário e sistema) as mensagens trocadas durante a interação.

Assim, a qualidade da comunicação depende fortemente da compatibilidade (adequação) entre a representação proposta pela interface e a representação mental que o usuário tem de sua tarefa. Quanto maior o fosso ou distância entre essas representações, maior o esforço sensorio-motor-cognitivo que o usuário deve despender para realizar sua tarefa com o uso do sistema. Inversamente, quanto menor a distância, maior facilidade de

uso, de aprendizado e maior satisfação com a utilização do sistema, o que implica numa maior *usabilidade*¹ do sistema.

Portanto, para assegurar um alto grau de usabilidade do sistema, o processo de concepção de um sistema computacional, ou mais particularmente, de sua interface, deve necessariamente se apoiar sobre o conhecimento que o usuário tem de sua tarefa, das suas funcionalidades, e dos meios disponíveis para sua realização.

1.3 Contribuições da Ergonomia Cognitiva

Um número considerável de estudos em ergonomia e em psicologia cognitiva tem contribuído significativamente nas duas últimas décadas para a compreensão e a análise dos aspectos envolvidos na interação homem-computador. Tais contribuições dizem respeito a princípios de concepção, recomendações ergonômicas, teorias e modelos cognitivos. Podemos citar, entre outros: modelagem do usuário (modelo do Processador Humano [Card 1983]), modelagem da interação homem-computador (Teoria da Ação [Norman 1983]), modelagem da interface (modelo lingüístico [Green 1985]), modelo do processo de concepção do sistema (CLG [Moran 1981]), modelagem de desempenho (GOMS [Card 1983]), guia ergonômico ([Scapin 1987]). No entanto, essas contribuições carecem principalmente de uma formalização que responda ao problema concreto da integração explícita dos aspectos do usuário e de sua tarefa no processo de concepção e desenvolvimento de interfaces. Na ausência de métodos ou técnicas formais, os projetistas têm-se apoiado inevitavelmente em ferramentas que facilitam a realização e a reutilização de protótipos de interfaces (geradores de interface ou UIMS²). A integração dos fatores humanos passa ainda pela

¹ Facilidade para usar, aprender e lembrar

² User Interface Management Systems.

colaboração com ergonomistas (principalmente para avaliação de protótipo/produto final) e/ou pela utilização de manuais ou guias de estilos.

Hoje, um dos objetivos principais da ergonomia cognitiva, partindo das contribuições e conceitos introduzidos pelos diferentes trabalhos já realizados, concerne a representação e o estabelecimento de uma correspondência formal (no sentido da informática) de modelos cognitivos (tarefa, usuário) e modelos da interação (conceitualização da interface) incorporando resultados da ergonomia (critério, recomendações, etc.), de maneira a permitir a concepção, especificação e desenvolvimento de interfaces que respondam aos objetivos e características do usuário. Dentro desta perspectiva, podemos citar os trabalhos de Johnson [Johnson 1988; Johnson 1991], Markopoulos [Markopoulos 1992, Markopoulos 1997], Shneiderman [Shneiderman 1998], Scapin [Scapin 1989], que enfatizam a necessidade de/ou propõem representações adequadas da tarefa que permitam a sua manipulação e análise, e de mecanismos que permitem estabelecer a partir da representação da tarefa uma especificação conceitual da interface.

O domínio de estudo abordado neste trabalho é a modelagem da tarefa do usuário no contexto da concepção de interfaces homem-computador de sistemas computacionais.

1.4 Análise da Tarefa

Um princípio fundamental da ergonomia é o conhecimento do usuário e do trabalho a ser realizado ([Cibys 1996, Sebillote 1995b]). O trabalho é visto segundo dois componentes básicos: a tarefa e a atividade.

A tarefa ou trabalho prescrito refere-se àquilo que a pessoa deve realizar. Trata-se de “o que é para fazer”, o objetivo a alcançar. Sua análise

é feita a partir de entrevistas dirigidas ou outros métodos habituais de pesquisa, buscando evidenciar as características do processo de realização. A tarefa é descrita em termos de objetivos, procedimentos, decomposição em sub-tarefas, restrições, etc.

A atividade ou trabalho realizado, refere-se ao modo como a pessoa realmente realiza sua tarefa. A análise da atividade é feita através de observações “in loco” ou a partir do *log* de sessões de trabalho real. São colhidas informações sobre as operações efetuadas, seu encadeamento, suas dificuldades, frequência de uso, etc.

Tradicionalmente, a metodologia proposta pela Análise Ergonômica do Trabalho (AET) prevê o encadeamento de 2 etapas de análise: a análise da tarefa e a análise da atividade. Na primeira etapa, são realizadas entrevistas dirigidas com usuário(s) buscando-se uma descrição apropriada da tarefa de modo a evidenciar as características do processo de realização. O reconhecimento do processo da tarefa é particularmente importante quando o objetivo da análise é a concepção de um novo sistema ou de um sistema informatizado que venha apoiar o sistema atual. A etapa seguinte da análise refere-se à validação das descrições e informações que foram coletadas e que compõem as representações sobre o trabalho. Essa etapa prevê a realização de observação “in-loco” do trabalho realizado pelo usuário com o sistema. O resultado da análise é um documento (relatório) contendo uma descrição detalhada e hierarquizada do trabalho segundo o ponto de vista do usuário. O relatório deve também prever recomendações sobre as funcionalidades a serem projetadas para o sistema, recomendações ergonômicas para a concepção da interface com o usuário do futuro sistema e, se for o caso, um diagnóstico das situações problemáticas e as soluções possíveis. 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]).

1.5 Modelagem da Tarefa

Os trabalhos em análise da tarefa a partir da perspectiva da ergonomia cognitiva têm-se concentrado na construção de modelos preditivos de desempenho ou de avaliação da interação do usuário com um sistema particular, como, por exemplo, GOMS [Card 1983], CCT [Kieras 1985] e UAN [Hix 1993]. Esses modelos podem produzir predições/avaliações sobre o desempenho do usuário ou do tempo necessário para realizar uma tarefa sem a necessidade de utilização do sistema. No entanto, eles não ajudam o projetista no processo de concepção de um sistema. Nesse contexto, a análise da tarefa diz respeito às representações mentais que o usuário possui da tarefa, independentemente de uma situação particular de trabalho.

O problema da representação e da manipulação do conhecimento é um problema comum encontrado nos campos de psicologia, da ergonomia e da informática, particularmente na inteligência artificial (IA). A esse respeito, a contribuição da IA vem sendo fundamental para a obtenção de métodos formais de representação do conhecimento que o usuário possui acerca de sua tarefa. Alguns trabalhos pioneiros em ergonomia cognitiva tem sido desenvolvidos, incorporando uma contribuição expressiva da IA para a modelagem da tarefa do usuário objetivando a obtenção de um formalismo próprio, tanto para o processo de descrição e análise da tarefa quanto para o processo de produção de uma especificação conceitual da interface a partir da descrição da tarefa. Dois desses trabalhos são hoje reconhecidos como representativos do esforço que vem sendo dedicado nos últimos 10 anos nessa direção: MAD e TKS.

Tanto MAD (*Méthode Analytique de Description*), introduzido por Scapin e Pierret-Golbreich [Scapin 1989], quanto TKS (*Task Knowledge Structure*), introduzido por Johnson e outros [Johnson 1988], são

formalismos de descrição de tarefas que assumem que o conhecimento que uma pessoa tem acerca de uma tarefa é hierarquicamente estruturado de acordo com o paradigma do planejamento hierárquico (PH) de Sacerdoti [Sacerdoti 1974a, Sacerdoti 1974b, Sacerdoti 1975]. Ambos os métodos manipulam mais ou menos os mesmos conceitos (tarefas, ações, objetivos, etc.) e têm por base a exploração de diferentes elos funcionais e estruturais entre eles. As estruturas formais utilizadas para descrever os conceitos são baseadas no conceito de *frame* também oriundo da IA ([Minsky 1975]).

1.6 TAOS (*Task and Action Oriented System*)

TAOS é um formalismo de aquisição e representação do conhecimento baseado na modelagem do domínio, desenvolvido por J.H. de Medeiros ([Medeiros 1995a, Kessel 1995, Medeiros 1995b, Medeiros 1995c]). TAOS se insere entre os trabalhos teóricos e experimentais desenvolvidos nesses últimos anos em IA no campo da construção de sistemas baseados em conhecimento (SBC¹) fundado na modelagem do domínio. Esses trabalhos têm mostrado as vantagens de metodologias baseadas na construção de modelos a partir da fase de aquisição de conhecimento até a fase de projeto do SBC [Silva 1999].

TAOS é centrado na análise das tarefas a serem executadas no domínio. Essa análise dá origem a uma conceitualização, isto é, a uma representação estruturada das entidades ou conceitos estáticos (objetos, métodos e situações) e dinâmicos (processos, planos e ações que operam sobre as entidades estáticas).

¹ Sistemas Baseados em Conhecimento são sistemas que fazem uso de conhecimento e o representam explicitamente.

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. Tal construção baseia-se em uma metodologia ascendente-descendente. Nesta metodologia deve ser construída uma hierarquia de planos 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. As entidades estáticas e as relações entre essas entidades (com os respectivos atributos), serão catalogadas à medida em que os métodos destinados à execução o exijam. A primeira fase da aquisição é, portanto, ascendente. A segunda fase, descendente, serve-se do esquema já construído para completar o processo de aquisição, a fim de acrescentar novos conhecimentos, e/ou validar os conhecimentos adquiridos até então. 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. Isso permite modelar o conhecimento do domínio fazendo uma verificação preliminar desse conhecimento, tanto do ponto de vista da sua completude (aquisição de conhecimento completo) quanto de sua coerência.

1.7 Objetivo e Hipóteses

MAD e TKS são formalismos para descrição e análise de tarefas projetados por ergonomistas e/ou psicólogos cognitivos com vistas à concepção de interfaces homem-computador e que incorporam elementos conceituais oriundos da IA. TAOS é um formalismo oriundo da IA concebido para a aquisição e representação do conhecimento sobre um domínio e foi validado no domínio da biologia molecular.

O objetivo deste nosso trabalho é demonstrar ou apresentar evidências suficientes para a confirmação das seguintes hipóteses:

Hipótese 1: A tarefa e a análise da tarefa do usuário são conceitos fundamentais a serem considerados no processo de concepção de interfaces homem-computador afim de se construir sistemas computacionais com alto grau de usabilidade.

Hipótese 2: O formalismo MAD é um elemento de solução para o problema de representação e análise da tarefa no processo de concepção de interfaces homem-computador de sistemas computacionais.

Hipótese 3 (Hipótese central): TAOS é um formalismo de aquisição e representação do conhecimento que pode ser usado convenientemente para a representação e análise de tarefas do usuário visando a concepção de interfaces homem-computador de sistemas computacionais.

1.8 Metodologia e Organização do Trabalho

Adotaremos a seguinte metodologia para demonstrar ou apresentar evidências suficientes para confirmar as nossas hipóteses de trabalho:

- 1- mostrar que MAD atende aos requisitos mínimos exigidos para a descrição e a análise da tarefa;
- 2- mostrar que TAOS também atende aos requisitos mínimos exigidos para a descrição e a análise da tarefa;
- 3- definir um caso representativo de uma tarefa usual realizada através de uma interação com um sistema computacional (tarefa-

exemplo), descrevê-la utilizando os dois formalismos MAD e TAOS e comparar as descrições obtidas.

Assim, adotaremos um exemplo padrão de tarefa (o caso) que usaremos durante todo o desenrolar do trabalho e a partir do qual construiremos inicialmente uma descrição MAD e posteriormente uma descrição TAOS, e que servirá de elemento de convicção (de evidência) da veracidade de nossas hipóteses. A tarefa-exemplo é a tarefa de navegar na web, que descreve a utilização de uma ferramenta para obtenção de uma informação em *sites da Internet*.

O trabalho foi organizado em 6 capítulos e 2 anexos com os seguintes conteúdos:

O capítulo 2 apresenta inicialmente um extrato da Teoria da Ação ([Norman 1983], que confirma a nossa primeira hipótese (hipótese 1) da necessidade fundamental de se considerar a tarefa e sua análise no processo de concepção de interfaces homem-computador de sistemas computacionais com alto grau de usabilidade. Também são apresentados nesse capítulo os requisitos mínimos exigidos para tal e alguns dos principais formalismos utilizados para aquele fim.

O capítulo 3 descreve o formalismo MAD, a sua satisfação dos requisitos mínimos exigidos para análise da tarefa e apresenta sua utilização para tal através da representação da tarefa-exemplo.

O capítulo 4 descreve o formalismo TAOS segundo sua perspectiva de aquisição e representação do conhecimento e apresenta sua utilização para a descrição e análise de tarefa através da representação da tarefa-exemplo.

O capítulo 5 apresenta as evidências que justificam nossa hipótese central, ou seja, uma análise mostrando que TAOS satisfaz os requisitos mínimos exigidos para a representação e a análise da tarefa do usuário e

sua utilização através da representação da tarefa-exemplo, mostrando suas vantagens e desvantagens em relação a MAD.

O capítulo 6 encerra o trabalho com as conclusões, sugestões de continuidade e trabalhos futuros.

O anexo A detalha a descrição da tarefa-exemplo navegar na web em MAD.

O anexo B detalha a descrição da tarefa-exemplo navegar na web em TAOS.

Capítulo 2 - A Tarefa

2.1 Introdução

Como vimos no capítulo anterior, o papel de uma interface homem-computador de um sistema computacional é estabelecer a comunicação entre o usuário e o sistema e tornar compreensível para cada um dos intervenientes (usuário e sistema) as mensagens trocadas durante a interação. No entanto, se do lado do sistema os conceitos utilizados são precisos e formais, do lado do usuário não existe modelo geral satisfatório que englobe todos os seus aspectos comportamentais e que possa ser utilizado para modelar o conjunto dos processos sensoriais, fisiológicos e psicológicos presentes na comunicação homem-sistema computacional. A ciência cognitiva apresenta modelos e teorias interessantes mas muito restritos, reportando-se cada um a fenômenos particulares (linguagem, visão, memória, representação mental, etc.) e não há suporte formal ou automático para a utilização desses conhecimentos em um modelo computacional. Os resultados práticos da pesquisa nessa disciplina são ainda muito modestos: *“les travaux d’ergonomie de logiciels sont encore souvent trop qualitatifs pour être exploitables”*¹ dizia Senach em [Senach 1990] e a situação não mudou muito desde lá. Os principais resultados são os guias ergonômicos propostos por ergonômistas e os guias de estilo propostos por ambientes de desenvolvimento de sistemas. No entanto, alguns trabalhos podem ser explorados para explicar e descrever os

¹ Os trabalhos de ergonomia de software são ainda muito qualitativos para serem explorados (utilizados).

mecanismos cognitivos envolvidos no processo de comunicação e determinar as causas das dificuldades que podem ocorrer. Nesse aspecto, a Teoria da Ação exerce um papel essencial na compreensão da importância da tarefa do usuário no processo de concepção de interfaces, ao descrever as atividades cognitivas postas em jogo durante a realização de uma tarefa com o auxílio de um sistema computacional e ao identificar os pontos de possíveis problemas.

2.2 Teoria da Ação

A Teoria da Ação ([Norman 1983]) é fundada na noção de *modelo mental*. A noção de *modelo mental* corresponde ao fato de que um indivíduo se constrói uma representação mental de seu próprio modelo e do ambiente com o qual ele interage. Esta representação lhe serve de base para elaborar suas intenções, traduzi-las em ações e interpretar o resultado dessas ações. Os modelos mentais refletem, portanto, os conhecimentos do indivíduo e a organização dos mesmos. Por conseguinte, os modelos mentais evoluem constantemente em função da interação com o ambiente e da experiência adquirida. Diversos trabalhos de psicologia cognitiva têm mostrado o papel fundamental da manipulação de modelos mentais na resolução de problemas e na aprendizagem e utilização de dispositivos interativos complexos ([Johnson-Laird 1981, Rasmussen 1980, Young 1981, Kieras 1984, Falzon 1989]).

De acordo com Norman, cada indivíduo tem seus próprios modelos que guiam seu comportamento. No contexto de um processo geral de comunicação, o sucesso da comunicação está ligado à capacidade de cada participante compreender o modelo mental do outro. Na comunicação entre indivíduos, cada modelo é evolutivo adaptativo: os diferentes indivíduos em presença confrontam e adaptam suas representações mentais. No caso da

interação homem-computador, o usuário adapta sua representação mental a partir da representação perceptível do sistema (imagem). Essa imagem é construída pelo projetista da interface a partir de sua própria representação mental do sistema e da representação mental que ele possui do usuário-tipo. A figura 2.1 abaixo ilustra essa idéia:

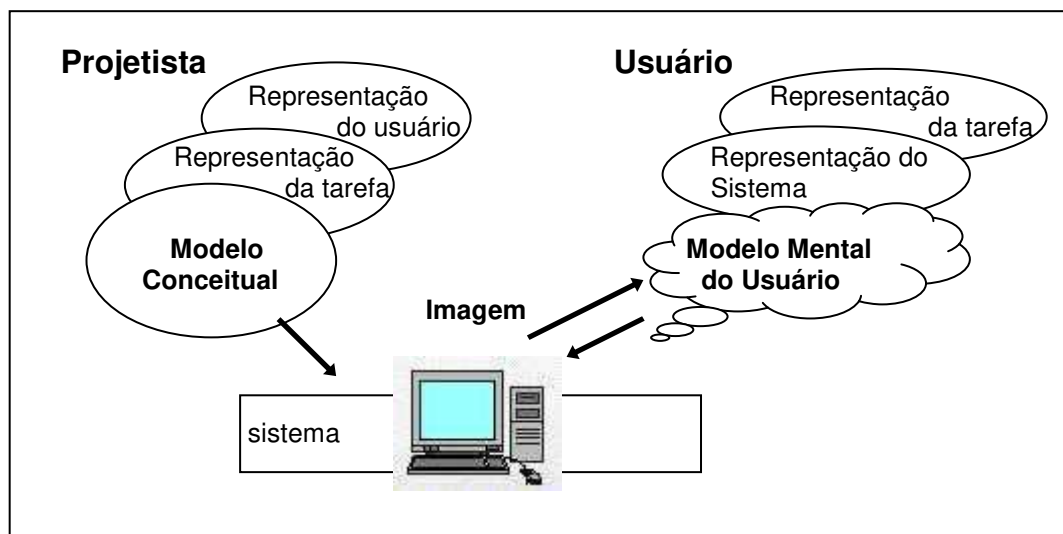


Figura 2.1: Tipos de Modelos

Norman diferencia três tipos de modelos: o *modelo mental do usuário*, a *imagem do sistema* e o *modelo conceitual do sistema computacional*.

O *modelo mental do usuário* é a representação mental que o usuário elabora a propósito do sistema. Ele resulta da interpretação que o usuário faz da imagem do sistema e de sua própria concepção da tarefa. Podemos falar na verdade de um *modelo mental sobre o sistema* e de um *modelo mental sobre a tarefa*. A *imagem do sistema* inclui desde os aspectos físicos de visualização do computador e dispositivos de comunicação até o estilo de interação, forma e conteúdo da informação. O terceiro tipo de modelo que Norman distingue é o *modelo conceitual do sistema computacional*. O *modelo conceitual* é criado pelos projetistas para prover uma representação precisa e completa de um sistema computacional.

De acordo com Norman, a realização de uma tarefa com o auxílio de um sistema computacional coloca em confronto os modelos mentais (do sistema e da tarefa) e a imagem do sistema e tem um caráter iterativo, compreendendo as seguintes etapas cognitivas:

1. *estabelecimento de um objetivo ou meta* (expresso em termos do modelo mental da tarefa);
2. *formulação de uma intenção* (expresso em termos do modelo mental da tarefa);
3. *especificação do plano de ação* (é a tradução da intenção em uma seqüência de ações expressas em termos do modelo mental do sistema. É uma representação mental de um plano de ação);
4. *execução do plano de ação* (implica a utilização do sistema motor, sensorial e cognitivo para execução do plano com os dispositivos de interação disponíveis. Provoca mudança na imagem);
5. *percepção da imagem* (projeção da nova imagem no modelo mental do sistema);
6. *interpretação da imagem* (interpretação da imagem em termos do modelo mental da tarefa);
7. *avaliação* (comparação do resultado da interpretação com o objetivo visado);

O usuário itera globalmente sobre as sete etapas descritas até a realização da tarefa projetada ou até seu abandono definitivo. A decomposição da tarefa pode fazer apelo recursivamente a um novo processo iterativo para a realização de uma sub-tarefa. Na prática, a atividade real do usuário não progride sempre seguindo seqüencialmente as diferentes etapas apresentadas que podem se desenrolar em uma ordem qualquer. Por exemplo, o usuário pode encadear duas ações sem fazer a

avaliação dos resultados da primeira se sua habilidade o permite. Além disso, sua atividade pode mesmo cessar de ser uma atividade de planificação para se tornar uma atividade do tipo reativa aos eventos e aos dados. No entanto, o esquema proposto pela Teoria da Ação fornece um quadro descritivo aproximativo do processo cognitivo posto em ação por um usuário não-especialista durante a realização de uma tarefa.

Norman exprime a diferença entre as representações da imagem e aquelas do modelo mental do usuário pela noção de *distância* (*fosso, barreira*). E distingue a *distância de execução* da *distância de avaliação*: a *distância de execução* denota o esforço de tradução entre o modelo mental do usuário e a representação física externa imposta pela imagem e a *distância de avaliação* traduz o esforço cognitivo inverso, isto é, o esforço de adaptar sua representação mental à nova imagem do sistema e de interpretar essa representação em termos compatíveis com suas intenções.

O usuário precisa “traduzir” em duas direções. Na primeira, o usuário chega ao sistema computacional com uma tarefa em mente. Ele terá que saber como traduzir e expressar a tarefa em procedimentos operacionais e em comandos providos pelo sistema computacional. Na segunda, depois que o comando tiver sido fornecido ao sistema, o usuário deverá saber como interpretar o comportamento do sistema, como também determinar o sucesso ou a falha na experiência da realização da tarefa. Isto é, o usuário tem que saber como traduzir os objetivos abstratos de alto nível da tarefa em ações físicas na interação com o computador. E saber também como interpretar/avaliar as respostas físicas do sistema em relação aos objetivos da tarefa ([Nielsen 1986, Nielsen 1990, Haan 1992, Haan 1999]).

Essas barreiras, que o usuário deve “saltar” a cada iteração durante a realização de sua tarefa, são inevitáveis, e só há duas maneiras de as reduzir: (i) da parte do sistema, por intermédio de sua imagem, ou (ii) da parte do usuário, pelo desenvolvimento de modelos mentais apropriados.

Como a atividade do usuário já é organizada, seja pela formação, experiência da tarefa ou a prática de tarefas análogas (com ou sem sistema computacional), essas distâncias devem ser diminuídas, agindo-se sobre a imagem do sistema. Em caso contrário, o usuário seria estrangido a uma reorganização mental que poderia conduzir a uma degradação do desempenho ou mesmo ao abandono da atividade ([Bisseret 1979, Falzon 1989]).

De acordo com a Teoria da Ação, portanto, as diferenças (distâncias) entre as representações propostas pela imagem do sistema e os conceitos e objetos do universo conceitual (mental) do usuário estão na origem das dificuldades do usuário em utilizar o sistema para a realização de sua tarefa. O objetivo do projetista de interface é, portanto, gerar uma representação adequada do sistema para reduzir a distância entre os modelos mentais do usuário e a imagem do sistema. Isto é, produzir uma imagem a mais adaptada possível quer sobre o plano perceptível (vista, audição etc.) quer sobre o plano conceitual (tarefa), de modo a induzir no usuário um modelo do sistema compatível com seus objetivos, habilidades e competências.

Assim, o ponto de partida de um projeto de interface homem-computador deve ser o recenseamento dos conceitos da tarefa conhecidos do usuário e dos conceitos pertinentes para o domínio (estudo da tarefa e estudo do domínio), produzindo um modelo conceitual que o projetista supõe que o usuário tem de sua tarefa (modelo da tarefa). Sistemas (mais particularmente interfaces) bem projetados são aqueles em que os modelos conceituais dos projetistas são oriundos de uma análise aprofundada da tarefa e do domínio e que formam a base para a imagem do sistema, o que, por sua vez, é a base do modelo mental do usuário (modelo do sistema). Isto produz uma imagem do sistema, materiais de instrução (manuais) e modelos mentais do usuário coerentes, o que facilita o uso e o aprendizado do sistema desenvolvido. Ou seja, produz um sistema com um alto grau de usabilidade.

2.3 Requisitos da Análise da Tarefa

A tarefa se refere àquilo que os usuários (ou operadores) devem realizar durante a preparação/operação/manutenção de um sistema. Como vimos na seção anterior, os usuários de um sistema possuem uma representação própria (modelo mental) de como realizar as tarefas que lhes interessam ou são solicitadas. Numa perspectiva de concepção de sistemas, essa representação é o alvo da análise da tarefa e não a tarefa efetuada num dado momento numa dada situação ([Sebillote 1988]). O objetivo não é avaliar performances durante a realização de uma tarefa, mas sim identificar as representações mentais dos usuários e descrever essas representações independentemente de uma dada situação. A análise da tarefa procura descrever essas representações buscando evidenciar os conceitos efetivamente conhecidos pelos usuários e as características do processo de realização da tarefa pelos usuários aos quais se destina o sistema.

Como afirma Richard ([Richard 1990]), as representações mentais são estruturas cognitivas transitórias, construções que constituem o conjunto das informações consideradas pelo sistema cognitivo durante a realização da tarefa. Richard explica que a ação (ou tarefa) pode ser vista sob um duplo aspecto: 1) a execução da ação, isto é, seu modo de realização e 2) o resultado da ação, isto é, o estado a que ela chega. Ele considera três tipos de informações que são essenciais relativos à ação:

- o resultado da ação que exprime uma mudança de estado e descreve o estado resultante (componente declarativo da ação);
- o(s) procedimento(s) para atingir esse objetivo (componente procedimental da ação);
- os pré-requisitos que definem quais condições devem ser satisfeitas para que a ação possa ser executada.

Da literatura consultada ([Richard 1990, Sebillote 1988a, Cibys 1996]) constata-se que a análise da tarefa a partir de dados colhidos junto aos usuários é um processo que deve procurar evidenciar, entre outros:

- (i) os objetivos que os usuários procuram atingir;
- (ii) sua lógica própria de realização da tarefa (plano de ações ou estrutura de tarefas e sub-tarefas);
- (iii) os objetos e conceitos conhecidos pelos usuários e sua utilização durante a realização da tarefa;
- (iv) os procedimentos que eles utilizam para atingir seus objetivos (métodos);
- (v) as condições necessárias à aplicação desses procedimentos;
- (vi) as possíveis incoerências e incompletude da descrição;
- (vii) as dificuldades e fatores de criticalidade (frequência, importância, etc.).

O resultado da análise da tarefa pode ser aplicado para apoiar as ações do projeto do sistema em pelo menos três momentos:

1. especificação do sistema: a funcionalidade deve ser projetada em função das necessidades dos usuários na realização da tarefa;
2. projeto da interface: a seqüência do diálogo deve ser compatível com a seqüência de tarefas obtidas da decomposição, por exemplo;
3. elaboração de manuais e treinamento: a estrutura da tarefa pode ser usada para organizar manuais e tutoriais visando o treinamento inicial do sistema.

Para levar a bom termo a análise de tarefa, duas recomendações devem ser seguidas [Nanard 1990]:

1. a análise deve ser feita segundo o ponto de vista do usuário e não daquele do projetista (não é o projetista que será o usuário do sistema);
2. a análise da tarefa deve ser realizada seguindo métodos validados por psicólogos.

Uma descrição da tarefa do usuário pode ser obtida através da utilização de alguma técnica de coleta de dados como, por exemplo, entrevista e/ou questionário, e uma análise rigorosa dessa descrição pode ser realizada através de um método tal como a análise de protocolos. A análise de protocolos foi introduzida por Newell e Simon [Newell 1984] para a identificação de processos psicológicos e tem sido utilizada principalmente para identificar os processos em tarefas de resolução de problemas. Ela permite analisar ao mesmo tempo a estrutura de uma tarefa do ponto de vista dos processos que intervêm durante a sua realização e os objetos produtos da tarefa, a partir de transcrições de produções verbais recolhidas por entrevistas.

Um protocolo consiste na descrição da sucessão de atividades efetivamente seguidas por uma pessoa para realizar uma certa tarefa. Ele evidencia a descrição das atividades relativa à condução do processo e os objetos produzidos pelo processo. A análise de protocolos consiste em primeiro decompor a descrição, às vezes incompleta, de protocolos obtidos em segmentos representativos de usuários, em seguida inferir a partir dessa decomposição um modelo dos processos subjacentes pelos quais uma pessoa realiza sua tarefa e, finalmente, verificar a validade do modelo confrontando-o com protocolos que não intervêm diretamente na sua elaboração ou com alguma outra técnica admitida.

2.4 Modelagem da Tarefa

Embora seja evidente a importância de se realizar previamente a análise de tarefa no projeto de um sistema (ou de sua interface), a integração dessa atividade no processo de desenvolvimento de sistemas computacionais não é tão freqüente assim: os profissionais da informática não são capacitados ou treinados para tal, exigindo a presença de profissionais especializados (psicólogos, ergonomistas, etc.), o que acarreta problemas sérios de comunicação, quer entre as pessoas, quer entre os processos envolvidos no projeto. Uma das tendências atuais da pesquisa em ergonomia cognitiva é a procura de um formalismo capaz de descrever uma tarefa segundo um modelo bem elaborado de forma que possa ser integrado em um método de concepção e utilizado não só para a descrição e análise de tarefas mas também para a especificação de interfaces. O objetivo dos vários trabalhos nessa direção é estabelecer uma representação formal da tarefa do usuário que possa ser passível de análise por ferramentas computacionais e que possa ser utilizada em seguida para a obtenção automática ou semi-automática de uma especificação conceitual inicial da interface, integrando heurísticas deduzidas de recomendações ergonômicas.

O problema de análise da tarefa no contexto de concepção de interfaces pode ser visto como análogo ao problema de desenvolvimento de uma base de conhecimento no contexto da IA: aquisição/extração do conhecimento do usuário (*expert*) sobre sua tarefa, representação formal desse conhecimento (base de conhecimento) e análise/verificação da representação (controle). Com essa visão multidisciplinar, diversos trabalhos têm sido desenvolvidos incorporando experiências e resultados das duas áreas, ergonomia e inteligência artificial.

Em ergonomia, o conceito de tarefa é utilizado como um suporte para a descrição da atividade de operadores. A atividade é uma realização da tarefa. O objetivo de uma descrição orientada a tarefa é estabelecer de maneira genérica os diversos conhecimentos postos em prática pelos operadores (usuários) em sua atividade. Trata-se de uma projeção do mundo real sobre um *espaço-conhecimento* permitindo instanciar e/ou verificar as diferentes realizações da tarefa, ou seja, de um modelo da tarefa do usuário. A ciência da computação e em particular a inteligência artificial (IA) contém elementos de solução para o problema de como representar o conhecimento que o usuário tem de sua tarefa de modo a ser utilizado convenientemente na concepção de interfaces homem-computador orientadas para a tarefa. Alguns trabalhos em ergonomia cognitiva tem se apoiado sobre paradigmas e técnicas da IA para resolver o problema da representação e verificação desse conhecimento. Dentre esses trabalhos citamos dois que se destacam pelo pioneirismo nessa área, TKS (*Task Knowledge Structure*) e MAD (*Méthode Analytique de Description des tâches*) que fazem uso de modelos à base de *frames* e do planejamento hierárquico para representar e estruturar tarefas.

Nesses trabalhos, salvo algumas variações, o conceito de tarefa é modelado por uma entidade ou estrutura descrevendo de maneira declarativa os aspectos funcionais e operacionais da tarefa. Os aspectos funcionais correspondem às condições de execução da tarefa (por exemplo, os tipos de parâmetros e resultados) e seus efeitos. Os aspectos operacionais descrevem as ações ou as sub-tarefas a executar em resposta a um contexto funcional dado. As tarefas são descritas em vários níveis de abstração, desde a tarefa-objetivo (nível mais alto) até as tarefas mais simples que podem ser descritas por simples ações elementares (nível mais baixo). Essa decomposição estrutural em vários níveis de abstração representa o plano idealizado pelo usuário para realizar seu trabalho.

O objetivo de uma tal modelagem é poder descrever, de maneira coerente e completa, a tarefa do usuário sob as formas estruturais e funcionais afim de as explorar para a especificação da interface.

2.5 TKS (*Task Knowledge Structure*)

O “modelo” TKS é uma abordagem teórica e metodológica para a representação de tarefas. A tarefa é modelada por uma estrutura TKS (*frame*) que consiste em uma série de atributos associados aos diferentes tipos de conhecimentos utilizados durante sua realização: a estrutura de objetivo/sub-objetivos (sub-tarefas); o plano que representa a execução completa de um objetivo; a estrutura de procedimentos que contém os procedimentos alternativos para executar um sub-objetivo particular; as condições de execução de um objetivo ou de um grupo de procedimentos, isto é, as estratégias; enfim, os objetos e ações, representados segundo uma estrutura taxonomica, associados aos procedimentos. Essas estruturas TKS são ligadas a outras estruturas TKS através das relações definidas pelos atributos formando uma estrutura de controle hierárquico, como ilustra a figura 2.2.

Um método comportando diversas técnicas de coleta de dados (observação direta, construção de listas de ações, de objetos, de procedimentos) foi desenvolvido, a fim de identificar os diferentes conhecimentos, formando as estruturas TKS simples, e de construir as propriedades genéricas, isto é, as estruturas TKS genéricas, estruturas comuns a diversas tarefas simples, dando lugar a um modelo de tarefas independente do contexto de utilização. TKS também propõe um modelo intermediário construído a partir da representação da tarefa e que serve de suporte à construção de um modelo da interação ([Johnson 1988, Wilson 1992, Markopoulos 1997]).

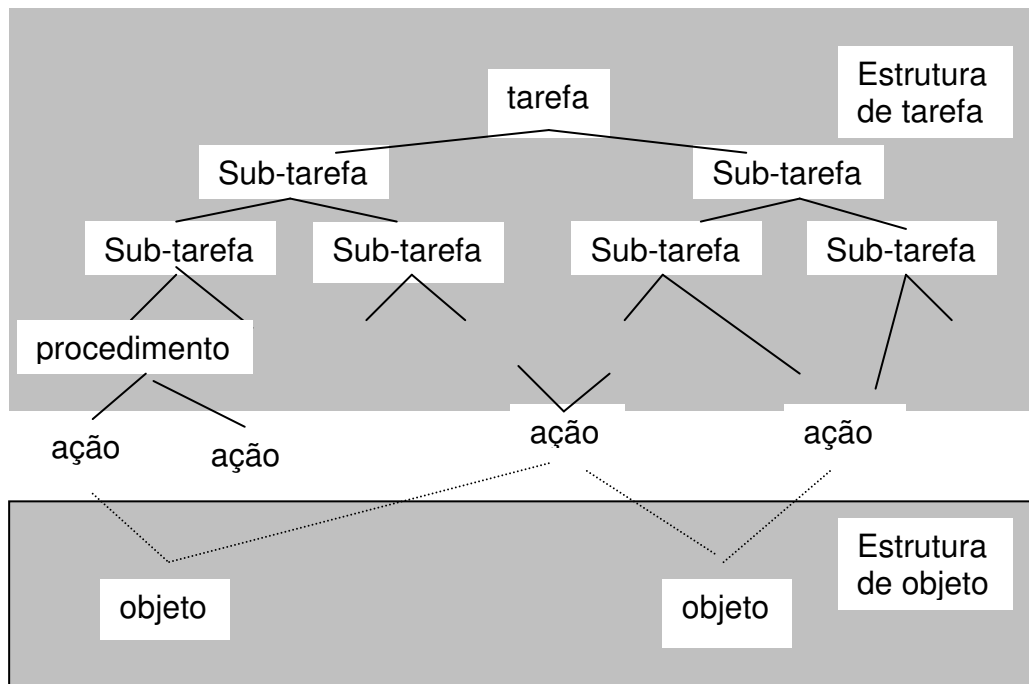


Figura 2.2: Elementos do Modelo TKS [Markopoulos 1997]

2.6 MAD (*Méthode Analytique des Description des tâches*)

MAD é uma linguagem (formalismo) de descrição de tarefas orientada-objeto. A idéia inicial de MAD de dispor de um modelo geral e flexível de descrição de tarefas permitindo a mais longo termo integrar a aplicação de regras ergonômicas no processo de concepção de interfaces ([Hammouche 1993, Gamboa 1997]). A sua implementação através de uma ferramenta computacional (EMAD) objetiva dispor de um meio prático para entrar os dados sobre a tarefa e para assegurar que a tarefa está bem descrita, de maneira coerente e a mais exhaustiva (completa) possível (condições, objetos, estrutura, etc.).

O conceito de tarefa é representado por um objeto genérico (*frame*) chamado *Tarefa*. Um objeto *Tarefa* descreve os aspectos funcionais da

tarefa e comporta os seguintes atributos: nome (identifica a tarefa); estado-inicial (lista de objetos que intervêm como argumento de entrada da tarefa); estado-final (lista de objetos que intervêm como argumento de saída da tarefa e que correspondem a objetos modificados ou criados pela tarefa); objetivo (sub-conjunto do estado final, explicitando o objetivo procurado pela execução da tarefa); pré-condições e pós-condições (conjunto de predicados, exprimindo as restrições sobre o estado-inicial e o estado-final, que devem ser verificadas antes e depois da execução da tarefa, respectivamente); atributos (descrevem as características particulares de certas tarefas) ([Scapin 1987, Sebillote 1994]).

Para definir os aspectos operacionais da tarefa, MAD faz uso de dois conceitos: *tarefa composta* e *tarefa elementar*, que são representados pelas especializações seguintes do *frame Tarefa*:

- o *frame Tarefa-Elementar* comporta, além dos atributos do *frame Tarefa*, o atributo *corpo*, cujo valor é um objeto do tipo *Ação* (por exemplo, as sub-tarefas da tarefa-exemplo, Ativar Navegador e Visualizar Navegador ilustradas no apêndice A);
- o *frame Tarefa-Composta* comporta, além dos atributos do *frame* mãe, o atributo *corpo*, cujo valor é um objeto do tipo *Estrutura*, que permite descrever a decomposição da tarefa em sub-tarefas (por exemplo, a sub-tarefa da tarefa-exemplo, Alternativas de Utilização ilustrada no apêndice A).

O objeto *Estrutura* comporta o atributo *construtor* cujo valor indica o carácter seqüencial, paralelo, iterativo ou alternativo das sub-tarefas, e um atributo *argumentos* constituído de uma lista de sub-tarefas. Uma representação MAD da tarefa consiste em uma coleção de objetos *Tarefa* relacionados com outros objetos *Tarefa* pela relação de decomposição hierárquica em sub-tarefas definida através do atributo *corpo*.

2.7 Discussão

Na perspectiva de concepção de interface homem-computador, o objetivo da análise da tarefa não é avaliar o desempenho de execução de uma tarefa, mas descrever precisamente uma tarefa independentemente de situações particulares. Para isso, os trabalhos em ergonomia cognitiva propõem e fazem uso da noção de representação mental que o usuário tem de sua tarefa. A descrição de uma tarefa obtida da representação mental de um sujeito é diferente de uma descrição tirada da observação em situação real de trabalho. Não se trata de uma representação detalhada e histórica do que se passou com o sujeito quando da execução da tarefa e sim de sua “racionalização”, do que ele abstraiu de detalhe de sua atividade e que ele poderia ensinar ou aconselhar alguém que fosse executar a tarefa pela primeira vez. Do ponto de vista da IA, trata-se de um “conhecimento compilado”. Do ponto de vista ergonômico, trata-se da “lógica do usuário” e conhecer a lógica dos usuários potenciais de uma aplicação futura torna possível relacionar os objetivos e características da tarefa com as características da interface proposta ([Sebillote 1991]).

Modelar a tarefa segundo um modelo do tipo hierárquico não é meramente a escolha de uma técnica. Para tarefas do tipo tarefa de escritório ou administrativa, a representação mental dos sujeitos acerca da planificação de suas ações se constata ser do tipo hierárquico ([Sebillote 1988 Sebillote 1991]). Além disso, é importante distinguir a elaboração de um plano de ações da realização desse plano. O plano de ação ou de condução da atividade em seu conjunto é elaborado como um guia para a realização da tarefa e é geralmente expresso a níveis abstratos e planejado de maneira hierárquica. A realização do plano poderá ser feita de maneira oportunista. Em geral, trata-se de tarefas complexas para as quais tendo construído um plano geral, os sujeitos podem referenciá-lo no curso da execução da tarefa: a tarefa ela mesma é executada de outra forma.

TKS e MAD são tentativas de integrar a descrição da tarefa e a análise dessa descrição no processo de concepção de interfaces. Eles se baseiam na possibilidade de extrair a representação mental que um usuário faz de sua tarefa, através de um método de coleta de dados, e de descrever essa representação através da modelagem de tarefas utilizando um formalismo adaptado à tarefa. Eles propõem portanto uma metodologia de aquisição do conhecimento sobre a tarefa e sua descrição que leva em conta os objetivos visados pelos usuários quando da execução das tarefas.

Mas recentemente, Ben Shneiderman ([Shneiderman 1998]) introduziu o modelo OAI (*Object-Action Interface Model*). Nesse modelo, o projeto inicia com a modelagem da tarefa, que inclui o universo dos objetos com os quais o usuário trabalha para realizar suas intenções e as ações que ele aplica a esses objetos. Da modelagem da tarefa se passa em seguida para uma modelagem da interface, relacionando os elementos da tarefa com os elementos da interface ([Hix 1993]).

2.8 Conclusão

Nesse capítulo apresentamos inicialmente alguns resultados da Teoria da Ação que confirmam a nossa primeira hipótese sobre a necessidade fundamental de se levar em conta a tarefa no processo de concepção de interface homem-computador, afim de se construir sistemas computacionais com alto grau de usabilidade. Além disso, apresentamos os requisitos necessários para se levar a bom termo a análise da tarefa e as tendências atuais na procura de um formalismo capaz de descrever uma tarefa segundo um modelo que possa ser integrado em um método de concepção de interfaces.

Capítulo 3 - MAD e a Análise da Tarefa

3.1 Introdução

Conforme dito na seção 1.7, a metodologia que adotamos nesse trabalho para confirmar a nossa hipótese central sobre a adequação de TAOS para a modelagem da tarefa é a seguinte:

- 1- mostrar que MAD atende aos requisitos mínimos exigidos para a descrição e a análise da tarefa;
- 2- mostrar que TAOS também atende aos requisitos mínimos exigidos para a descrição e a análise da tarefa;
- 3- definir um caso representativo de uma tarefa usual realizada através de uma interação com um sistema computacional (tarefa-exemplo), descrevê-la utilizando os dois formalismos MAD e TAOS e comparar as descrições obtidas.

A adoção de MAD como modelo de comparação e não TKS, por exemplo, deve-se a dois fatores principais: MAD é um modelo validado por experiências levadas a efeito em várias situações de trabalho, desde tarefas de escritório até tarefas mais complexas como, por exemplo, o controle aéreo ([Sebillote 1994a]), e pela disponibilidade de amplo material sobre o assunto (artigos e relatórios de pesquisa de trabalhos de Scapin, Sebillote, Johnson entre outros) (ver referências bibliográficas e bibliografia).

Assim, vamos introduzir inicialmente nesse capítulo uma descrição detalhada de MAD, exemplificando com os elementos do caso, e a seguir faremos uma análise da adequação de MAD aos objetivos propostos (satisfação dos requisitos). Em seguida, nos capítulos 4 e 5, descreveremos TAOS e a representação TAOS do caso e faremos uma análise comparativa com MAD, respectivamente.

3.2 Metodologia MAD

MAD não é apenas um método de análise de tarefa baseado em um modelo da tarefa mas também uma metodologia de análise da tarefa numa perspectiva de concepção de interface para um sistema computacional. MAD define uma metodologia em três passos, *coleta de dados*, *descrição da tarefa* e *validação da descrição* para análise da tarefa do usuário e cujos métodos são os seguintes:

- para a *coleta de dados*, entrevistas semi-dirigidas e uma análise sistemática do *protocolo de base* produzido a partir das entrevistas, produzindo um *protocolo derivado*;
- para a *descrição de tarefas*, modelagem da tarefa utilizando um formalismo orientado a tarefa e adaptado aos objetivos da descrição (formalismo MAD).
- Para a *validação da descrição*, submissão da descrição ou modelo aos usuários e/ou confrontar o modelo com o que fazem os usuários em situação de trabalho.

As entrevistas devem ser um misto de espontâneas e dirigidas (semi-dirigidas), orientadas para se obter uma descrição do tipo hierárquico. Sebillote em [Sebillote 1991] propõe uma técnica desenvolvida por Graesser

e outros ([Graesser 1980]) conhecida como técnica do “porque e como”. Essas questões (porque e como) permitem identificar quais ações são objetivos e quais ações são modos de realização desses objetivos e assim obter a estrutura da tarefa em sub-tarefas. Cada entrevista deve ser registrada e retranscrita por escrito (*protocolo de base*).

A partir do protocolo de base, uma análise sistemática desse documento deve produzir um outro documento (protocolo derivado) compreendendo os seguintes dados e que constituirá uma primeira descrição (textual) da tarefa:

- os nomes das tarefas e sub-tarefas e suas definições;
- os objetivos que deseja atingir o sujeito, quando ele faz referência a uma tarefa, mesmo se ele não a nomeia;
- a seqüência de ações ou método que ele utiliza para realizar um objetivo ou uma tarefa ou sub-tarefa;
- todas as expressões reveladoras de um elemento do plano (objetivo ou sub-objetivo) e de uma dependência entre eles e todas as expressões permitindo compreender um plano subjacente ou seguir uma seqüência de ações;
- tudo o que parece ter um interesse para a descrição da tarefa, como, por exemplo, as informações (objetos, conceitos, etc.) e as condições envolvidas e necessárias para a sua realização;
- todos os casos de exceção.

Após a obtenção do protocolo derivado, deve-se obter uma descrição em termos do formalismo definido por MAD. Essa descrição pode ser feita à mão ou, como é o desejado, com a ajuda de uma ferramenta computacional que implemente o modelo de tarefa definido por MAD permitindo uma construção incremental e uma análise concomitante da completude e da coerência da descrição.

Dos modelos ou descrições individuais deve-se obter um modelo geral de descrição da tarefa, que leve em conta os elementos obrigatórios mencionados por todos e elementos alternativos ou facultativos à realização da tarefa por usuário diferentes. O modelo ou descrição geral obtido deve poder levar em conta todas as descrições e deve em seguida ser validado.

3.3 Tarefa-Exemplo: Navegar na Web

Navegar na web (*world wide web*) para obter informação é uma das tarefas mais freqüentes hoje realizadas com o auxílio de um sistema computacional. Esse fato relevante, além da existência e disponibilidade de usuários de um *navegador* (sistema computacional que permite a realização da tarefa) no próprio grupo de pesquisa em interface homem-computador do CCT (Centro de Ciências e Tecnologia) da UFPB (Universidade Federal da Paraíba), nos levou à escolha dessa tarefa como a tarefa-exemplo (o caso) a ser utilizado nesse trabalho.

Ao cabo de inúmeras entrevistas (reuniões) com os usuários da *web* escolhidos (elementos do grupo), produzimos uma descrição textual inicial da tarefa (protocolo derivado) da qual apresentamos um extrato a seguir:

- a tarefa principal mencionada foi navegar na web (ou mais freqüentemente navegar na rede). Nessa tarefa, um usuário utiliza um navegador como ferramenta para localizar, recuperar, visualizar e enviar informações através da *internet*;
- o objetivo principal verbalizado pelos usuários pode ser sintetizado pela frase: “a busca de informação”;
- os objetos, entidades ou conceitos envolvidos na realização da tarefa (antes, durante e após) foram descritos como sendo: usuário, navegador, tela, informação, arquivo, endereço, etc.;

- as condições de realização da tarefa: ícone do navegador visível na tela, navegador disponível na tela, usuário disponível para navegar na *web*, informação procurada existente e disponível, etc;
- o plano para atingir o objetivo verbalizado pelos usuários pode ser descrito da seguinte maneira:

ativar o navegador, visualizar o navegador na tela e a seguir, executar uma das seguintes tarefas: escolher uma opção de navegação, solicitar ajuda ou finalizar tarefa. A decomposição até o terceiro nível da tarefa navegar na web é a seguinte:

- tarefa principal: Navegar na Web.
Objetivo: a busca de informação.

Essa tarefa é subdividida em cinco sub-tarefas:

{Ativar o Navegador, Visualizar o Navegador, Escolher Opção, Solicitar Ajuda e Finalizar Tarefa}

As sub-tarefas Ativar Navegador e Visualizar o Navegador são executadas de maneira seqüencial.

Ativar Navegador

Objetivo: o usuário seleciona o navegador na tela, através de duplo clique sobre o ícone do navegador.

Visualizar Navegador

Objetivo: o usuário vê o navegador na tela.

Após a visualização do navegador na tela, uma das três outras sub-tarefas é realizada: Escolher Opção, Solicitar Ajuda e Finalizar Tarefa. Essas sub-tarefas e seus objetivos são descritos como:

Escolher Opção

Objetivo: o usuário descreve o endereço que precisa ou manipula um arquivo ou edita um arquivo ou carrega determinada página no navegador.

Solicitar Ajuda

Objetivo: o usuário procura informação de ajuda.

Finalizar Tarefa

Objetivo: o usuário termina de utilizar o navegador.

A sub-tarefa Escolher Opção é por sua vez subdividida em 4 sub-tarefas:

{Descrever o Endereço no Navegador, Manipular o Arquivo, Editar Arquivo, Ir Para determinada página no Navegador}

·
·
·

Como se percebe, a descrição completa da tarefa é um texto longo e difícil de analisar tanto do ponto de vista da sua coerência como da sua completude.

De posse dessas informações colhidas após uma série de reuniões com os usuários escolhidos, passamos à fase de representar essas informações no formalismo MAD, conforme descrito a seguir.

3.4 Formalismo MAD

Os conceitos “chaves” do formalismo MAD são *tarefa*, *estrutura* e *ação*. A conceito de tarefa é descrito por um objeto genérico (*frame*) chamado *Tarefa*. Um objeto desse tipo descreve os aspectos funcionais de

uma tarefa e comporta as seguintes características (atributos): nome (identifica a tarefa); estado-inicial (lista de objetos que intervêm como argumento de entrada da tarefa com seus respectivos tipos); estado-final (lista de objetos que intervêm como argumento de saída da tarefa e que correspondem a objetos modificados ou criados pela tarefa com seus respectivos tipos); objetivo (sub-conjunto do estado final explicitando o objetivo procurado pela execução da tarefa); pré-condições e pós-condições (conjunto de predicados exprimindo as restrições sobre o estado-inicial e sobre o estado-final que devem ser verificadas antes e depois da execução da tarefa, respectivamente). O *frame Tarefa* e seus respectivos pares <atributo, valor>:

Tarefa : nome da tarefa
 estado-inicial = lista de pares objeto: Tipo
 estado-final = lista de pares objeto: Tipo
 objetivo = lista de pares objeto: Tipo
 pré-condições = lista de condições
 pós-condições = lista de condições
 corpo = indefinido

Os aspectos operacionais da tarefa são descritos a partir de dois objetos especializações do objeto *Tarefa*, *Tarefa_Composta* e *Tarefa_Elementar*, definidos como segue:

Tarefa_Composta : nome da tarefa
 estado-inicial = lista de pares objeto: Tipo
 estado-final = lista de pares objeto: Tipo
 objetivo = lista de pares objeto: Tipo
 pré-condições = lista de condições
 pós-condições = lista de condições
 corpo = estrutura: Estrutura

Tarefa_Elementar : nome da tarefa
 estado-inicial = lista de pares objeto: Tipo
 estado-final = lista de pares objeto: Tipo
 objetivo = lista de pares objeto: Tipo
 pré-condições = lista de condições
 pós-condições = lista de condições
 corpo = ação: Ação

O objeto *Tarefa_Composta* é uma especialização do objeto *Tarefa* e define o valor do atributo *corpo* como sendo um objeto do tipo *Estrutura* que permite descrever a decomposição da tarefa em sub-tarefas.

O objeto *Tarefa_Elementar* é uma especialização do objeto *Tarefa* e define o valor do atributo *corpo* como sendo um objeto do tipo *Ação*; ou seja, ele descreve uma ação simples (ação física ou procedimento computacional).

Um objeto do tipo *Ação* comporta um atributo *função*, cujo valor é uma função (do usuário ou do sistema) e um objeto do tipo *Estrutura* comporta, além do atributo *construtor*, cujo valor (SEQ, PAR, ALT ou IT) indica o caráter seqüencial, paralelo, alternativo ou iterativo das sub-tarefas, o atributo *sub-tarefas* cujo valor é uma lista de objetos dos tipos *Tarefa_Composta* e/ou *Tarefa_Elementar* que representam as sub-tarefas:

Estrutura : nome da estrutura
 construtor = SEQ / PAR / ALT / IT
 sub-tarefas = lista de sub-tarefas

Ação : nome da ação
 função = nome e atributos da função

Uma representação MAD da tarefa consiste em uma coleção de objetos do tipo *Tarefa_Composta* relacionados com outros objetos dos tipos *Tarefa_Composta* e *Tarefa_Elementar* ou *Ação* pelas relações de decomposição hierárquica em sub-tarefas definidas através dos objetos *Estrutura*, e com os objetos do tipo *Objeto* ou suas especializações que representam os objetos e conceitos envolvidos na tarefa.

3.5 Representação MAD da Tarefa-Exemplo

A tarefa Navegar na Web é uma tarefa composta e é descrita em MAD pelo objeto *Navegar_na_Web* do tipo *Tarefa_Composta* abaixo, onde o par nome:Objeto indica o nome de um objeto do tipo *Objeto* ou uma especialização de *Objeto*:

```

Tarefa_Composta : Navegar_na_Web
  estado-inicial = {navegador:Navegador, usuário:Usuário,
                    tela:Tela}
  estado-final   = {navegador:Navegador, usuário:Usuário,
                    informação:Informação }
  objetivo       = {informação:Informação}
  pré-condições = {visível_ícone(navegador, tela),
                    disponível(usuario)}
  pós-condições = {obtido(informação)}
  corpo         = plano:Estrutura

```

O plano geral de navegar na web é descrito pelo seguinte objeto *plano* do tipo *Estrutura*:

```

Estrutura : plano
  construtor = SEQ
  sub-tarefas = {Ativar_Navegador:Tarefa,
                 Visualizar_Navegador:Tarefa,
                 Alternativas_de_Utilização:Tarefa}

```

onde as sub-tarefas *Ativar_Navegador*, *Visualizar_Navegador* e *Alternativas_de_Utilização* são realizadas em sequência nessa ordem. As sub-tarefas *Ativar_Navegador* e *Visualizar_Navegador* são tarefas simples e são representadas pelos seguintes objetos:

```

Tarefa_Elementar : Ativar_Navegador
  estado-inicial = {navegador:Navegador, usuário:Usuário,
                    tela:Tela, mouse:Mouse }
  estado-final   = {navegador:Navegador, tela:Tela}
  objetivo       = {navegador:Navegador, tela:Tela}
  pré-condições = {visível_ícone(navegador, tela),
                    disponível (usuário)}
  pós-condições = {disponível(navegador, tela)}
  corpo         = ativar_navegador:Ação

```

A ação *ativar_navegador* é representada pelo seguinte objeto do tipo *Ação*:

Ação : *ativar_navegador*
 função = clicar_ícone(navegador, tela)

Tarefa_Elementar : *Visualizar_Navegador*
 estado-inicial = {navegador:Navegador, usuário:Usuário,
 tela:Tela}
 estado-final = {navegador:Navegador, usuário:Usuário,
 tela:Tela}
 objetivo = navegador:Navegador, usuário:Usuário,
 tela:Tela}
 pré-condições = {disponível(navegador, tela),
 disponível (usuário)}
 pós-condições = {visto(navegador, usuário, tela)}
 corpo = visualizar_navegador:Ação

A ação *visualizar_navegador* é representada pelo seguinte objeto do tipo *Ação*:

Ação : *visualizar_navegador*
 função = olhar(navegador,tela)

A sub-tarefa *Alternativas_de_Utilização* define as alternativas que tem o usuário para utilizar o Navegador em seu trabalho depois de visualizado na tela, conforme o protocolo derivado: escolher uma opção de navegação, solicitar ajuda e finalizar tarefa:

Tarefa_Composta : *Alternativas_de_Utilização*
 estado-inicial = {navegador:Navegador, usuário:Usuário,
 tela:Tela}
 estado-final = {navegador:Navegador, usuário:Usuário,
 informação:Informação}
 objetivo = {informação:Informação}
 pré-condições = {visto(navegador, usuário, tela),
 disponível((navegador, tela))}
 pós-condições = {obtido(informação)}
 corpo = alternativas:Estrutura

As alternativas de utilização são descritas pelo seguinte objeto alternativas do tipo *Estrutura*:

```

Estrutura : alternativas
  construtor = ALT
  sub-tarefas = {Escolher_Opção:Tarefa,
                 Solicitar_Ajuda:Tarefa,
                 Finalizar_Tarefa:Tarefa}

```

A figura a seguir mostra graficamente a decomposição até o 4º nível da tarefa Navegar_na_Web:

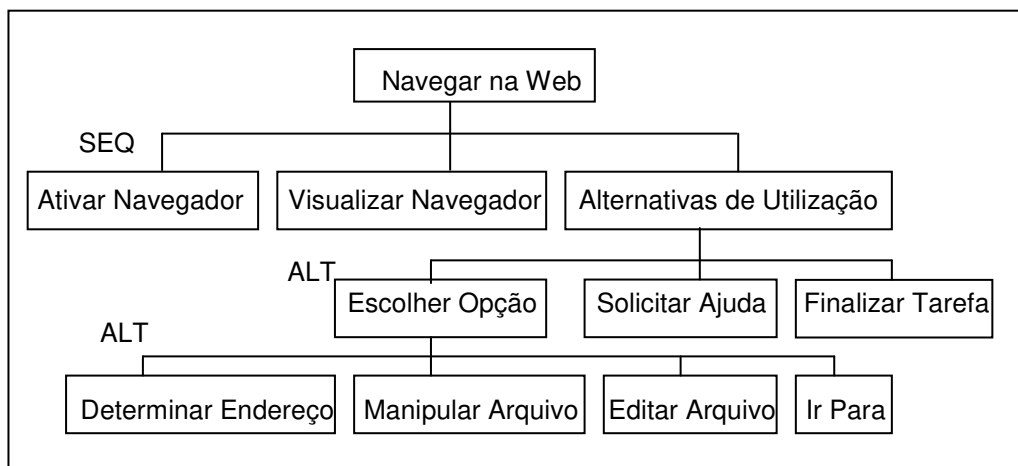


Figura 3.1: Decomposição Hierárquica da Tarefa Navegar na Web

A visualização gráfica completa da tarefa-exemplo Navegar na Web em MAD encontra-se no apêndice A.

3.6 Representação Orientada-Objeto de MAD

A representação da base de objetos de uma tarefa como descrita por MAD é definida segundo um modelo a três níveis: *meta*, *classe* e *instância*:

- o conceito de tarefa é representado no mais alto nível de abstração por um objeto do *nível meta*, a meta-classe *Sh-tarefa*, que fornece uma descrição genérica da arquitetura de todo objeto do tipo tarefa. Uma tarefa, quaisquer que sejam suas especificidades, se caracteriza por uma lista de atributos descrevendo seu objetos na entrada, seus objetos na saída, seu objetivo, etc.. No entanto, tanto o número quanto a natureza desses atributos diferem de acordo com as tarefas: são dependentes de cada tarefa em particular.
- o *nível classe* define uma tarefa específica descrita por um esquema de classe distinto, um objeto do tipo tarefa (classe *Tarefa* ou uma especialização), instancia da meta-classe *Sh_Tarefa* descrevendo a lista de entradas, a lista de saídas, a lista de sub-tarefas, os tipos (classes) dos diferentes objetos manipulados, etc., específicos à essa tarefa. Por exemplo, a tarefa MAD Navegar na Web seria representada pela classe *Navegar_na_Web* descrita abaixo:

```

Classe : Navegar_na_Web
Instancia-de      = Sh_Tarefa
especialização-de = Tarefa_Composta
estado-inicial    = {navegador:Navegador, usuário:Usuário,
                    tela:Tela}
estado-final      = {navegador:Navegador, usuário:Usuário,
                    informação:Informação }
objetivo          = {informação:Informação}
pré-condições    = {visível_ícone(navegador, tela),
                    disponível(usuario)}
pós-condições    = {obtido(informação)}
corpo             = plano:Estrutura

```

- o *nível instancia* corresponde a uma realização particular da tarefa, instanciando os objetos descrevendo os diferentes atributos da tarefa. Por exemplo:

Instancia : Navegar_na_Web#1
 Instância-de = Navegar_na_Web
 estado-inicial = {netscape:Navegador, Joao:Usuário, tela#1:Tela}
 estado-final = {netscape:Navegador, Joao:Usuário, informação:Informação }
 objetivo = {informação:Informação}
 pré-condições = {visível_ícone(netscape, tela#1), disponível(Joao)}
 pós-condições = {obtido(informação)}
 corpo = plano:Estrutura

A figura 3.2 abaixo, ilustra esse modelo em três níveis:

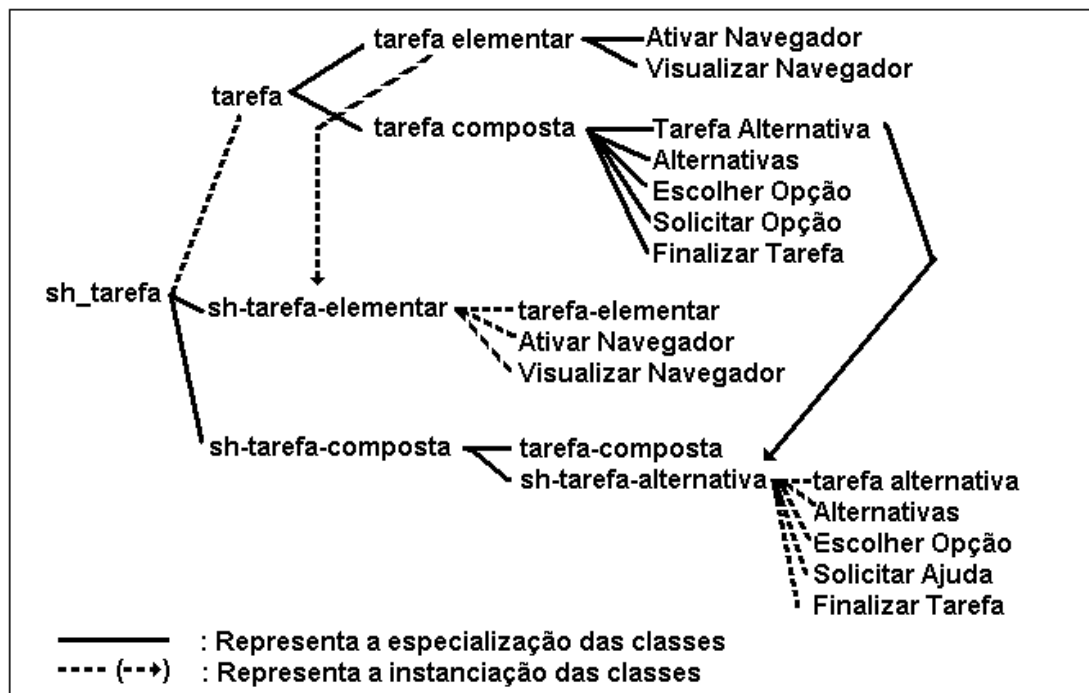


Figura 3.2: Elementos do Modelo MAD

Uma ferramenta computacional gráfica (EMAD) de ajuda à aquisição de conhecimento, fazendo a ponte entre o método de coleta de dados provenientes do usuário e sua representação segundo o modelo MAD, foi construída ([Scapin 1988, Pierret 1989]) em cima de uma plataforma a base de objetos (linguagem Shirka). Ela consiste basicamente de três editores: o

editor de árvore, que permite o usuário explicitar interativamente e graficamente a decomposição estrutural de uma tarefa em sub-tarefas, o editor de tarefa, que permite a consulta e a modificação das informações caracterizando a tarefa associada a cada nó da árvore e o editor de objeto que permite a consulta e a modificação dos objetos referenciados em uma tarefa. Para descrever uma tarefa, o usuário pode alternar uma abordagem descendente com uma abordagem ascendente, permitindo (i) a construção incremental de tarefas mais complexas a partir de componentes existentes, (ii) combinar ao mesmo tempo tarefas já definidas na base e novas sub-tarefas específicas e (iii) definir uma nova tarefa pela modificação de uma tarefa já definida na base. A coerência e a completude da descrição são verificadas *au fur et à mesure*¹ da construção das descrições.

3.7 Satisfação dos Requisitos

A idéia por trás de MAD é a de dispor de um formalismo geral de descrição de tarefas com vistas a obtenção de um conhecimento suficiente e validado sobre a tarefa (representação que dela faz o usuário) e integrá-lo no processo de concepção de interfaces. A descrição da tarefa segundo MAD já é uma forma de análise da tarefa além de se constituir em um suporte conceitual e computacional para a integração de princípios ergonômicos no processo de concepção. Esse segundo ponto sobre integração de princípios ergonômicos não está dentro do contexto desse trabalho.

Do ponto de vista de análise da tarefa, MAD preenche os requisitos mínimos para tal, quais sejam (seção 2.3) evidenciar:

¹ ao mesmo tempo

- os objetivos da tarefa

Os objetivos de uma tarefa se referem à uma particular situação ou estado do mundo que se deseja atingir com sua execução. MAD define a realização de uma tarefa como uma transformação de um estado inicial (lista de objetos valor do atributo estado-inicial) em um estado objetivo particular (lista de objetos valor do atributo objetivo). Cada objeto tarefa contém atributos para descrever a situação inicial antes da execução da tarefa e o objetivo ou situação final após a sua realização.

- a lógica própria de realização da tarefa

Isso significa a explicitação do plano de ação do usuário para atingir os objetivos a que se propõe. MAD utiliza o modelo de planejamento hierárquico usado em IA e descreve a tarefa em uma hierarquia de níveis de abstração. Cada nível de representação descreve a tarefa em mais ou menos detalhes, desde o mais alto (simplificação do plano) até o mais baixo (plano detalhado). Essa decomposição em sub-tarefas é representada em MAD através do atributo corpo do objeto ou frame Tarefa, cujo valor é o par composto da lista de sub-tarefas e a maneira de agenciamento dessas sub-tarefas (seqüencial, paralela, etc.).

- os objetos envolvidos

Isso diz respeito a quais os objetos ou informações são utilizados e quais são produzidos em cada etapa. MAD utiliza os atributos estado-inicial e estado-final para tal e cujos valores são os objetos que definem o estado do mundo antes e depois da realização da tarefa ou sub-tarefa, respectivamente. Os objetos são representados como objetos ou frames do tipo Objeto (ou sua especialização) e seus atributos são introduzidos segundo a necessidade.

- os procedimentos utilizados

Os procedimentos ou métodos utilizados para atingir os objetivos dizem respeito às diferentes maneiras ou estratégias de se realizar a tarefa. Em MAD isso pode ser explicitado através do construtor ALT que permite representar as diferentes maneiras de decomposição da tarefa.

- as condições necessárias para a realização da tarefa

Essas condições dizem respeito às restrições sobre o estado do mundo e que devem ser necessariamente verificadas tanto para a tarefa poder ser iniciada como após a sua execução. MAD permite a explicitação dessas condições através dos atributos pré-condição e pós-condição do objeto Tarefa e cujos valores são predicados sobre os objetos referenciados pelos atributos estado-inicial e estado-final, respectivamente.

- as incoerências e incompletude da descrição

MAD conduz à exploração sistemática e exaustiva de todos os elementos do formalismo. O modelo de representação a base de objetos permite a construção de ferramentas computacionais que suportam a verificação automática da coerência e da completude da descrição da tarefa do ponto de vista de seus atributos (número, valor e tipo) e restrições (condições).

- fatores de criticalidade

A descrição da tarefa em MAD permite que possa ser efetuado um levantamento tanto em termos dos objetos envolvidos e que dizem respeito à sua necessidade e ordem em que se tornam disponíveis, assim como em termos das tarefas e sub-tarefas e que dizem respeito à sua freqüência e modo de utilização.

3.8 Algumas Limitações do MAD

Embora satisfaça os requisitos para a análise de tarefas, como mostrado acima, MAD apresenta alguns inconvenientes na sua formulação para tal. Em primeiro lugar, o modo de descrição da maneira como as sub-tarefas de uma dada tarefa devem ser realizadas é muito rígido: o valor do atributo corpo é uma lista contendo o construtor e a lista de sub-tarefas que ele agencia. Essa rigidez é claramente evidenciada na maneira como é representada a decomposição em sub-tarefas da tarefa-exemplo Navegar na Web. Segundo o protocolo derivado a tarefa é decomposta nas sub-tarefas: Ativar Navegador, Visualizar Navegador, Escolher Opção, Solicitar Ajuda e Finalizar Tarefa, sendo as tarefas Ativar Navegador e Visualizar Navegador executadas em seqüência, enquanto que as tarefas Escolher Opção, Solicitar Ajuda e Finalizar Tarefa são executadas de maneira alternativa. Com o valor que o atributo corpo pode tomar em MAD, não é possível representar diretamente essa maneira ou estratégia de realização da tarefa, exigindo a criação de uma sub-tarefa composta adicional que na tarefa real do usuário não existe (sub-tarefa Alternativas_de_Utilização). Isto gera uma árvore de descrição da tarefa maior, que em conseqüência, faz com que a descrição da tarefa em MAD seja mais complexa do que realmente é.

Um outro aspecto de rigidez do formalismo MAD é o fato de que o estado do mundo (ou situação) antes e depois da realização da tarefa está intrinsecamente ligado à tarefa e não pode ser referenciado a partir de outra tarefa: o valor de cada atributo estado-inicial, estado-final e objetivo é uma lista de objetos (extensional). Desejável seria definir a situação como um objeto autônomo e referenciar-la independentemente da tarefa, como será esclarecido mais adiante.

Finalmente, embora referencie os objetos envolvidos na tarefa, MAD não faz alusão aos aspectos relacionados com esses objetos, como, por

exemplo, ser agente ou instrumento de uma ação (por ex. usuário e navegador) e que competências possui (lista de ações que ele está apto a realizar). A descrição dos objetos envolvidos com a tarefa não faz parte do modelo (o *frame Objeto* não define nenhum atributo especial).

3.9 Conclusão

De acordo com a análise acima e com os relatos das experiências levadas a cabo em situações complexas ([Sebillote 1987, Sebillote 1988a, Sebillote 1991, Sebillote 1994a]), MAD atende aos propósitos básicos para os quais foi concebido de ferramenta conceitual e formal para a descrição e análise da tarefa, apesar de algumas limitações que não invalidam a sua utilização, o que confirma a nossa segunda hipótese, o formalismo MAD é um elemento de solução para o problema de representação e análise da tarefa no processo de concepção de interfaces homem-computador de sistemas computacionais.

Capítulo 4 – Formalismo TAOS

4.1 Introdução

O problema da escolha de uma representação para o conhecimento sobre um domínio envolve as etapas de identificação do conhecimento e construção de um modelo conceitual do domínio usando uma dada metodologia.

TAOS, através do seu módulo TAME, define uma linguagem e uma metodologia ascendente-descendente de construção do modelo do domínio. Essa linguagem possui uma sintaxe, uma semântica e uma terminologia que atentam para os aspectos dinâmicos presentes no domínio. Isto permite, dentro do contexto de aquisição do conhecimento sobre um domínio, uma verificação preliminar da coerência e da completude da modelagem desse domínio ([Medeiros 1995a, Kessel 1995, Medeiros 1995b, Medeiros 1995c]).

Neste capítulo, será feita uma apresentação dos conceitos do formalismo TAOS e sua utilização como ferramenta de modelagem da tarefa do usuário no processo de concepção de interface homem-computador de um sistema computacional.

4.2 Modelo TAOS

O modelo de representação proposto pelo formalismo TAOS considera que podem existir em um domínio, dois tipos de entidades ou

conceitos: os *conceitos estáticos* (os objetos, os métodos e as situações) e os *conceitos dinâmicos* (os processos, as ações e os planos). 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 figura 4.1 abaixo, apresenta o modelo TAOS de representação do conhecimento de um domínio, distinguindo os conceitos estáticos e dinâmicos. 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.

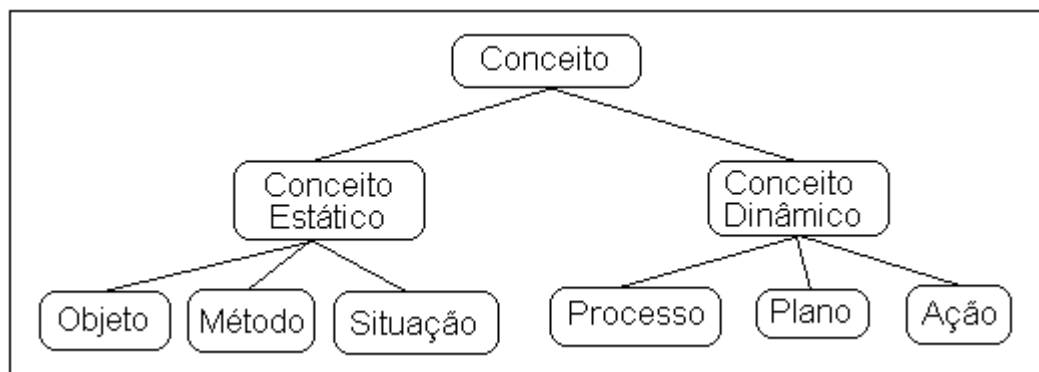


Figura 4.1: Hierarquia de Conceitos de TAOS

A classe de mais alto nível, a classe *Conceito*, define uma entidade (real ou abstrata) do domínio no mais alto grau de abstração (apenas seu nome e descrição):

Classe Conceito

nome	= cadeia de caracteres
descrição	= cadeia de caracteres
<superconceito>	= lista de superclasses
<subconceito>	= lista de subclasses
<atributo-adicional>	= tipo do atributo

Os atributos indicados pelos caracteres “<” e “>” são opcionais e podem ser adicionados na medida da necessidade e dos objetivos da modelagem.

4.3 Conceitos Estáticos

A classe *Conceito_Estatico*, sub-classe da classe *Conceito*, define uma entidade ou um conceito cujos valores não se alteram durante um intervalo de tempo considerável:

Classe Conceito_Estático

nome	= cadeia de caracteres
descrição	= cadeia de caracteres
<superconceito>	= lista de superclasses
<subconceito>	= lista de subclasses
<instante>	= tempo
<localização>	= posição
<atributo-adicional>	= tipo de atributo

A classe *Objeto* é utilizada para definir um objeto ou entidade (um conceito estático) envolvido na execução de uma ação:

Classe Objeto

nome	= cadeia de caracteres
descrição	= cadeia de caracteres
<superconceito>	= lista de superclasses
<subconceito>	= lista de subclasses
<instante>	= tempo
<localização>	= posição
<componentes>	= lista componentes
<atributo-adicional>	= tipo de atributo

Um objeto qualquer envolvido com a execução de uma ação deve ocupar uma posição no espaço em um dado instante (atributos *instante* e *localização*) e deve ser decomponível em suas partes constituintes (atributo

componentes). A consideração desses atributos é, no entanto, opcional, introduzidos apenas se a necessidade assim o exigir. A classe *Objeto* é especializada nas sub-classes *Agente* e *Instrumento*:

Classe Agente

nome	= cadeia de caracteres
descrição	= cadeia de caracteres
<superconceito>	= lista de objetos
<subconceito>	= lista de subclasses
<instante>	= tempo
<localização>	= posição
<componentes>	= lista componentes
competência	= lista de objetos do tipo ação
<interlocutores>	= lista de objetos do tipo agente
<atributo-adicional>	= tipo de atributo

A classe *Agente* define um tipo de objeto habilitado a executar uma ação. O agente realiza ações que ele conhece e para as quais é competente (atributo *competência*). O agente pode eventualmente tomar decisões a partir de informações que ele recebe de outros agentes (atributo *interlocutores*).

Classe Instrumento

nome	= cadeia de caracteres
descrição	= cadeia de caracteres
<superconceito>	= lista de objetos
<subconceito>	= lista de subclasses
<instante>	= tempo
<localização>	= posição
<componentes>	= lista componentes
operador	= lista de objetos do tipo agente
emprego	= lista de objetos do tipo plano/ação
<atributo-adicional>	= tipo de atributo

A classe *Instrumento* define um tipo de objeto (uma ferramenta) que é utilizado pelos agentes para executar suas ações (emprego). Um instrumento deve ser manipulado apenas pelos agentes autorizados a tal e que são indicados na lista que é valor do atributo *operador*.

A classe *Método* define o método ou estratégia para executar um plano:

Classe Método

nome	= cadeia de caracteres
descrição	= cadeia de caracteres
<superconceito>	= lista de superclasses
<subconceito>	= lista de subclasses
<instante>	= tempo
<localização>	= posição
corpo	= expressão
<atributo-adicional>	= tipo de atributo

O valor do atributo *corpo* é uma expressão composta ou simples, que utiliza os operadores SEQ, OR, XOR, AND e SIM para descrever a decomposição do plano em sub-planos/ações segundo a seguinte gramática:

<expressão>	::= <opr> (<lista_expressão>)
<lista_expressão>	::= <ação>,<lista_expressão>
<lista_expressão>	::= <plano>,<lista_expressão>
<lista_expressão>	::= <expressão>,<lista_expressão>
<lista_expressão>	::= <ação>,< expressão_simples >
<lista_expressão>	::= <plano>,<expressão_simples>
<lista_expressão>	::= <expressão>,<expressão_simples>
<expressão_simples>	::= <ação> <plano> <expressão>
<opr>	::= SEQ OR XOR AND SIM

Os operadores SEQ, OR, XOR, AND, SIM permitem estabelecer relações temporais (precedência) e/ou lógicas entre os sub-planos e/ou ações que compõem um plano.

A classe *Situação* define o estado do mundo (objetos e as restrições sobre os objetos) em um determinado instante e as maneiras (planos) de chegar a esse estado :

Classe Situação

nome	= cadeia de caracteres
descrição	= cadeia de caracteres
<superconceito>	= lista de superclasses
<subconceito>	= lista de subclasses
<instante>	= tempo
objetos	= lista de objetos
restrição	= expressão
como-atingir	= lista de objetos do tipo plano/ações
<atributo-adicional>	= tipo de atributo

O valor do atributo *restrição* é uma expressão lógica sobre os objetos referenciados na lista de objetos do atributo *objetos* (utilizando funções, variáveis e conectivos lógicos) e reflete as condições que a situação deve observar. O valor do atributo *como-atingir* é uma lista de planos ou ações que indicam as diversas maneiras de se atingir a situação.

4.4 Conceitos Dinâmicos

Um conceito (ou entidade) é considerado como dinâmico se ele representa uma evolução de uma situação observada dentro de um intervalo de tempo. Essa evolução pode ser ocasionada pela intervenção intencional de um agente ou pela resposta (reação) automática de um artefato. A modelagem dessa evolução pode ser expressa através de diferentes níveis de abstração: a partir de um plano geral (mais alto nível), por desdobramento deste plano (decomposição), chega-se às ações elementares (ações físicas). A classe *Conceito_Dinâmico* representa no mais alto nível esse tipo de entidade ou conceito:

Classe Conceito_Dinâmico

nome	= cadeia de caracteres
descrição	= cadeia de caracteres
<superconceito>	= lista de superclasses
<subconceito>	= lista de subclasses
pré-situação	= objeto do tipo situação
pós-situação	= objeto do tipo situação
<atributo-adicional>	= tipo de atributo

O atributo *pré-situação* define a situação inicial e o atributo *pós-situação* define a situação resultante da execução de um plano ou de uma ação (conceito dinâmico).

A classe *Processo* define um conjunto de situações observadas em instantes diferentes (dispostas em ordem cronológica parcial ou total) e permite registrar a execução de um plano:

Classe Processo

nome	= cadeia de caracteres
descrição	= cadeia de caracteres
<superconceito>	= lista de superclasses
<subconceito>	= lista de subclasses
pré-situação	= objeto do tipo situação
pós-situação	= objeto do tipo situação
situações	= lista de objetos do tipo situação
<atributo-adicional>	= tipo de atributo

Um plano é uma sequência de ações organizada de maneira a se chegar a um objetivo fixado. Ele pode ser decomposto em vários níveis de abstração, desde a sua nomeação (mais alto nível) até as ações elementares, e pode ser realizado de diversas maneiras (métodos):

Classe Plano

nome	= cadeia de caracteres
descrição	= cadeia de caracteres
<superconceito>	= lista de superclasses
<subconceito>	= lista de subclasses
pré-situação	= objeto tipo situação
pós-situação	= objeto tipo situação
ocorrência	= um dos valores: (0,0),(0,1),(0,n),(1,1),(1,n),(while(situação))
ações	= lista de objetos do tipo ação
sub-planos	= lista de objetos do tipo plano
como-realizar	= lista de objetos do tipo método
<atributo-adicional>	= tipo de atributo

O atributo *ocorrência* define o número de vezes que o plano (ou ação) é executado. Os valores possíveis desse atributo são os seguintes:

- (0,0): O plano, sub-plano ou a ação ocorre 0 vezes (NOT)
- (0,1): O plano, sub-plano ou ação ocorre 0 ou 1 vez
- (0,n): O plano, sub-plano ou ação ocorre 0 ou mais vezes
- (1,1): O plano, sub-plano ou ação ocorre 1 vez
- (1,n): O plano, sub-plano ou ação ocorre 1 ou mais vezes
- (*while*(situação)): O plano, sub-plano ou ação continuam ocorrendo enquanto a situação seja válida

Quando a ocorrência é (0,0), (0,1) ou (0,n) o “0” indica que o plano, sub-plano ou a ação é facultativo. Quando a ocorrência é (1,1) ou (1,n) o “1” indica que o plano, sub-plano ou ação tem que ocorrer pelo menos 1 vez.

A classe *Ação* descreve uma ação elementar. Utilizando a nomenclatura de árvores, na estrutura hierárquica de uma plano, as ações são as folhas da árvore e não podem ser mais decompostas:

Classe Ação

nome	= cadeia de caracteres
descrição	= cadeia de caracteres
<superconceito>	= lista de superclasses
<subconceito>	= lista de subclasses
pré-situação	= objeto do tipo situação
pós-situação	= objeto do tipo situação
ocorrência	= um dos valores (0,0),(0,1),(0,n),(1,1),(1,n),(while(situação))
agentes	= lista de objetos do tipo agente
instrumentos	= lista de objetos do tipo instrumento
status	= automática ou não automática
<ação–inversa>	= objeto do tipo ação
<ação–desfazer>	= objeto do tipo ação
<atributo-adicional>	= tipo de atributo

O atributo *status* define se a ação é uma ação automática (realizada por um artefato ou instrumento como resposta ou reação a uma ação de um agente) ou uma ação não automática (realizada por um agente).

4.5 Representação TAOS da Tarefa-Exemplo

Vamos utilizar o formalismo TAOS para representar a tarefa-exemplo Navegar na Web a partir do protocolo derivado descrito na seção 3.3. Como dito na seção anterior (4.4), um conceito (ou entidade) é um conceito dinâmico se ele representa uma evolução de uma situação observada dentro de um intervalo de tempo. Assim, uma tarefa é um conceito dinâmico pois trata-se de uma evolução de uma situação ou estado do mundo inicial para uma situação ou estado objetivo, envolvendo objetos (pessoas, ferramentas, informações, etc.) e a utilização de métodos ou estratégias para atingir o objetivo fixado. Então, em TAOS, o conceito de tarefa pode ser descrito através de uma classe especialização da classe *Plano*. Vamos utilizar, para descrever uma classe especialização de alguma classe pré-definida do modelo TAOS, por questão de homogeneização da apresentação, o mesmo

esquema de descrição usado na seção 3.5 para descrever os *frames* especialização dos *frames* pré-definidos de MAD, ou seja, em primeiro lugar vem o nome da superclasse (Plano) seguido do nome da sub-classe (Navegar_na_Web), e em seguida, a lista de seus atributos:

Plano : Navegar_na_Web
 descrição = "Usuário navega na Web por meio de um navegador a busca de uma informação"
 pré-situação = Situação_Inicial
 pós-situação = Situação_Objetoivo
 ocorrência = (1,n)
 ações = {Ativar_Navegador, Visualizar_Navegador, Finalizar_Tarefa}
 sub-planos = {Escolher_Opção, Solicitar_Ajuda}
 como-realizar = {Método_Navegar_na_Web}

O valor do atributo *pré-situação* é um objeto (classe *Situação_Inicial*, sub-classe da classe *Situação*) onde deve estar descrito os objetos (usuário, navegador) que o usuário precisa para a execução da tarefa (ou plano) e as restrições sobre esses objetos para que a tarefa possa ser executada. Da mesma forma, o valor do atributo *pós-situação* é a classe *Situação_Objetoivo*, sub-classe da classe *Situação*, que deve descrever a situação a ser atingida (o objetivo) pela execução da tarefa. O valor do atributo *ocorrência*, (1,n), indica que o plano pode ser iterado qualquer número de vezes (no mínimo 1 vez). O valor do atributo *ações* é a lista de ações elementares da primeira decomposição do plano e o valor do atributo *sub-planos* é a lista de planos (sub-planos) dessa decomposição. O valor do atributo *como-realizar* é a classe *Método_Navegar_na_Web*, sub-classe da classe *Método*, que deve descrever a maneira de agenciamento das ações e sub-planos.

Situação : Situação_Inicial
 descrição = "Situação ou estado do mundo antes da realização da tarefa"
 objetos = {Usuário, Navegador, Tela}
 restrição = visível_icone(Navegador, Tela) AND disponível(Usuário)
 como-atingir = {}

Situação : Situação_Objetivo
 descrição = "Objetivo a atingir"
 objetos = {Informação}
 restrição = obtido(Informação)
 como-atingir = {Navegar na Web}

De acordo com o protocolo derivado, a tarefa navegar na web é decomposta em 5 sub-tarefas: Ativar_Navegador, Visualizar_Navegador, Escolher_Opção, Solicitar_Ajuda e Finalizar_Tarefa, onde as duas primeiras são executadas em sequência seguido da execução de uma das outras três. Esse agenciamento pode ser descrito através da classe *Método_Navegar_na_Web*, sub-classe da classe *Método*:

Método : Método_Navegar_na_Web
 descrição = "Estratégia para navegar na web"
 corpo = SEQ(Ativar_Navegador, Visualizar_Navegador,
 OR(Escolher_Opção, Solicitar_Ajuda, Finalizar_Tarefa))

Para a realização da tarefa navegar na web o usuário terá que em primeiro lugar ativar o navegador. Esta é uma ação elementar não automática, já que o usuário tem realizá-la através de uma ação física:

Ação : Ativar_Navegador
 descrição = Inicializa Navegador do Usuário
 pré-situação = Situação_Inicial
 pós-situação = Navegador_na_Tela
 ocorrência = (1,1)
 agentes = {Usuário}
 instrumentos = {Navegador}
 status = não_automática

Situação : Navegador_na_Tela
 descrição = "Navegador visível e disponível na tela para utilização pelo usuário"
 objetos = {Navegador, Usuário, Tela}
 restrição = disponível(Navegador, Tela) AND disponível(Usuário)
 como-atingir = {Ativar_Navegador}

Tendo concluído com sucesso a ação *Ativar_Navegador*, o usuário executa a ação (não automática) *Visualizar_Navegador* que é descrita como:

Ação : *Visualizar_Navegador*
 descrição = "Usuário visualiza o navegador na tela"
 pré-situação = *Navegador_na_Tela*
 pós-situação = *Navegador_Visto*
 ocorrência = *while(Navegador_na_Tela)*
 agentes = {Usuário}
 instrumentos = {Navegador}
 status = não_automática

Situação : *Navegador_Visto*
 descrição = "Usuário olha navegador ativado na tela"
 objetos = {Navegador, Usuário, Tela}
 restrição = *disponível(Navegador, Tela) AND disponível(Usuário)*
 como-atingir = {*Visualizar_Navegador*}

O agente da tarefa navegar na web é o usuário e a tarefa que ele é capaz de fazer (competência) é representada pelo plano *Navegar_na_Web*:

Agente : Usuário
 descrição = "Usuário que realiza a tarefa navegar na web"
 competencia = {*Navegar_na_Web*}

O instrumento que o usuário (operador) utiliza para a realização da tarefa *Navegar na Web* é um navegador:

Instrumento : *Navegador*
 descrição = "Navegador utilizado pelo usuário para navegar na web"
 operador = {Usuário}
 emprego = {*Navegar_na_Web*}

A figura a seguir mostra graficamente a decomposição até 3º nível da tarefa *Navegar na Web* representada em TAOS:

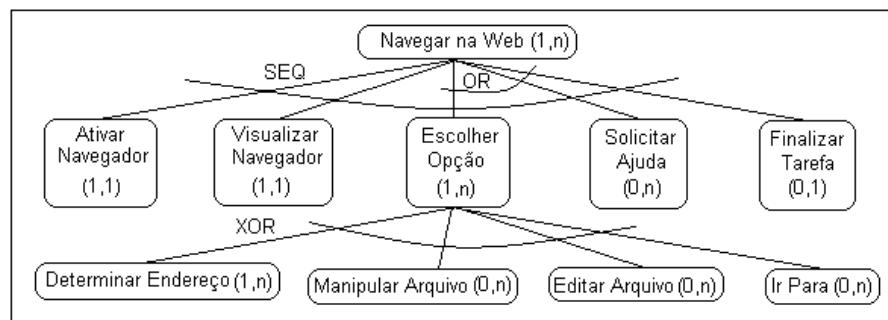


Figura 4.2: Tarefa Navegar na Web em TAOS

A visualização gráfica completa da tarefa-exemplo navegar na web em TAOS encontra-se no apêndice B.

4.6 Conclusão

Neste capítulo foi feita a apresentação dos conceitos do formalismo TAOS. Este formalismo foi concebido com o objetivo de auxiliar na tarefa de aquisição e representação de conhecimento sobre um domínio. Também ilustramos, através da representação da tarefa-exemplo, que TAOS se presta aos objetivos de análise da tarefa no contexto de concepção de interface homem-computador de um sistema computacional, o que fortalece a nossa convicção da veracidade da hipótese central do nosso trabalho.

Capítulo 5 - TAOS e a Análise da Tarefa

5.1 Introdução

No capítulo anterior apresentamos o formalismo TAOS e uma ilustração, através da representação da tarefa-exemplo, de que TAOS é adequado para os objetivos de análise da tarefa. Neste capítulo apresentaremos as evidências suficientes para confirmar a nossa hipótese central da adequação do formalismo TAOS para o processo de análise da tarefa no contexto de concepção de interface homem-computador. Ou seja, mostraremos que TAOS satisfaz os requisitos exigidos para análise da tarefa e faremos uma análise comparativa entre MAD e TAOS usando como referência a tarefa-exemplo. Em seguida, apresentaremos como alguns dos inconvenientes observados durante a utilização de MAD na representação da tarefa-exemplo podem ser evitados com a utilização do TAOS.

5.2 Satisfação dos Requisitos

O formalismo TAOS preenche os requisitos exigidos para análise da tarefa, ou seja, a modelagem da tarefa no formalismo TAOS é capaz de evidenciar :

- os objetivos da tarefa

Os objetivos de uma tarefa se referem a uma particular situação ou estado do mundo que se deseja atingir com a execução da tarefa. Isso pode

ser evidenciado em TAOS através do valor do atributo pós-situação da classe Plano (ou especialização) que define a tarefa-objetivo. Esse valor é um objeto do tipo Situação.

- a lógica própria de realização da tarefa

Isso significa a explicitação do plano de ação do usuário para atingir os objetivos a que se propõe. Em TAOS, essa lógica de realização (decomposição em sub-planos) é descrita através do atributo corpo da classe Método, cujo valor é uma expressão composta ou simples (de planos/ações) que define a maneira de agenciamento (seqüencial, paralela, simultânea, etc.) desses sub-planos (sub-tarefas). A sintaxe da expressão é definida através da gramática descrita na seção 4.3.

- os objetos envolvidos

Isso diz respeito a quais objetos ou informações são utilizados e quais são produzidos em cada etapa. Em TAOS, se utiliza para tal o atributo objetos da classe Situação, cujo valor é uma lista dos objetos que definem o estado do mundo antes ou depois da realização do plano, sub-plano ou ação.

- os procedimentos utilizados

Os procedimentos ou métodos utilizados para atingir os objetivos dizem respeito às diferentes maneiras ou estratégias de se realizar a tarefa. Em TAOS, isso pode ser explicitado através do atributo *como-realizar* da classe *Plano*, cujo valor é uma lista de objetos do tipo *Método* que permite representar as diferentes maneiras de realização da tarefa.

- as condições necessárias para a realização da tarefa

Essas condições dizem respeito às restrições sobre o estado do mundo que devem ser necessariamente verificadas tanto para que a tarefa possa ser iniciada como após a sua execução. TAOS permite a explicitação

dessas condições através do atributo *restrição* da classe *Situação* cujo valor é um predicado sobre os objetos referenciados pelo atributo *objetos* da mesma classe.

- as incoerências e incompletude da descrição

O módulo TAME de TAOS guia a construção do modelo da tarefa (plano) através da utilização de uma metodologia ascendente-descendente, permitindo uma construção incremental do modelo e fazendo a verificação *a priori* da coerência e da completude da descrição.

- fatores de criticalidade

A descrição da tarefa (plano) em TAOS permite que possa ser efetuado um levantamento tanto em termos dos objetos envolvidos e que dizem respeito à sua necessidade e ordem em que tornam-se disponíveis, assim como em termos das tarefas (planos) e sub-tarefas (sub-planos) e que dizem respeito à sua frequência e modo de utilização. Além disso, as classes do modelo TAOS possuem atributos opcionais como *instante* e *localização* que permitem estabelecer o momento e lugar em que a tarefa (plano) é realizada quando for necessário incluir este tipo de informação nas análises de frequência da tarefa modelada. Por outro lado, a classe *Processo* permite que seja efetuada uma simulação da realização da tarefa, ou seja, explicitar a seqüência de situações que são alcançadas durante a realização da tarefa.

5.3 Análise Comparativa MAD x TAOS

Como vimos na seção anterior, TAOS atende aos requisitos exigidos para análise da tarefa, o que demonstra a veracidade da nossa hipótese central. Vamos realizar a seguir uma comparação entre MAD e TAOS

tomando como referência a tarefa-exemplo (o caso) e os requisitos da análise da tarefa:

- os objetivos da tarefa

Em MAD, o objetivo está representado no *frame Tarefa* pelo atributo *objetivo*, cujo valor é uma lista de objetos. Em TAOS, a definição do objetivo envolve duas classes, a classe *Plano* através do atributo *pós-situação* cujo valor é um objeto da classe *Situação* onde então são listados os objetos que definem a situação objetivo:

Tarefa em MAD : nome da tarefa
estado-inicial
estado-final
objetivo = lista de objetos
pré-condições
pós-condições
corpo

Plano em TAOS: nome do plano
pré-situação
pós-situação =
ocorrência
ações
sub-planos
como-realizar

Classe Situação: nome da situação
objetos = lista de objetos
restrição
como-atingir

Na tarefa-exemplo:

objetivos da Tarefa

Tarefa : Navegar_na_Web
estado-inicial = {navegador:Navegador,
usuário:Usuário,tela:Tela}
estado-final = {navegador:Navegador,
usuário:Usuário,
informação:Informação}
objetivo = {informação:Informação}
pré-condições = {visível_ícone(navegador,
tela), disponível(usuário)}
pós-condições = {obtido(informação)}
corpo = plano:Estrutura

Plano : Navegar_na_Web
pré-situação = Situação_Inicial
pós-situação = Situação_Objeto
ocorrência = (1,n)
ações = {Ativar_Navegador,
Visualizar_Navegador,
Finalizar_Tarefa}
sub-planos = {Escolher_Opção,
Solicitar_Ajuda}
como-realizar = {Método_Navegar_na_Web}

Situação : Situação_Objeto
descrição = "Objetivo a atingir"
objetos = {Informação}
restrição = obtido(Informação)
como-atingir = {Navegar na Web}

Como vemos, em MAD, é mais simples representar os objetivos que em TAOS, porém, isso acarreta a limitação referenciada na seção 3.8 qual seja, o estado do mundo (objetivo) é intrinsecamente ligado à descrição da tarefa, o que é evitado em TAOS com a descrição distribuída em duas classes (*Plano* e *Situação*).

- a lógica própria de realização da tarefa

Em MAD, a lógica de realização da tarefa é representada através do atributo *corpo* do objeto ou *frame Tarefa* cujo valor é um objeto do tipo *Estrutura* ou do tipo *Ação*. No caso de ser uma tarefa composta, o objeto *Estrutura* associado contém os atributos *construtor*, cujo valor indica a maneira de agenciar as sub-tarefas (SEQ, ALT, PAR, etc.) e *sub-tarefas* cujo valor é a lista de sub-tarefas em que a tarefa é decomposta. Em TAOS, essa representação envolve o uso da classe *Plano* através dos atributos *ações*, *sub-planos* e *como-realizar* e de classes do tipo *Método* que constituem a lista que é o valor do atributo *como-realizar*. Em cada classe do tipo *Método* se define então, através de uma expressão, uma decomposição ou maneira de agenciamento das sub-tarefas (sub-planos/ações).

Tarefa em MAD : nome da tarefa

estado-inicial
estado-final
objetivo
pré-condições
pós-condições
corpo = Estrutura/Ação

Plano em TAOS: nome do plano

pré-situação
pós-situação
ocorrência
ações
sub-planos
como-realizar =

Estrutura em MAD : nome estrutura

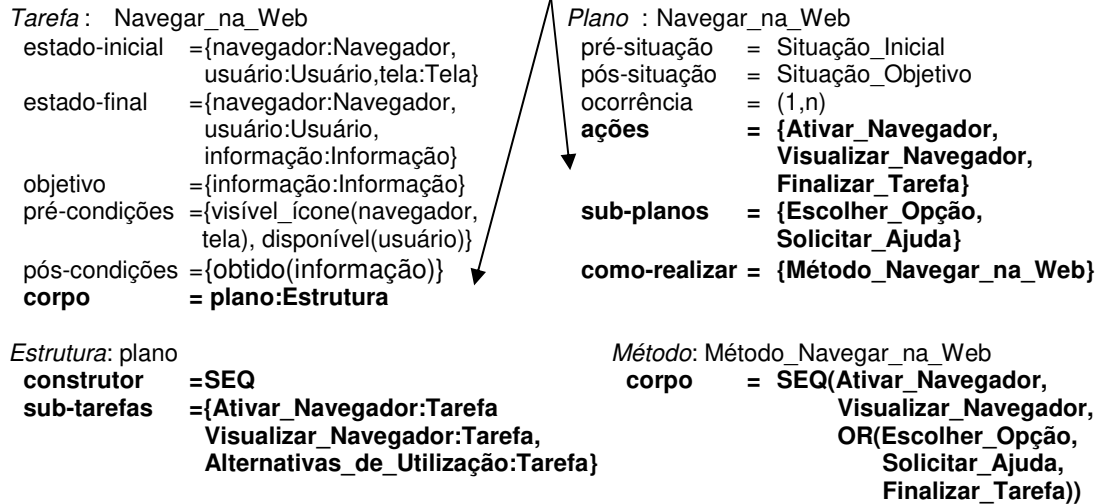
construtor = SEQ/ALT/...
sub-tarefas = lista sub-tarefas

Método em TAOS: nome do método

corpo = expressão

Na tarefa-exemplo:

lógica da realização da tarefa



Em TAOS, os planos, sub-planos e ações (de acordo com o *operador* que os relaciona: OR, XOR, etc.), podem ser agrupados em expressões compostas. Isto evita o inconveniente encontrado em MAD (seção 3.8) ao não ser possível representar diretamente a lógica da realização da tarefa, quando se trata de expressões compostas como a ilustrada acima, exigindo a criação de uma sub-tarefa composta adicional (Alternativas_de_Utilização) que na tarefa real do usuário não existe.

- os objetos envolvidos

Em MAD, os objetos ou informações envolvidos em cada etapa são explicitamente referenciados no próprio objeto ou *frame* que define a tarefa, como valor dos atributos *estado-inicial* e *estado-final*. Em TAOS, a classe que define a tarefa referencia uma outra classe (através dos atributos *pré-situação* e *pós-situação*) do tipo *Situação*, onde então são explicitados os objetos envolvidos.

Tarefa em MAD : nome da tarefa
estado-inicial = lista objetos
estado-final = lista objetos
 objetivo
 pré-condições
 pós-condições
 corpo

Plano em TAOS: nome do plano
pré-situação =
pós-situação =
 ocorrência
 ações
 sub-planos
 como-realizar

Situação em TAOS: nome
 descrição
objetos = lista objetos
 restrição
 como-atingir

Na tarefa-exemplo:

objetos envolvidos

<p><i>Tarefa</i> : Navegar_na_Web estado-inicial = {navegador:Navegador, usuário:Usuário,tela:Tela} estado-final = {navegador:Navegador, usuário:Usuário, informação:Informação} objetivo = {informação:Informação} pré-condições = {visível_ícone(navegador, tela), disponível(usuário)} pós-condições = {obtido(informação)} corpo = plano:Estrutura</p>	<p><i>Plano</i> : Navegar_na_Web pré-situação = Situação_Inicial pós-situação = Situação_Objeto ocorrência = (1,n) ações = {Ativar_Navegador, Visualizar_Navegador, Finalizar_Tarefa} sub-planos = {Escolher_Opção, Solicitar_Ajuda} como-realizar = {Método_Navegar_na_Web}</p>
---	---

Situação: Situação_Inicial
descrição = "Situação ou estado do mundo
antes da realização da tarefa"
objetos = {Usuário, Navegador, Tela}
restrição = visível_ícone(Navegador, Tela)
AND disponível(Usuário)
como-atingir = {}

O modelo TAOS discrimina os objetos envolvidos na realização de uma tarefa, através das sub-classes *Agente* e *Instrumento*. Isto possibilita a descrição de aspectos importantes relacionados com esses objetos, como, por exemplo, ser um objeto habilitado a executar uma ação (ver classe *Agente* na seção B.3 do apêndice B) ou um objeto utilizado por um agente na execução de uma ação (classe *Instrumento*). MAD não fornece como parte do modelo a possibilidade de representação dessa informação.

- os procedimentos utilizados

MAD representa através do construtor ALT (utilizado no valor do atributo *corpo* da tarefa composta) as diferentes maneiras ou estratégias de realização da tarefa. Em TAOS, isso pode ser explicitado através do atributo *como-realizar* da classe *Plano* cujo valor é uma lista de objetos do tipo Método:

Tarefa em MAD : nome da tarefa
estado-inicial
estado-final
objetivo
pré-condições
pós-condições
corpo = Estrutura

Plano em TAOS: nome do plano
pré-situação
pós-situação
ocorrência
ações
sub-planos
como-realizar =

Estrutura em MAD : nome estrutura
construtor = ALT
sub-tarefas = lista sub-tarefas

Método em TAOS: nome do método
corpo = expressão

MAD utiliza o *construtor* ALT tanto para representar a decomposição da tarefa em sub-tarefas e seu agenciamento (atributo *corpo*), como na explicitação das diferentes maneiras ou estratégias de se realizar a tarefa (utilizando também o atributo *corpo*). Isto pode gerar confusão no entendimento da relação existente entre as sub-tarefas ou procedimentos. A ilustração desse caso na tarefa-exemplo abaixo, mostra: a utilização do *construtor* ALT no agenciamento das sub-tarefas da sub-tarefa Abrir Arquivo (figura 5.1 -a-) e a representação das diferentes maneiras de executar a sub-tarefa Solicitar Ajuda (figura 5.1 -b-):

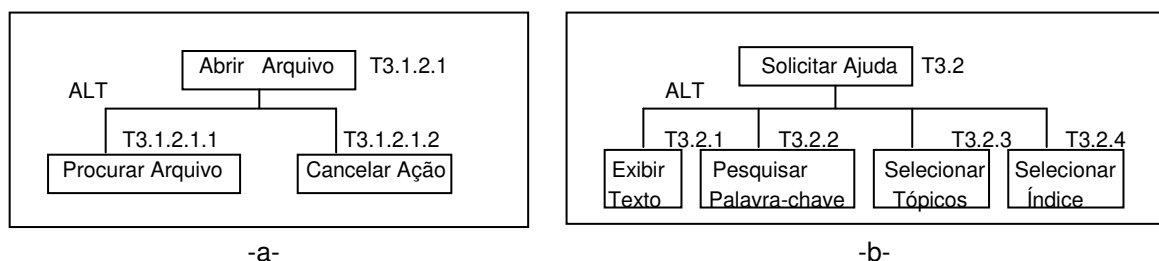


Figura 5.1: Utilização do Construtor ALT em MAD

TAOS possui representações diferentes para cada caso, ou seja, para a decomposição das tarefas e seu agenciamento, atributo *corpo* da classe *Método*, e para as diferentes maneiras ou estratégias de se realizar a tarefa, atributo *como-realizar* da classe *Plano*.

- as condições necessárias para a realização da tarefa

MAD permite a explicitação das restrições sobre o estado do mundo através dos atributos *pré-condição* e *pós-condição* do objeto *Tarefa*, cujos valores são predicados sobre os objetos referenciados pelos atributos *estado-inicial* e *estado-final*, respectivamente. Em TAOS isto é definido utilizando duas classes, a classe *Plano* através dos atributos *pré-situação* e *pós-situação* cujos valores são objetos da classe *Situação* onde então estão listados os objetos envolvidos (atributo *objetos*) e as restrições (atributo *restrição*) sobre os objetos:

Tarefa em MAD : nome da tarefa

estado-inicial
estado-final
 objetivo
pré-condições = predicados
pós-condições = predicados
 corpo

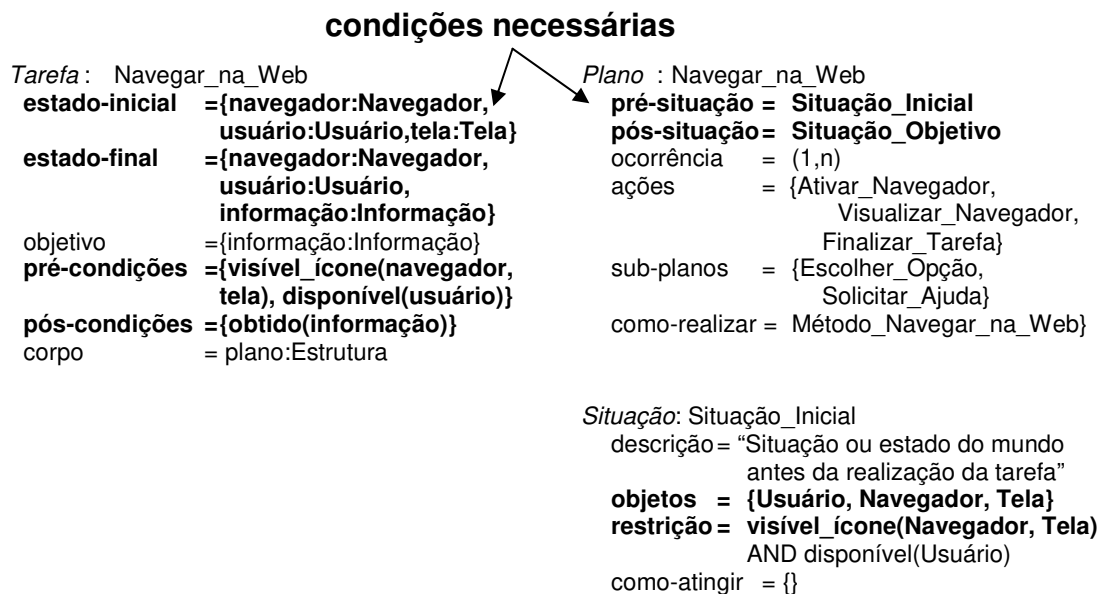
Plano em TAOS: nome do plano

pré-situação =
pós-situação =
 ocorrência
 ações
 sub-planos
 como-realizar

Situação em TAOS: nome

descrição
objetos
restrição
 como-atingir

Na tarefa-exemplo:



Em TAOS, a vantagem de ter os objetos e as restrições representados nas situações, facilita a identificação dos objetos a partir das condições de uma forma mais natural.

- as incoerências e incompletude da descrição

O modelo de representação de MAD e o módulo TAME de TAOS conduzem de uma forma igualmente eficaz a construção incremental da descrição da tarefa, permitindo a verificação *a priori* da coerência e da completude da descrição.

- fatores de criticalidade

A descrição da tarefa tanto MAD como em TAOS permite que possa ser efetuado um levantamento tanto em termos dos objetos envolvidos e que dizem respeito à sua necessidade e ordem em que tornam-se disponíveis, assim como em termos das tarefas (planos) e sub-tarefas (sub-planos) e que dizem respeito à sua freqüência e modo de utilização.

5.4 Conclusão

De acordo com a análise efetuada neste capítulo, TAOS preenche os requisitos exigidos do ponto de vista de análise da tarefa, o que confirma a validade da hipótese central do nosso trabalho. Este fato já tinha sido evidenciado no capítulo anterior pela utilização do formalismo na representação da tarefa-exemplo. Também neste capítulo, foi evidenciado, através da análise comparativa com MAD, que TAOS apresenta algumas vantagens em relação a MAD, tais como, uma maior modularidade na representação de uma tarefa, pois obriga a utilização de objetos diferentes para descrever uma tarefa e os elementos dessa tarefa (modificação num elemento não implica em modificação na tarefa), uma maior naturalidade na expressão da decomposição de uma tarefa, pois evita a introdução de tarefas artificiais na descrição, e um maior poder de expressão com referência aos objetos envolvidos, pois classifica-os em agentes e instrumentos de uma ação. Ou seja, podemos afirmar que TAOS não apenas pode ser usado convenientemente para a representação e análise de tarefa, o que confirma nossa hipótese central, como apresenta vantagens em relação a um dos principais formalismos existentes para tal fim.

Capítulo 6 - Conclusões, Contribuições e Trabalhos Futuros

6.1 Conclusões

Neste trabalho apresentamos o estado da arte no domínio da análise e modelagem da tarefa no contexto de concepção de interfaces homem-computador de sistemas computacionais, o estudo dos formalismos MAD (“*Méthode Analytique de Description*”) e TAOS (“*Task and Action Oriented System*”) e a validação do formalismo TAOS para a análise da tarefa no contexto de concepção de interfaces homem-computador.

Para a validação do formalismo TAOS, demonstramos ou apresentamos evidências suficientes para a confirmação das nossas três hipóteses, cumprindo portanto, com nosso objetivo:

Hipótese 1: No capítulo 2, apresentamos resultados da Teoria da Ação que confirmam que a tarefa e a análise da tarefa do usuário são elementos fundamentais a serem considerados no processo de concepção de interfaces homem-computador, afim de se construir sistemas computacionais com alto grau de usabilidade.

Hipótese 2: De acordo com a análise realizada no capítulo 3 e com os relatos das experiências levadas a cabo em situações complexas ([Sebillote 1987, Sebillote 1998a, Sebillote 1991, Sebillote 1994a]), confirmamos que o formalismo MAD é um elemento de solução para o problema de

representação e análise da tarefa no processo de concepção de interfaces homem-computador de sistemas computacionais.

Hipótese 3 (Hipótese central): No capítulo 4, apresentamos o formalismo TAOS e evidências, através da representação da tarefa-exemplo, de que TAOS é adequado para os objetivos de análise da tarefa. No capítulo 5, apresentamos evidências suficientes para confirmar a nossa hipótese central da adequação do formalismo TAOS para o processo de análise da tarefa no contexto de concepção de interface homem-computador. Ou seja, mostramos que TAOS satisfaz os requisitos exigidos para análise da tarefa e fizemos uma análise comparativa entre MAD e TAOS usando como referência a tarefa-exemplo. Também apresentamos como alguns dos inconvenientes observados durante a utilização de MAD na representação da tarefa-exemplo podem ser evitados com a utilização do TAOS.

Neste trabalho, a existência de grupos de trabalho em IA e ergonomia, orientados tanto para a aquisição e a representação de conhecimento como a análise e modelagem da tarefa, possibilitaram, impulsionaram e fortaleceram a nossa pesquisa, em particular, os grupos de Inteligência Artificial e de Software Design da Universidade Federal da Paraíba (Campina Grande, Brasil) e os grupos: INRIA (Project de Psychologie Ergonomique pour l'Informatique) na França e de Interfaces Homem Computador do Queen Mary College da Inglaterra.

TAOS proporciona uma análise e uma modelagem até certo ponto complexa, no sentido de ter-se que estudar os diferentes conceitos do formalismo para a descrição da tarefa. Mas seu estudo só traz benefícios, já que TAOS facilita, através de sua expressividade e dos conceitos do modelo, realizar uma análise exaustiva *a priori* da tarefa de maneira organizada, e assim obter uma descrição completa e coerente da tarefa.

Uma boa modelagem da tarefa é obtida quando o projetista entende o usuário e seu conhecimento sobre a tarefa e se apoia em um bom

formalismo para descrever a tarefa. Um bom modelo da tarefa, pode levar a um bom modelo de interação dentro da concepção de interfaces homem-computador. Uma boa interface leva a um usuário satisfeito e portanto a uma redução de custos e do tempo de treinamento.

6.2 Contribuições

A maior contribuição deste trabalho é apresentar TAOS como um formalismo que, mesmo concebido para auxiliar na tarefa de aquisição e representação de conhecimento sobre um domínio, pode ser utilizado também para descrever a tarefa do usuário no contexto da concepção de interfaces homem-computador de sistemas computacionais.

TAOS poderia auxiliar os projetistas com uma linguagem comum para se comunicar entre eles e seus usuários no processo da modelagem da tarefa de uma maneira organizada, completa e coerente.

Existem muitos métodos informais para descrever a tarefa onde a estrutura de representação da tarefa não oferece um verdadeiro apoio ao projetista para conceber uma interface homem-computador. Em resposta a esta problemática, este trabalho oferece um estudo formal da modelagem da tarefa no grupo de interfaces homem-computador da UFPB, como um elemento importante na concepção de interfaces homem-computador.

A tarefa navegar na web foi analisada neste trabalho, através dos formalismos MAD e TAOS. Essa tarefa foi escolhida como exemplo, haja vista sua utilização atual em larga escala em diferentes áreas de estudo.

Este estudo é de grande importância para a UFPB, sobretudo nas áreas de IA e Software Design, em linhas de pesquisa que envolvem modelagem da tarefa, levadas a cabo no doutorado de Engenharia Elétrica e no Mestrado de Informática do CCT. Abriram-se pontes de comunicação

com grupos internacionais de pesquisa na área, a exemplo do INRIA (Project de Psychoogie Ergonomique por l'Informatique), da França, e Queen Mary College, da Inglaterra.

A modelagem da tarefa contribui na tomada de decisões para o estudo de como deve ser feita a interação do usuário com o sistema.

TAOS e sua metodologia de modelagem da tarefa pode contribuir com o processo de avaliação que acompanha as várias etapas de concepção de interfaces homem-computador tendo como base a aquisição de conhecimento que o usuário tem sobre sua tarefa.

6.3 Sugestões e Trabalhos Futuros

TAOS é um formalismo aberto e deve ser implementado, em conjunto com seu autor, em um próximo trabalho de dissertação para que sirva como ferramenta de apoio aos projetos de interfaces homem-computador levados a efeito pelo grupo da UFPB.

Também a partir deste trabalho podem-se realizar pesquisas de como obter uma especificação inicial da interface a partir da descrição da tarefa como representada em TAOS e como inserir nessa transformação as recomendações ergonômicas necessárias para a obtenção de interfaces ergonômicas, ou seja, de alto grau de usabilidade.

Referências Bibliográficas

[Bisseret 1979]

Bisseret, A., *Introductory Elements to Ergonomics Research in Man-machine Systems*, Trends in Man-Machine Communications (Ed. Lienard, J.S.), Rocquencourt: INRIA, 1979.

[Card 1983]

Card S., Moran T. e Newell A., *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates, Publ., 1983.

[Cibys 1996]

Cibys W. A., *Ergonomia e Usabilidade de Software*, Labutil, Universidade Federal de Santa Catarina, Florianópolis, Brasil, 1996.

[Falzon 1989]

Falzon P., *Ergonomie Cognitive du Dialogue*, Presses Universitaires de Grenoble, 1989.

[Gamboa 1997]

Gamboa F. e 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.

[Graesser 1980]

Graesser A. C., Kowalsky D. J. e Smith D. A.. Memory for Typical and Atypical Actions in Scripted Activities, *Journal, Exper, Psychology: Human Learning and Memory*, vol. 6, 5, pp., 503-515.

[Green 1985]

Green M. W., *The Design of User Graphical Interfaces*, Ph. D. Thesis., Tech. Report CSRI-70, Computer Systems Research Institute, University of Toronto, Canada, April 1995.

[Haan 1992]

de 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.

[Haan 1999]

de Haan G., All Your Ever Wanted to Know about Extended Task-Action Grammar but Were Afraid to Ask, <http://home.worldonline.nl/~gdehaan/etag.html>, 1999.

[Hammouche 1993]

Hammouche H., *De la Modélisation des Tâches à la Spécification d'Interfaces Utilisateur*, Thèse de Docteur. Centre d'Estudes de la Navigation Aérienne et Institut National de Recherche en Informatique et en Automatique INRIA, France, 1993.

[Heemann 1997]

Heemann V., *Curso de Ergonomia em Sistemas de Informação*, Campina Grande, Julho de 1997.

[Hix 1993]

Hix D. e Hartson H. R., *Developing User Interfaces*, John Wiley & Sons, Inc., U.S.A., 1993.

[Johnson 1988]

Johnson P., Johnson H., Waddington R. e Shouls A. Task-Related Knowledge Structures: *Analysis, Modelling and Application*, Queen Mary College, University of London. 1988.

[Johnson 1991]

Johnson H. e Johnson P, *Task Knowledge Structures: Psychological Basis and Integration Into System Design*, Acta Psychologica 78, pp 3-26. Department of Computer Science, Queen Mary College, University of London. 1991.

[Johnson-Laird 1981]

Johnson-Laird P. N., *Mental Models in Cognitive Science*, Perspectives on Cognitive Science (Ed. Norman, D.A), Ablex, pp. 147-192. 1981.

[Kessel 1995]

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.

[Kieras 1984]

Kieras D.E. e Bovair S., *The Role of a Mental Model in Learning to Operate a Device*, Cognitive Science, vol. 8, No. 3. 1984.

[Kieras 1985]

Kieras D. E. e Polson P.G., *An Approach to the Formal Analysis of User Complexity*, Int. Journal of Man-Machine Studies vol. 22, pp. 365-394, 1985.

[Markopoulos 1992]

Markopoulos P., Pycock J., Wilson S. and Johnson P., *Adept – A Task Based Design Environment*, Queen Mary and Westfield College, 1992.

[Markopoulos 1997]

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.

[Medeiros 1995a]

Medeiros H. e Rousselot F., *Acquisition et Modélisation de Concepts Dynamiques: Le Système TAME*, Rapport ERIC R0102-96, Strasbourg, França, 1995.

[Medeiros 1995b]

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.

[Medeiros 1995c]

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.

[Minsky 1975]

Minsky M., *A Framework for Representing Knowledge*, The Psychology of Computer Vision. Winston, P., (ed), pp. 211-277. McGraw-Hill, New York, 1975.

[Moran 1981]

Moran T., *The Command Language Grammar*, A Representation for the User Interface of Interactive Computer Systems, International Journal of Man-Machine Studies, Vol 15, pp. 3-50, 1981.

[Nanard 1990]

Nanard J., *La Manipulation Directe en Interface Homme-Machine*, Thèse de Docteur, Université Montpellier II, France, 1990.

[Newell 1984]

Newell G. E., *Learning from Writing in Two Content Areas: A Case Study of Protocol Analysis*, Research in the Teaching of English, 18,3, Oct 84, pp. 265-287.

[Nielsen 1986]

Nielsen J., *A Virtual Protocol for Computer-Human Interaction*, Int. Journal of Man-Machine Studies, 24, pp.301-312. 1986.

[Nielsen 1990]

Nielsen J., *A Meta-Model for Interacting with Computers*, Interacting with Computers 2, 2, pp. 147-160.

[Norman 1983]

Norman D. A., *Some Observations on Mental Models. Mental Models*, 7-14. Editado por Dedre Gentner & Albert L. Stevens, Lawrence Erlbaum Associates, Hillsdale, New Jersey. 1983.

[Norman 1986]

Norman D. A. e Drapper S. W., *User Centered System Design*, Lawrence Erlbaum Associates, Publishers. 1986.

[Pierret 1989]

Pierret-Golbreich C., Delouis I. e Scapin D. L., *Un Outil D'Acquisition et de Representation des Taches Oriente-Objet*, Unité de Recherche, Inria-Rocquencourt, France 1989

[Rasmussen 1980]

Rasmussen, J., *The Human as a Systems Component*, Human Interaction with Computers (Ed. Smith, H.T. et Green, T.R.G.), Academic Press, 67-96. 1980.

[Richard 1990]

Richard J. F., *Les Activités Mentales*, Comprendre, Raisonner, Trouver des Solutions, Paris, A. Colin. 1990.

[Sacerdoti 1974a]

Sacerdoti E. D., *Planning in a Hierarchy of Abstraction Spaces*, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California, U.S.A., 1974.

[Sacerdoti 1974b]

Sacerdoti E. D., Informação referenciada de <http://www-cs-students.stanford.edu/~pdoyle/quailuail/notes/pdoyle/search.html#ABSTRIPS>, 1974.

[Sacerdoti 1975]

Sacerdoti E. D., *The Nonlinear Nature of Plans*, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California, U.S.A., 1975.

[Scapin 1987]

Scapin D. L., *Guide Ergonomique de Conception des Interfaces Homme-Machine*, Rapport INRIA No. 77, France, 1987.

[Scapin 1988]

Scapin D. L., *Vers des Outils Formels de Description des Taches Orientes Conception D'Interfaces*, Unité de Recherche, INRIA, Rocquencourt, France, 1988.

[Scapin 1989]

Scapin D. L. e Christine Pierret-Golbreich, *Towards a Method for Task Description: MAD*, Unité de Recherche, INRIA, Rocquencourt, France, 1989.

[Sebillote 1987]

Sebillote S., *La Coception de Scénarios Interactifs: Analyse de L'Activité*, Le Travail Humain, Tome 50, No. 4. Projet de Psychologie Ergonomique pour L'Informatique, INRIA, Domaine de Voluceau, Rocquencourt, Le Chesnay Cedex, France 1987.

[Sebillote 1988a]

Sebillote S., *Théories et Methodologies – Décrire des Taches Selon les Objectifs des opérateurs*, De l'Interview a la Formalization, Unité de Recherche, INRIA, Rocquencourt, France, 1988.

[Sebillote 1988b]

Sebillote S., *Hierarchical Planning as Method for Task Analysis: the example of office task analysis*, Behaviour and Information Technology, 1988, vol. 7, No. 3, pp. 275-293. INRIA, Rocquencourt, Le Chesnay-Cedex, France, 1988.

[Sebillote 1991]

Sebillote S., *Théories et Méthodologies, Décrire des Taches Selon les Objectives des Opérateurs, de l'Interview a la Formalisation*, Projet de Psychologie Ergonomique pour l'Informatique, INRIA, Domaine de Voluceau, Rocquencourt, Le Chesnay Cedex, France, 1991.

[Sebillote 1994a]

Sebillote S. e Alonso B., *Description MAD de la tâche de "contrôle aérien" exécutée par deux contrôleurs*, Projet de Psychologie Ergonomique pour l'Informatique, INRIA, Rocquencourt, Août, 1994.

[Sebillote 1994b]

Sebillote S., Alonso B., Fallah D., Hamouche H., Scapin D.L., e Varnild E., *Note de Recherche Concernant le Formalisme MAD*, Projet de Psychologie Ergonomique pour l'Informatique, INRIA, Ronquencourt, Novembre, 1994.

[Sebillote 1994c]

Sebillote S., *Méthodologie Pratique d'Analyse de la Tâche en Vue de l'Extraction de Caractéristiques Pertinentes pour la Conception d'Interfaces*, Rapport de Recherche No. 163, INRIA, Mars, 1994.

[Sebillote 1994d]

Sebillote S. e Fallah D., *Description MAD d'une tâche complexe: "Resource un incendie en mer"*, Rapport de Contrat, Étude réalisée dans le cadre d'un projet ESPRIT:INTUITIVE P6593, INRIA, Roquencourt, Mai, 1994.

[Sebillote 1994e]

Sebillote S. e Scapin D. L., *From Users' Task Knowledge to High-Level Interface Specification*, International Journal of Human-Computer Interaction 6(1) pp. 1-15, INRIA, Ronquencourt, Le Chesnay, Cedex, France. 1994.

[Sebillote 1995b]

Sebillote S., *Task TD-5: Methodology Guide to Task Analysis with the Goal of Extracting Relevant Characteristics for Interfaces*, Esprit 3 Projet: P6593, "INTUITIVE", INRIA Ronquencourt, April 1995.

[Senach 1990]

Senach B., *Evaluation de l'ergonomie des interfaces homme-machine: Une Revue de la Littérature*, Rapport de Recherche, INRIA, No. 1180, Mars, 1990.

[Shneiderman 1998]

Shneiderman B., *Designing the User Interface – Strategies for effective Humman-Computer Interaction*, Addison Wesley Publishing Co., 3rd Edition, 1998.

[Silva 1999]

da Silva J. C., *Aquisição de Conhecimentos e Manutenção para uma sociedade de Agentes Tutores Artificiais*, Dissertação de mestrado, Universidade Federal da Paraíba, COPIN, 1999.

[Wilson 1992]

Wilson S., Johnson P., Kelly C., Cunningham .J e Markopoulos P., *Beyond Hacking: A Model Based Approach to User Interface Design*, Department of Computer Science, Queen Mary and Westfield College, University of London, Inglaterra, 1992.

[Young 1981]

Young, R.M. *The Machine Inside the Machine: User's Models of Pocket Calculators*, Int. Journal Man-Machine Studies, vol 15, 51-85, 1981.

Bibliografia

[Bastien 1992]

Bastien J.M.C e Scapin D. L., *A Validation of Ergonomic Criteria for the Evaluation of Human – Computer Interfaces*, INRIA, France, 1992.

[Bastien 1995]

Bastien J.M.C. e Scapin D. L., *Evaluating a User Interface With Ergonomic Criteria*, INRIA, France, 1995.

[Coutaz 1990]

Coutaz J., *Interfaces Homme – Ordinateur*, Bordas, Paris, France, 1990.

[Cox 1993]

Cox K. e Walker D., *User-Interface*, Prentice Hall. Singapore 1993.

[Guinn 1998]

Guinn C. I., *An Analysis of Initiative Selection in Collaborative Task-Oriented Discourse*, User Modeling and User-Adapted Interaction Vol 8, pp. 255-314, Kluwer Academic Publishers, Netherlands, 1998.

[Hackos 1998]

Hackos J. T. e Redish J. C., *User and Task Analysis for Interface*, John Wiley and Sons, Inc., 1998.

[Jakobsson 1999]

Jakobsson M., [Analysis and Design of User Interfaces](http://www.uwasa.fi/~mj/hci/hci7.html). <http://www.uwasa.fi/~mj/hci/hci7.html>, Department of Information Technology and Production Economics, Faculty of Industrial Management, University of Vaasa, Finland 1999.

[Kolski 1993]

Kolski C., *Ingénierie des Interfaces Homme-Machine*, Editions Hermès, Paris 1993.

[Lula 1992]

Lula B., *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.

[Moço 1996]

Moço S., *O Uso de Cenários como uma Técnica de Apoio para Avaliações Ergonômicas de Softwares Interativos*, Dissertação de mestrado. Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção, Florianópolis, 1996.

[Norton 1991]

Norton P. e Heid J., *Desvendando o Macintosh*, Editora Campus, Rio de Janeiro, Brasil, 1991.

[Normand 1992]

Normand V., *Le Modèle SIROCO: de la Spécification Conceptuelle des Interfaces Utilisateur à leur Réalisation*, Thèse de Docteur. Université Joseph Fourier, Grenoble I, França, 1992.

[Pierret 1990]

Pierret-Golbreich C. e Delouis I., *Task Centered Representation for Expert Systems at the Knowledge Level*, Laboratoire de Recherche en Informatique, Equipe Intelligence Artificielle et Systèmes d'Inférences, Université Paris Sud, Orsay Cedex, France 1990.

[Preece 1993]

Preece J., *A Guide to Usability*, Addison-Wesley Publishing Company, Great Britain, 1993.

[Queiroz 1998]

de Queiroz J. E. R. e Turnell. M. F. Q. V., *Avaliando a Avaliação: Um SIG como Estudo de Caso*, Relatório Técnico RT00170 COPELE/CCT/UFPB, Campina Grande, Julho, 1998.

[Queiroz 1999]

de Queiroz J. E. R., *Estudo da Relação entre a Usabilidade de Interfaces Computacionais e Fatores de Avaliação*, (Exame de Qualificação), COPELE/CCT/UFPB, Campina Grande, Janeiro, 1999.

[Redmond 1995]

David Redmond-Pyle e Alan Moore, *Graphical User Interface Design and Evaluation*, Prentice Hall, 1995.

[Sebillote 1986]

Sebillote S.. *La Planification Hiérarchique comme Méthode d'Analyse de la Tâche*, Analyse de Taches de Bureau, Research Report 599, INRIA, Rocquencourt, Le Chesnay, France, 1986.

[Sebillote 1995c]

Sebillote S. e Fallah D., *Task TD-2: Task Description "Resolve the Emergency Situation: Fire"*, Raport de Contrat, Esprit 3 Projet: P6593. "INTUITIVE", INRIA Ronquencourt, April, 1995.

[Siochi 1989]

Siochi A. C. e Hartson H. R., *Task-Oriented Representation of Asynchronous User Interfaces*, Department of Computer Science, Chi'89 Proceedings, Virginia, U.S.A., May , 1989.

[Treu 1994]

Treu S., *User Interface Design*, University of Pittsburgh, Plenum Press, New York 1994.

[Turnell 1998]

Turnell M.F.Q.V., *Notas de Aula – Conceitos e Projeto de Interfaces Usuário – Computador*, Laboratório de Interfaces Homem – Máquina, DEE – CCT/UFPB, 1998.

[Waern 1990]

Waern Y., *On the Dynamics of Mental Models*, Mental Models and Human-Computer Interaction 1, Editado por D. Ackermann e M.J. Tauber (Editors), Elsevier Science Publishers B. V. North-Holland, 1990.

Apêndice A - Modelagem da Tarefa-Exemplo Navegar na Web em MAD

A.1 Notação MAD

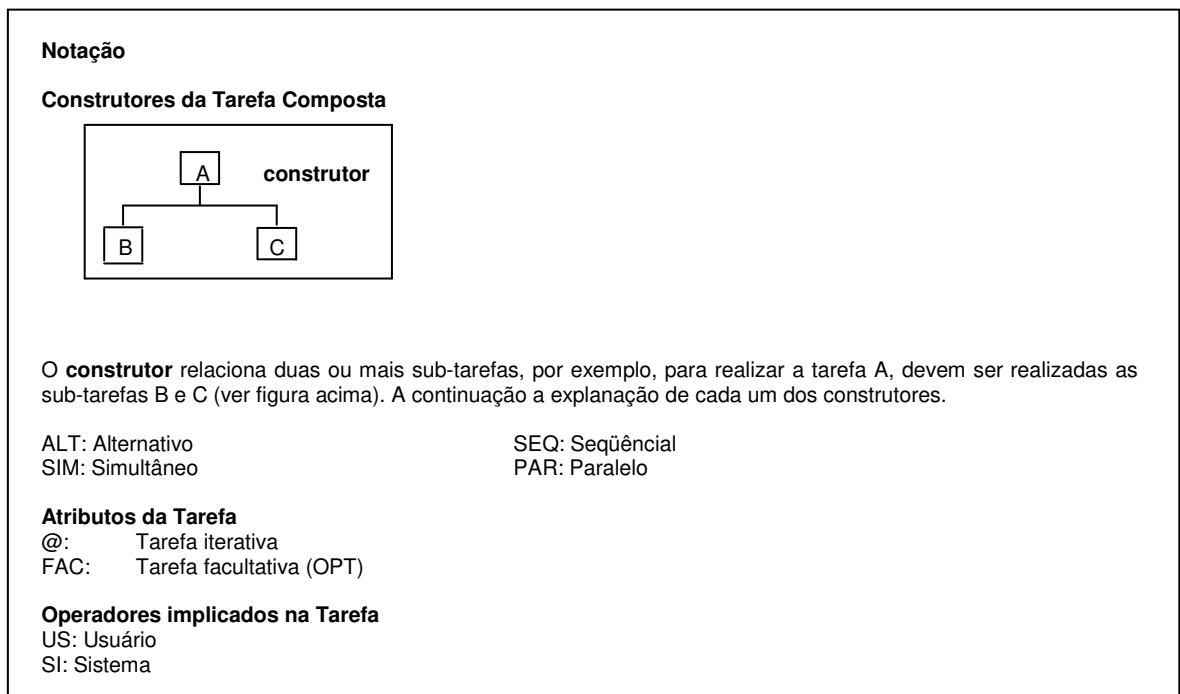


Figura A.1: Notação MAD

A.2 Descrição da Árvore Hierárquica da Tarefa

A continuação a representação gráfica da árvore que decompõe a tarefa Navegar na Web (NW) em sub-tarefas ou ações elementares.

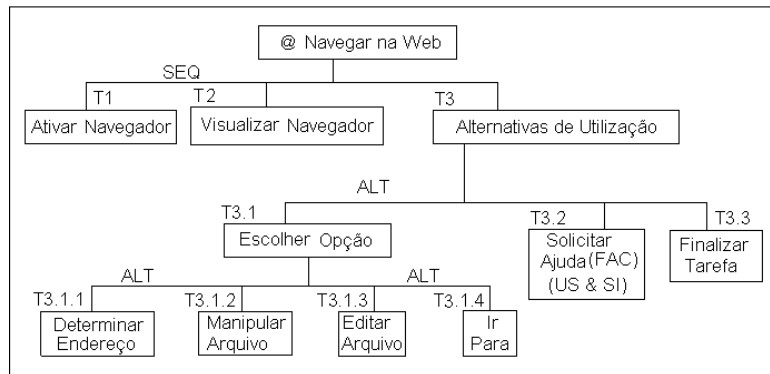


Figura A.2: Decomposição da Tarefa NW

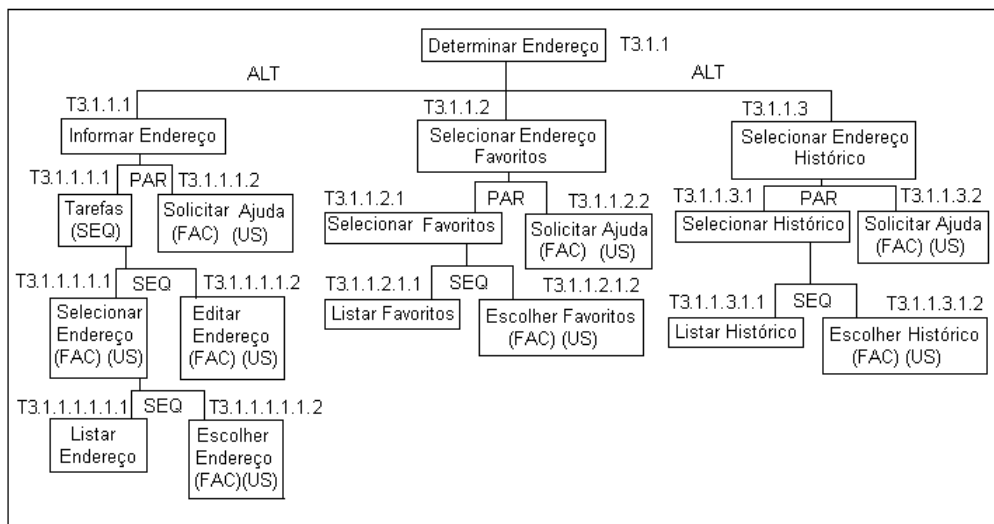


Figura A.3: Sub-Tarefa Determinar Endereço de NW

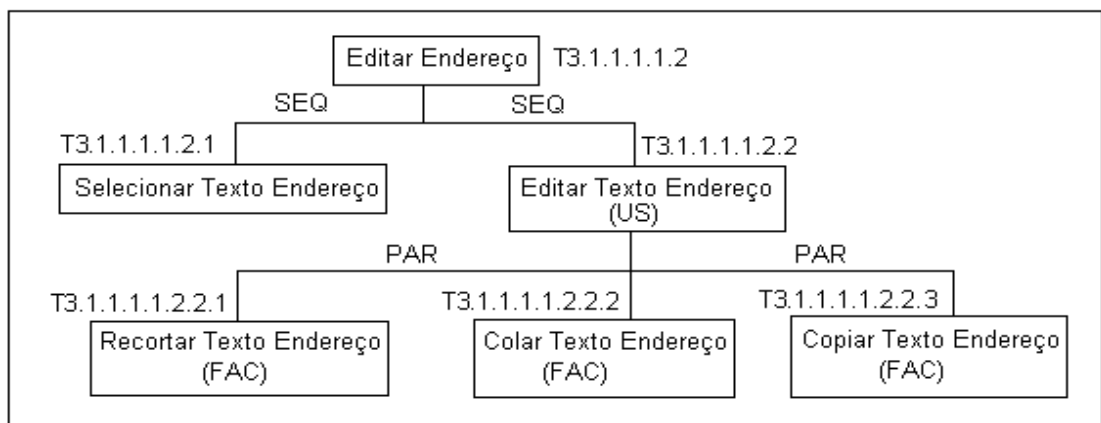


Figura A.4: Sub-Tarefa Editar Endereço de NW

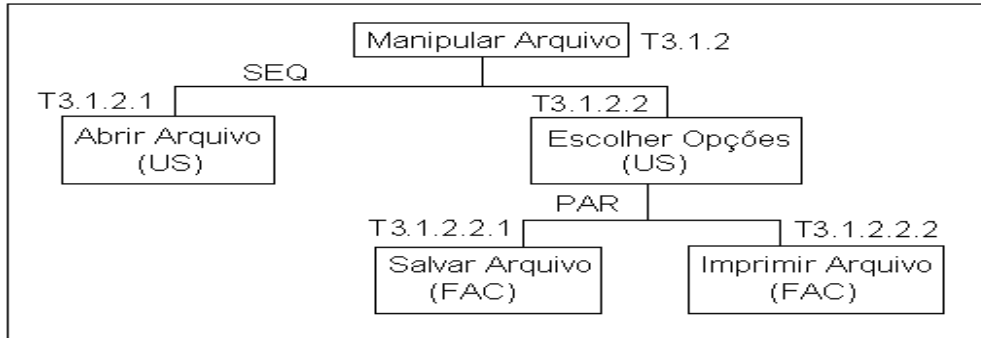


Figura A.5: Sub-Tarefa Manipular Endereço de NW

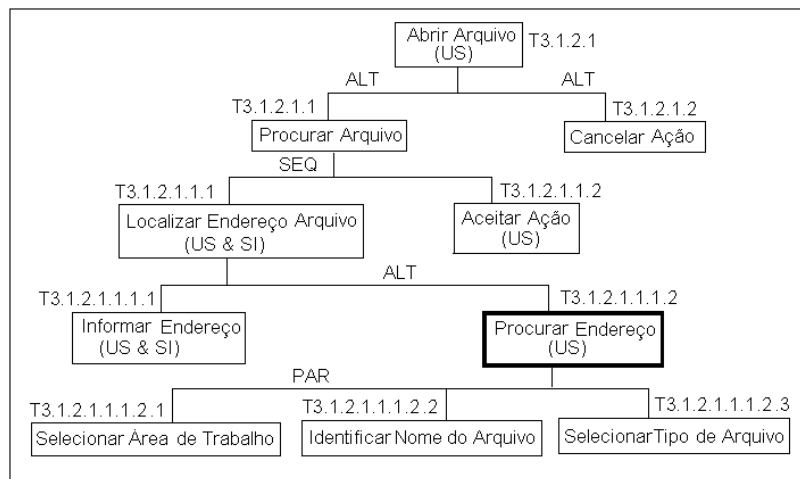


Figura A.6: Sub-Tarefa Abrir Arquivo de NW

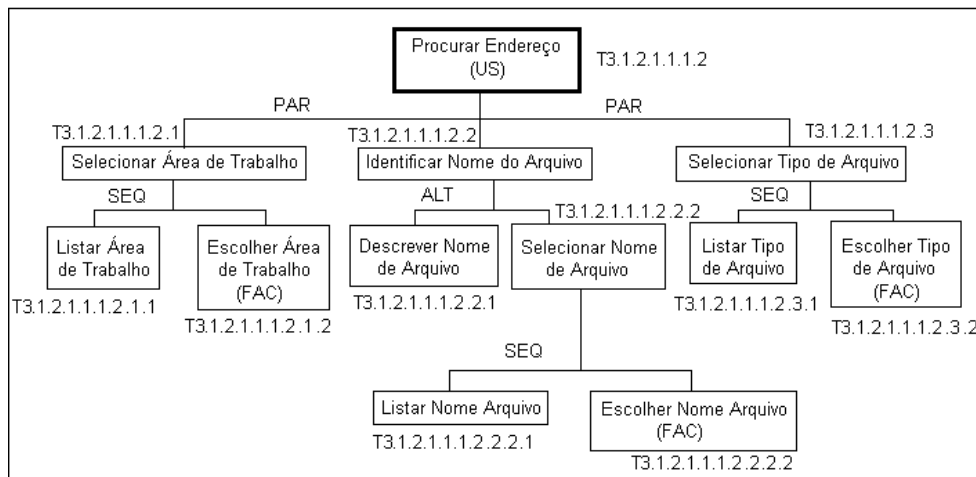


Figura A.7: Sub-Tarefa Procurar Endereço de NW

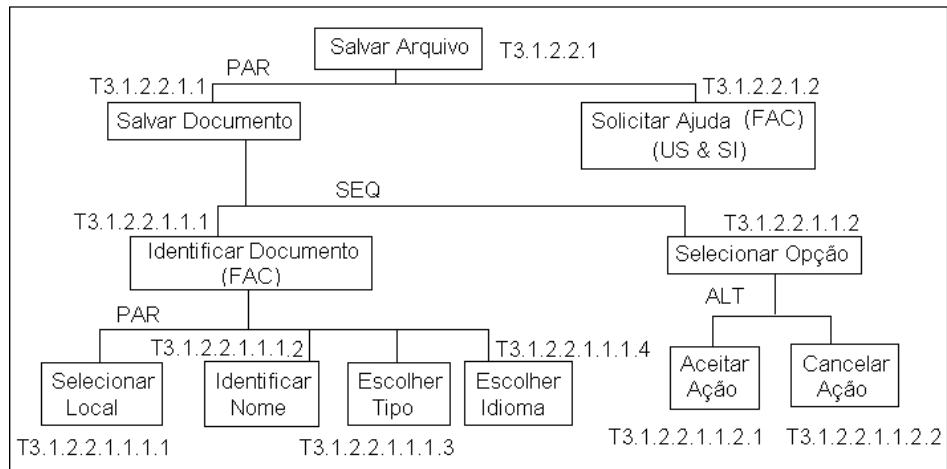


Figura A.8: Sub-Tarefa: Salvar Arquivo de NW

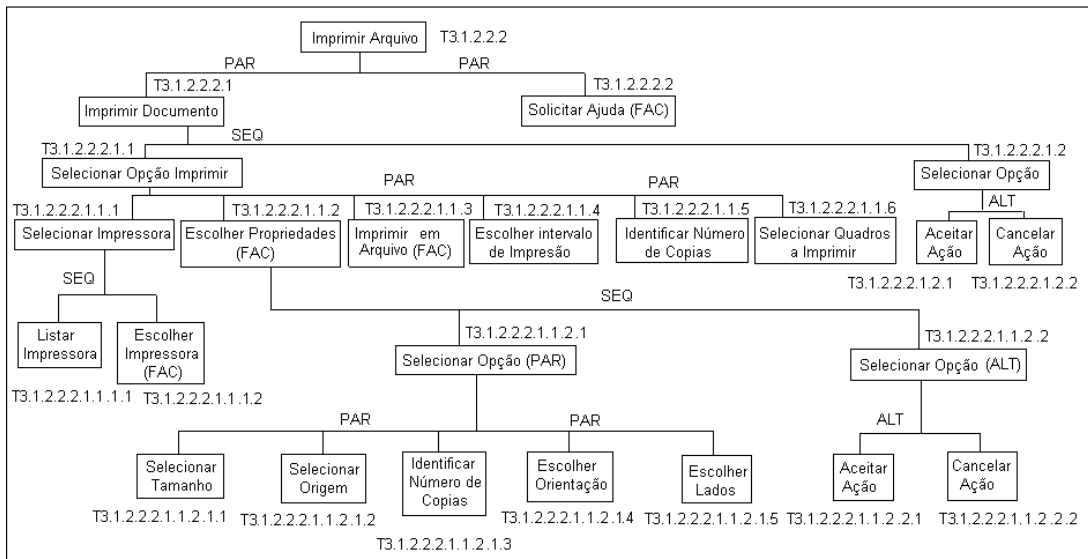


Figura A.9: Sub-Tarefa Imprimir Arquivo de NW

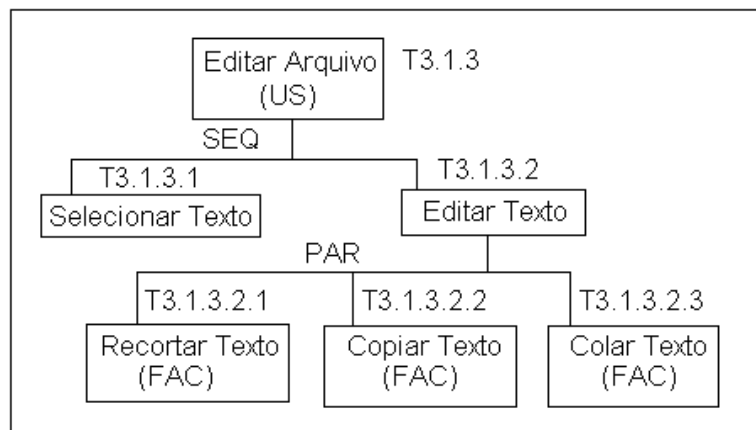


Figura A.10: Sub-Tarefa Editar Arquivo de NW

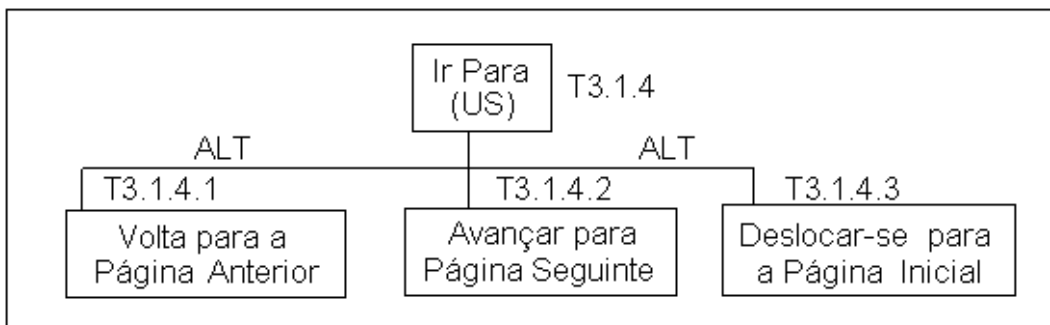


Figura A.11: Sub-Tarefa Ir Para de NW

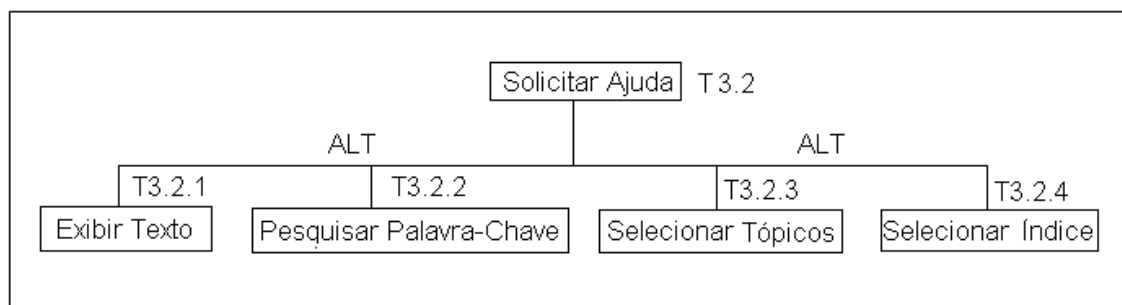


Figura A.12: Sub-Tarefa Solicitar Ajuda de NW

A.3 Definição de Cada Objeto-Tarefa

Em seguida se ilustra a definição de cada objeto-tarefa. A definição de cada tarefa está composta do conjunto de elementos ou atributos da tarefa: nome, estado inicial, objetivo, pré-condições, corpo da tarefa (tarefa elementar ou composta com sua estrutura e sub-tarefas), pós-condições, estado final, etc. As tarefas correspondentes a NW estão identificadas por seu número e nome correspondentes.

TAREFA: "Navegar na Web". Nível Abstrato

TAREFA: Navegar na Web	
ESTADO INICIAL: navegador, usuário (US), tela.	ESTADO FINAL: navegador, usuário, informação.

OBJETIVO: Navegar na Web	
PRÉ-CONDIÇÕES Visível_Ícone(navegador,tela), Disponível(usuário)	PÓS-CONDIÇÕES Obtida(informação)

ESTRUTURA DA TAREFA: Seqüencial (SEQ)	
NÍVEL SUPERIOR: /	TAREFA COMPOSTA: T1. Ativar Navegador T2. Visualizar Navegador T3. Alternativas de Utilização
TAREFA ELEMENTAR: /	

T1

TAREFA: Ativar Navegador	
ESTADO INICIAL: navegador, usuário, tela, mouse	ESTADO FINAL: navegador, tela

OBJETIVO: Inicializar o Navegador	
PRE-CONDIÇÕES Visível_Ícone(navegador, tela), Disponível(usuário)	POS-CONDIÇÕES Disponível(navegador, tela)

ESTRUTURA DA TAREFA: /	
NÍVEL SUPERIOR: "Navegar na Web". (nível abstrato)	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T2

TAREFA: Visualizar Navegador	
ESTADO INICIAL: navegador, usuário (US), tela	ESTADO FINAL: usuário, navegador, tela

OBJETIVO: O usuário vê o navegador na tela	
PRE-CONDIÇÕES Disponível(navegador, tela), Disponível(usuário)	POS-CONDIÇÕES Visto(navegador, usuário, tela)

ESTRUTURA DA TAREFA: /	
NÍVEL SUPERIOR: "Navegar na Web". (nível abstrato)	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3

TAREFA: Alternativas de Utilização	
ESTADO INICIAL: navegador, opções, endereço, favoritos, histórico, arquivo, lista de páginas, usuário (US)	ESTADO FINAL: opções

OBJETIVO: Utilizar opções do Navegador para manipular a informação	
PRÉ-CONDIÇÕES Disponível(navegador, tela), Disponível(usuário) Disponível(opções) ou Disponível(endereço) ou Disponível(favoritos) ou Disponível(histórico) ou Disponível(arquivo) ou Disponível(lista de páginas)	PÓS-CONDIÇÕES Selecionadas(opções), Solicitada(ajuda)

ESTRUTURA DA TAREFA: alternativa (ALT)	
NÍVEL SUPERIOR: "Navegar na Web". (nível abstrato)	TAREFA COMPOSTA: T3.1 Escolher Opção T3.2 Solicitar Ajuda T3.3 Finalizar Tarefa
TAREFA ELEMENTAR: /	

T3.1

TAREFA: Escolher Opção	
ESTADO INICIAL: ver T3	ESTADO FINAL: opções

OBJETIVO: Selecionar opções do Navegador para manipular informação	
PRÉ-CONDIÇÕES Ver T3	POS-CONDIÇÕES Selecionadas(opções)

ESTRUTURA DA TAREFA: alternativa (ALT)	
NÍVEL SUPERIOR: T3 Alternativas de Utilização	TAREFA COMPOSTA: T3.1.1 Determinar Endereço T3.1.2 Manipular Arquivo T3.1.3 Editar Arquivo T3.1.4 Ir Para
TAREFA ELEMENTAR:/	

T3.1.1

TAREFA: Determinar Endereço	
ESTADO INICIAL: ver T3	ESTADO FINAL: endereço

OBJETIVO: Procurar um endereço específico	
PRÉ-CONDIÇÕES Ver T3	POS-CONDIÇÕES Obtido(endereço) ou Selecionado(favoritos) ou Selecionado(histórico)

ESTRUTURA DA TAREFA: Alternativa (ALT)	
NÍVEL SUPERIOR: T3.1 Escolher Opção	TAREFA COMPOSTA: T3.1.1.1 Informar Endereço T3.1.1.2 Selecionar Favoritos T3.1.1.3 Selecionar Histórico
TAREFA ELEMENTAR:/	

T3.1.1.1

TAREFA: Informar Endereço	
ESTADO INICIAL: ver T3	ESTADO FINAL: endereço

OBJETIVO: Procurar informação que ajude a localizar endereço	
PRÉ-CONDIÇÕES Ver T3	POS-CONDIÇÕES Selecionado(endereço), Editado(endereço), Solicitada(ajuda)

ESTRUTURA DA TAREFA: Paralela (PAR)	
NÍVEL SUPERIOR: T3.1.1 Determinar Endereço	TAREFA COMPOSTA: T3.1.1.1.1 Tarefas (SEQ) T3.1.1.1.2 Solicitar Ajuda
TAREFA ELEMENTAR:/	

T3.1.1.1.1

TAREFA: Tarefas (SEQ)	
ESTADO INICIAL: ver T3	ESTADO FINAL: endereço

OBJETIVO: Selecionar endereço para editá-lo	
PRÉ-CONDIÇÕES Ver T3	POS-CONDIÇÕES Selecionado(endereço), Editado(endereço)

ESTRUTURA DA TAREFA: Seqüencial (SEQ)	
NÍVEL SUPERIOR: T3.1.1.1.1 Informar Endereço	TAREFA COMPOSTA: T3.1.1.1.1.1 Selecionar Endereço T3.1.1.1.1.2 Editar Endereço
TAREFA ELEMENTAR:/	

T3.1.1.1.1.1

TAREFA: Selecionar Endereço	
ESTADO INICIAL: ver T3	ESTADO FINAL: endereço

OBJETIVO: Selecionar endereço para trabalhar sobre o Navegador	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Listado(endereço) Escolhido(endereço)

ESTRUTURA DA TAREFA: Seqüencial (SEQ)	
NÍVEL SUPERIOR: T3.1.1.1.1 Tarefas (SEQ)	TAREFA COMPOSTA: T3.1.1.1.1.1 Listar Endereço T3.1.1.1.1.2 Escolher Endereço
TAREFA ELEMENTAR: /	

T3.1.1.1.1.1

TAREFA: Listar Endereço	
ESTADO INICIAL: ver T3	ESTADO FINAL: endereço

OBJETIVO: Visualizar informação de endereços existentes	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Listado(endereço)

ESTRUTURA DA TAREFA: /	
NÍVEL SUPERIOR: T3.1.1.1.1 Selecionar Endereço	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3.1.1.1.1.2

TAREFA: Escolher Endereço	
ESTADO INICIAL: ver T3	ESTADO FINAL: endereço

OBJETIVO: Selecionar o endereço desejado a partir dos endereços existentes	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Escolhido(endereço)

ESTRUTURA DA TAREFA: /	
NÍVEL SUPERIOR: T1.1.1.1.1. Visualizar Endereço	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3.1.1.1.1.2

TAREFA: Editar Endereço	
ESTADO INICIAL: ver T3	ESTADO FINAL: endereço

OBJETIVO: Visualizar informação de endereços existentes	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Selecionado(texto endereço) Editado(texto endereço)

ESTRUTURA DA TAREFA: seqüencial (SEQ)	
NÍVEL SUPERIOR: T3.1.1.1.1 Tarefas (SEQ)	TAREFA COMPOSTA: T3.1.1.1.2.1 Selecionar Texto Endereço T3.1.1.1.2.2 Editar Texto Endereço
TAREFA ELEMENTAR: /	

T3.1.1.1.2.1

TAREFA: Selecionar Texto Endereço	
ESTADO INICIAL: ver T3	ESTADO FINAL: texto endereço
OBJETIVO: Escolher texto do endereço existente	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Selecionado(texto endereço)
ESTRUTURA DA TAREFA: /	
NÍVEL SUPERIOR: T3.1.1.1.2 Editar Endereço	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3.1.1.1.2.2

TAREFA: Editar Texto Endereço	
ESTADO INICIAL: ver T3	ESTADO FINAL: texto endereço
OBJETIVO: Modificar texto existente	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Recortado(texto endereço) Colado(texto endereço) Copiado(texto endereço)
ESTRUTURA DA TAREFA: Paralela (PAR)	
NÍVEL SUPERIOR: T3.1.1.1.2 Editar Endereço	TAREFA COMPOSTA: T3.1.1.1.2.2.1 Recortar Texto Endereço T3.1.1.1.2.2.2 Colar Texto Endereço T3.1.1.1.2.2.3 Copiar Texto Endereço
TAREFA ELEMENTAR: /	

T3.1.1.1.2.2.1

TAREFA: Recortar Texto Endereço	
ESTADO INICIAL: ver T3	ESTADO FINAL: texto endereço
OBJETIVO: Suprimir texto de um endereço existente	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Recortado(texto endereço)
ESTRUTURA DA TAREFA: /	
NÍVEL SUPERIOR: T3.1.1.1.2.2 Editar Texto Endereço	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3.1.1.1.2.2.2

TAREFA: Colar Texto Endereço	
ESTADO INICIAL: ver T3	ESTADO FINAL: texto endereço
OBJETIVO: Acrescentar texto a um endereço existente	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Colado(texto endereço)
ESTRUTURA DA TAREFA: /	
NÍVEL SUPERIOR: T3.1.1.1.2.2 Editar Texto Endereço	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3.1.1.1.1.2.2.3

TAREFA: Copiar Texto Endereço	
ESTADO INICIAL: ver T3	ESTADO FINAL: texto endereço
OBJETIVO: Trasladar texto de um endereço existente	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Copiado(texto endereço)
ESTRUTURA DA TAREFA: /	
NIVEL SUPERIOR: T3.1.1.1.1.2.2 Editar Texto Endereço	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3.1.1.2

TAREFA: Selecionar Endereço Favoritos	
ESTADO INICIAL: ver T3	ESTADO FINAL: endereço de favoritos
OBJETIVO: Visualizar favoritos existentes	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Selecionados(favoritos) Solicitada(ajuda)
ESTRUTURA DA TAREFA: paralela (PAR)	
NIVEL SUPERIOR: T3.1.1 Determinar Endereço	TAREFA COMPOSTA: T3.1.1.2.1 Selecionar Favoritos T3.1.1.2.2 Solicitar Ajuda
TAREFA ELEMENTAR: /	

T3.1.1.2.1

TAREFA: Selecionar Favoritos	
ESTADO INICIAL: ver T3	ESTADO FINAL: favoritos
OBJETIVO: Visualizar favoritos existentes	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Listados(favoritos) Escolhidos(favoritos)
ESTRUTURA DA TAREFA: seqüencial (SEQ)	
NIVEL SUPERIOR: T3.1.1.2 Selecionar Endereço Favoritos	TAREFA COMPOSTA: T3.1.1.2.1.1 Listar Favoritos T3.1.1.2.1.2 Escolher Favoritos
TAREFA ELEMENTAR: /	

T3.1.1.2.1.1

TAREFA: Listar Favoritos	
ESTADO INICIAL: ver T3	ESTADO FINAL: favoritos
OBJETIVO: Visualizar favoritos existentes	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Listados(favoritos)
ESTRUTURA DA TAREFA: /	
NIVEL SUPERIOR: T3.1.1.2.1 Selecionar Favoritos	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3.1.1.2.1.2

TAREFA: Escolher Favoritos	
ESTADO INICIAL: ver T3	ESTADO FINAL: favoritos
OBJETIVO: Selecionar favoritos da lista	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Escolhidos(favoritos)
ESTRUTURA DA TAREFA: /	
NÍVEL SUPERIOR: T3.1.1.2.1 Selecionar Favoritos	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3.1.1.3

TAREFA: Selecionar Endereço Histórico	
ESTADO INICIAL: ver T3	ESTADO FINAL: endereço histórico
OBJETIVO: Visualizar histórico existente	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Selecionado(endereço histórico) Solicitada(ajuda)
ESTRUTURA DA TAREFA: paralela (PAR)	
NÍVEL SUPERIOR: T3.1.1 Determinar Endereço	TAREFA COMPOSTA: T3.1.1.3.1 Selecionar Histórico T3.1.1.3.2 Solicitar Ajuda
TAREFA ELEMENTAR: /	

T3.1.1.3.1

TAREFA: Selecionar Histórico	
ESTADO INICIAL: ver T3	ESTADO FINAL: histórico
OBJETIVO: Visualizar informação de endereços existentes	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Listado(histórico) Escolhido(endereço histórico)
ESTRUTURA DA TAREFA: seqüencial (SEQ)	
NÍVEL SUPERIOR: T3.1.1.3 Selecionar Endereço Histórico	TAREFA COMPOSTA: T3.1.1.3.1.1 Listar Histórico T3.1.1.3.1.2 Escolher Histórico
TAREFA ELEMENTAR: /	

T3.1.1.3.1.1

TAREFA: Listar Histórico	
ESTADO INICIAL: ver T3	ESTADO FINAL: Histórico
OBJETIVO: Visualizar favoritos existentes	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Listado(histórico)
ESTRUTURA DA TAREFA: /	
NÍVEL SUPERIOR: T3.1.1.3.1 Selecionar Histórico	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3.1.1.3.1.2

TAREFA: Escolher Histórico	
ESTADO INICIAL: ver T3	ESTADO FINAL: Histórico
OBJETIVO: Selecionar endereço histórico da lista	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Escolhido(endereço histórico)
ESTRUTURA DA TAREFA: /	
NÍVEL SUPERIOR: T3.1.1.3.1 Selecionar Histórico	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3.1.2

TAREFA: Manipular Arquivo	
ESTADO INICIAL: ver T3	ESTADO FINAL: Arquivo
OBJETIVO: Utilizar o arquivo existente segundo os propósitos do usuário	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Aberto(arquivo) Escolhidas(opções)
ESTRUTURA DA TAREFA: Seqüencial (SEQ)	
NÍVEL SUPERIOR: T3.1 Escolher Opção	TAREFA COMPOSTA: T3.1.2.1 Abrir Arquivo T3.1.2.2 Escolher Opções
TAREFA ELEMENTAR: /	

T3.1.2.1

TAREFA: Abrir Arquivo	
ESTADO INICIAL: ver T3	ESTADO FINAL: Arquivo
OBJETIVO: Utilizar o arquivo existente segundo os propósitos do usuário	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Procurado(arquivo) ou Cancelada(ação)
ESTRUTURA DA TAREFA: alternativa (ALT)	
NÍVEL SUPERIOR: T3.2.1 Manipular Arquivo	TAREFA COMPOSTA: T3.1.2.1.1 Procurar Arquivo T3.1.2.1.2 Cancelar Ação
TAREFA ELEMENTAR: /	

T3.1.2.1.1

TAREFA: Procurar Arquivo	
ESTADO INICIAL: ver T3	ESTADO FINAL: Arquivo
OBJETIVO: Utilizar o arquivo existente segundo os propósitos do usuário	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Procurado(arquivo) Aceitada(ação)
ESTRUTURA DA TAREFA: Seqüencial (SEQ)	
NÍVEL SUPERIOR: T3.1.2.1 Abrir Arquivo	TAREFA COMPOSTA: T3.1.2.1.1.1 Localizar Endereço Arquivo T3.1.2.1.1.2 Aceitar Ação
TAREFA ELEMENTAR: /	

T3.1.2.1.1.1

TAREFA: Localizar Endereço Arquivo	
ESTADO INICIAL: ver T3	ESTADO FINAL: Endereço do arquivo
OBJETIVO: Utilizar o arquivo existente segundo os propósitos do usuário	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Obtido(endereço)
ESTRUTURA DA TAREFA: Alternativa (ALT)	
NÍVEL SUPERIOR: T3.1.2.1.1 Arquivo	TAREFA COMPOSTA: T3.1.2.1.1.1 Informar Endereço T3.1.2.1.1.2 Procurar Endereço
TAREFA ELEMENTAR: /	

T3.1.2.1.1.1.2

TAREFA: Procurar Endereço	
ESTADO INICIAL: ver T3	ESTADO FINAL: Endereço
OBJETIVO: Localizar um endereço existente	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Selecionada(área de trabalho) Identificado(nome do arquivo) Selecionado(tipo de arquivo)
ESTRUTURA DA TAREFA: Paralela (PAR)	
NÍVEL SUPERIOR: T1.1.2.1.1.1 Localizar Endereço	TAREFA COMPOSTA: T3.1.2.1.1.1.2.1 Selecionar Área de Trabalho T3.1.2.1.1.1.2.2 Identificar Nome do Arquivo T3.1.2.1.1.1.2.3 Selecionar Tipo de Arquivo
TAREFA ELEMENTAR: /	

T3.1.2.1.1.1.2.1

TAREFA: Selecionar Área de Trabalho	
ESTADO INICIAL: ver T3	ESTADO FINAL: Área de trabalho
OBJETIVO: Localizar um endereço existente	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Listada(área de trabalho) Escolhida(área de trabalho)
ESTRUTURA DA TAREFA: sequencial (SEQ)	
NÍVEL SUPERIOR: T3.1.2.1.1.1.2 Procurar Endereço	TAREFA COMPOSTA: T3.1.2.1.1.1.2.1.1 Listar Área de Trabalho T3.1.2.1.1.1.2.1.2 Escolher Área de Trabalho
TAREFA ELEMENTAR: /	

T3.1.2.1.1.1.2.1.1

TAREFA: Listar Área de Trabalho	
ESTADO INICIAL: ver T3	ESTADO FINAL: Área de trabalho
OBJETIVO: Visualizar área de trabalho	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Listada(área de trabalho)
ESTRUTURA DA TAREFA: /	
NÍVEL SUPERIOR: T3.1.2.1.1.1.2.1 Selecionar Área de Trabalho	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3.1.2.1.1.1.2.1.2

TAREFA: Escolher Área de Trabalho	
ESTADO INICIAL: ver T3	ESTADO FINAL: Área de trabalho
OBJETIVO: Selecionar endereço na área de trabalho	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Escolhida(área de trabalho)
ESTRUTURA DA TAREFA: /	
NÍVEL SUPERIOR: T3.1.2.1.1.1.2.1 Selecionar Área de Trabalho	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3.1.2.1.1.1.2.2

TAREFA: Identificar Nome do Arquivo	
ESTADO INICIAL: ver T3	ESTADO FINAL: Nome de arquivo
OBJETIVO: Localizar um endereço existente	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Descrito(nome do arquivo) Selecionado(nome do arquivo)
ESTRUTURA DA TAREFA: alternativa (ALT)	
NÍVEL SUPERIOR: T3.1.2.1.1.1.2 Procurar Endereço	TAREFA COMPOSTA: T3.1.2.1.1.1.2.2.1 Descrever Nome de Arquivo T3.1.2.1.1.1.2.2.2 Selecionar Nome de Arquivo
TAREFA ELEMENTAR: /	

T3.1.2.1.1.1.2.2.1

TAREFA: Descrever Nome de Arquivo	
ESTADO INICIAL: ver T3	ESTADO FINAL: Nome do arquivo
OBJETIVO: Selecionar endereço na área de trabalho	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Descrito(nome do arquivo)
ESTRUTURA DA TAREFA: /	
NÍVEL SUPERIOR: T3.1.2.1.1.1.2.2 Identificar Nome do Arquivo	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3.1.2.1.1.1.2.2.2

TAREFA: Selecionar Nome de Arquivo	
ESTADO INICIAL: ver T3	ESTADO FINAL: Nome do arquivo
OBJETIVO: Localizar um endereço existente	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Listado(nome do arquivo) Escolhido(nome do arquivo)
ESTRUTURA DA TAREFA: sequencial (SEQ)	
NÍVEL SUPERIOR: T3.1.2.1.1.1.2.2 Identificar Nome do Arquivo	TAREFA COMPOSTA: T3.1.2.1.1.1.2.2.2.1 Listar Nome do Arquivo T3.1.2.1.1.1.2.2.2.2 Escolher Nome Arquivo
TAREFA ELEMENTAR: /	

T3.1.2.1.1.1.2.2.2.1

TAREFA: Listar Nome do Arquivo	
ESTADO INICIAL: ver T3	ESTADO FINAL: Nome de arquivo
OBJETIVO: Visualizar nomes de arquivos existentes	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Listado(nome do arquivo)
ESTRUTURA DA TAREFA: /	
NÍVEL SUPERIOR: T3.1.2.1.1.1.2.2.2 Selecionar Nome de Arquivo	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3.1.2.1.1.1.2.2.2.2

TAREFA: Escolher Nome do Arquivo	
ESTADO INICIAL: ver T3	ESTADO FINAL: Nome do Arquivo
OBJETIVO: Escolher nome do arquivo na lista	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Escolhido(nome do arquivo)
ESTRUTURA DA TAREFA: /	
NÍVEL SUPERIOR: T3.1.2.1.1.1.2.2.2 Selecionar Nome de Arquivo	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3.1.2.1.1.1.2.3

TAREFA: Selecionar Tipo de Arquivo	
ESTADO INICIAL: ver T3	ESTADO FINAL: Tipo do arquivo
OBJETIVO: Localizar um endereço existente	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Listado(tipo do arquivo) Escolhido(tipo do arquivo)
ESTRUTURA DA TAREFA: sequencial (SEQ)	
NÍVEL SUPERIOR: T3.1.2.1.1.1.2 Procurar Endereço	TAREFA COMPOSTA: T3.1.2.1.1.1.2.3.1 Listar Tipo de Arquivo T3.1.2.1.1.1.2.3.2 Escolher Tipo de Arquivo
TAREFA ELEMENTAR: /	

T3.1.2.1.1.1.2.3.1

TAREFA: Listar Tipo de Arquivo	
ESTADO INICIAL: ver T3	ESTADO FINAL: Tipo de arquivo
OBJETIVO: Visualizar tipos de arquivos existentes	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Listado(tipo do arquivo)
ESTRUTURA DA TAREFA: /	
NÍVEL SUPERIOR: T3.1.2.1.1.1.2.3 Selecionar Tipo de Arquivo	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3.1.2.1.1.2.3.2

TAREFA: Escolher Tipo de Arquivo	
ESTADO INICIAL: ver T3	ESTADO FINAL: Tipo do Arquivo
OBJETIVO: Escolher tipo do arquivo na lista	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Escolhido(tipo do arquivo)
ESTRUTURA DA TAREFA: /	
NÍVEL SUPERIOR: T3.1.2.1.1.2.3 Selecionar Tipo de Arquivo	TAREFA COMPOSTA: /
TAREFA ELEMENTAR: Sim	

T3.1.2.2

TAREFA: Escolher Opções	
ESTADO INICIAL: ver T3	ESTADO FINAL: Opções
OBJETIVO: Escolher opções segundo os propósitos do usuário	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Salvo(arquivo) Impresso(arquivo)
ESTRUTURA DA TAREFA: paralela (PAR)	
NÍVEL SUPERIOR: T3.1.2 Manipular Arquivo	TAREFA COMPOSTA: T3.1.2.2.1 Salvar Arquivo T3.1.2.2.2 Imprimir Arquivo
TAREFA ELEMENTAR: /	

T3.1.2.2.1

TAREFA: Salvar Arquivo	
ESTADO INICIAL: ver T3	ESTADO FINAL: Arquivo
OBJETIVO: Salvar documento segundo os objetivos do usuário	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Salvo(documento) Solicitada(Ajuda)
ESTRUTURA DA TAREFA: paralela (PAR)	
NÍVEL SUPERIOR: T3.1.2.2 Escolher Opções	TAREFA COMPOSTA: T3.1.2.2.1.1 Salvar Documento T3.1.2.2.1.2 Solicitar Ajuda
TAREFA ELEMENTAR: /	

T3.1.2.2.2

TAREFA: Imprimir Arquivo	
ESTADO INICIAL: ver T3	ESTADO FINAL: Arquivo
OBJETIVO: Imprimir documento segundo os objetivos do usuário	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Impresso(documento) Solicitada(ajuda)
ESTRUTURA DA TAREFA: paralela (PAR)	
NÍVEL SUPERIOR: T3.1.2.2 Escolher Opções	TAREFA COMPOSTA: T3.1.2.2.2.1 Imprimir Documento T3.1.2.2.2.2 Solicitar Ajuda
TAREFA ELEMENTAR: /	

T3.1.3

TAREFA: Editar Arquivo	
ESTADO INICIAL: ver T3	ESTADO FINAL: Arquivo
OBJETIVO: Utilizar o arquivo existente segundo os propósitos do usuário	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Selecionado(texto) Editado(texto)
ESTRUTURA DA TAREFA: Seqüencial (SEQ)	
NÍVEL SUPERIOR: T3.1 Escolher Opção	TAREFA COMPOSTA: T3.1.3.1 Selecionar Texto T3.1.3.2 Editar Texto
TAREFA ELEMENTAR: /	

T3.1.4

TAREFA: Ir Para	
ESTADO INICIAL: ver T3	ESTADO FINAL: Foi para
OBJETIVO: Movimentar-se no Navegador	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES FoiPara(pagina anterior) ou FoiPara(pagina seguinte) ou FoiPara (pagina inicial)
ESTRUTURA DA TAREFA: Alternativa (ALT)	
NÍVEL SUPERIOR: T3.1 Escolher Opção	TAREFA COMPOSTA: T3.1.4.1 Volta para a Página Anterior T3.1.4.2 Avançar para a Página Seguinte T3.1.4.3 Deslocar-se para a Página Inicial
TAREFA ELEMENTAR: /	

T3.2

TAREFA: Solicitar Ajuda	
ESTADO INICIAL: ver T3	ESTADO FINAL: Ajuda solicitada
OBJETIVO: Procurar informação que ajuda na navegação no Navegador	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Exibido(texto) Pesquisada(palavra chave) Selecionados(tópicos) Selecionado(Índice)
ESTRUTURA DA TAREFA: alternativa (ALT)	
NÍVEL SUPERIOR: T3 Alternativas de Utilização	TAREFA COMPOSTA: T3.2.1 Exibir Texto T3.2.2 Pesquisar Palavra – Chave T3.2.3 Selecionar Tópicos T3.2.4 Selecionar Índice
TAREFA ELEMENTAR: /	

T3.3

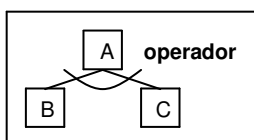
TAREFA: Finalizar Tarefa	
ESTADO INICIAL: ver T3	ESTADO FINAL: Informação
OBJETIVO: Usuário da por terminada a execução da tarefa	
PRE-CONDIÇÕES ver T3	POS-CONDIÇÕES Obtida(informação)
ESTRUTURA DA TAREFA: alternativa (ALT)	
NÍVEL SUPERIOR: T3 Alternativas de Utilização	TAREFA COMPOSTA:
TAREFA ELEMENTAR: Sim	

Apêndice B - Modelagem da Tarefa-Exemplo Navegar na Web em TAOS

B.1 Notação TAOS

Notação Gráfica

Operadores do Plano



O **operador** relaciona dois ou mais planos, sub-planos ou ações, por exemplo, para executar o plano (tarefa) A, devem ser executadas os sub-planos ou ações (sub-tarefas) B e C (ver figura acima). A continuação a explicação de cada um dos operadores.

OR: Alternativo

XOR: Or exclusivo

SEQ: As tarefas se executam numa ordem total

AND: Todas as tarefas são executadas sem qualquer ordem cronológica

SIM: Todas as tarefas são executadas ao mesmo tempo

Ocorrência (mín, máx)

(0,0): Ação ou plano proibido (equivalente ao NOT)

(0,1): Ação ou plano facultativo, mas se ocorrer, ocorrerá apenas uma vez

(0,n): Ação ou plano executado zero ou mais vezes. É facultativo mas si ocorrer, poderá ocorrer 1 ou mais vezes

(1,1): Ação ou plano executado apenas uma vez

(1,n): Ação ou plano executado uma ou mais vezes

<W <situação S>> : Ação ou plano executado enquanto a situação S persistir

Figura B.1: Notação TAOS

B.2 Descrição da Árvore Hierárquica da Tarefa

Apresentamos a seguir a representação gráfica ou árvore da nossa proposta de resolução da tarefa (plano) Navegar na Web (NW) em TAOS, descomposto em sub-tarefas (sub-planos) e ações elementares.

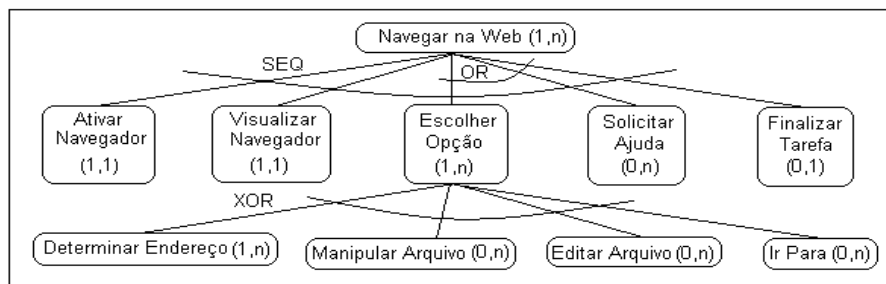


Figura B.2: Plano (Tarefa) Navegar na Web

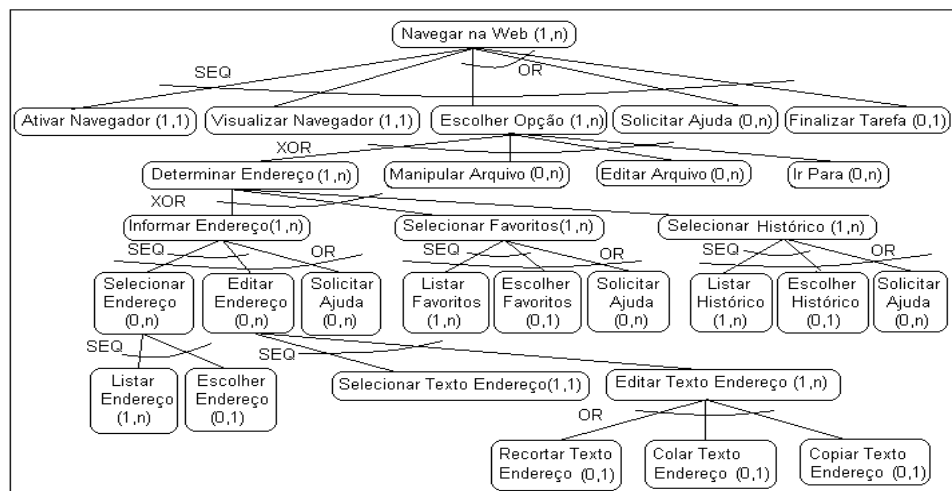


Figura B.3: Sub-Plano (Sub-Tarefa) Determinar Endereço de NW

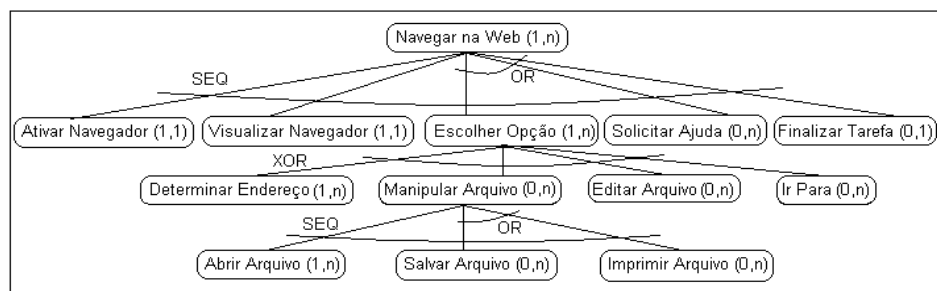


Figura B.4: Sub-Plano Manipular Arquivo de NW

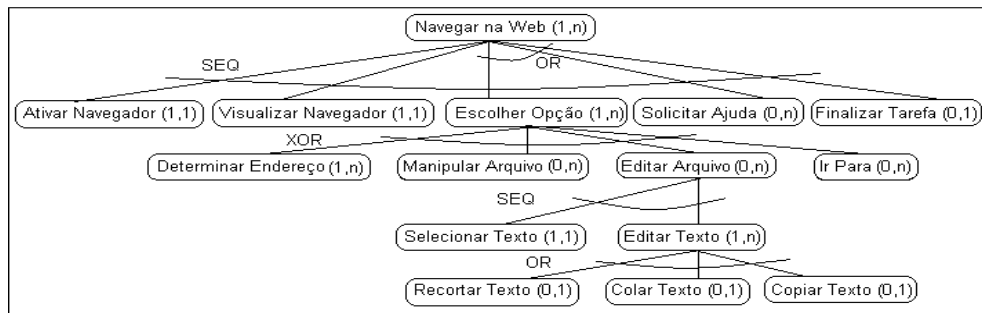


Figura B.5: Sub-Plano Editar Arquivo de NW

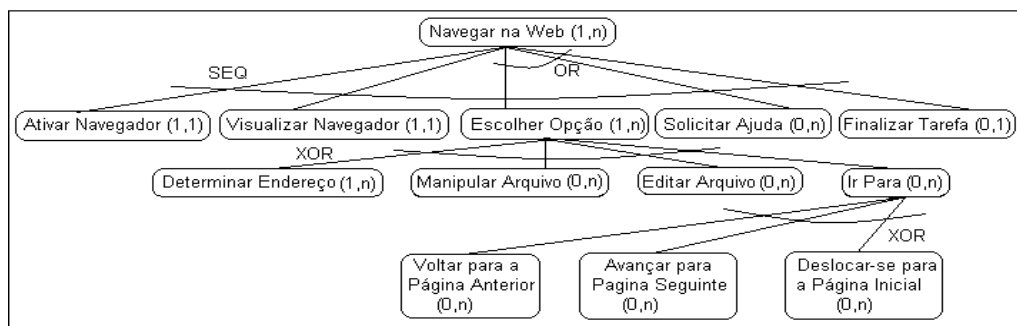


Figura B.6: Sub-Plano Ir Para de NW

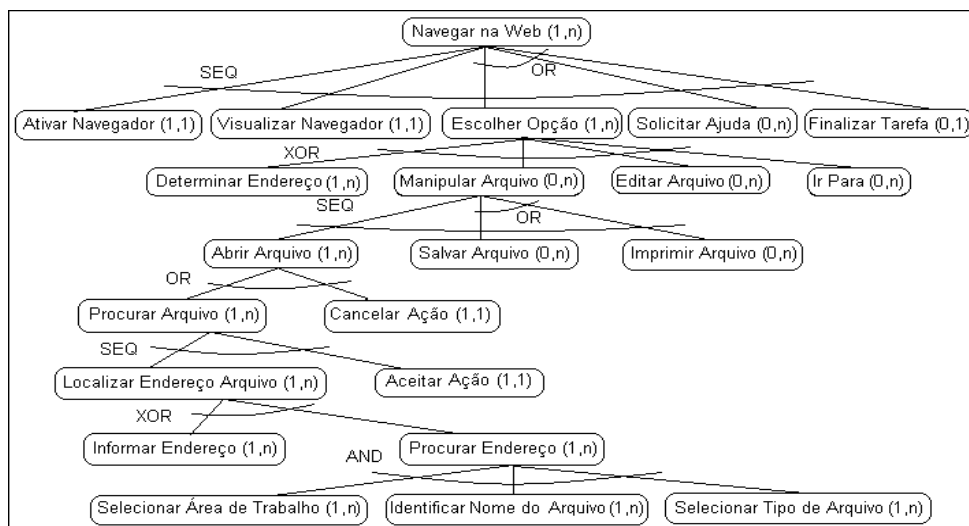


Figura B.7: Sub-Plano Abrir Arquivo de NW

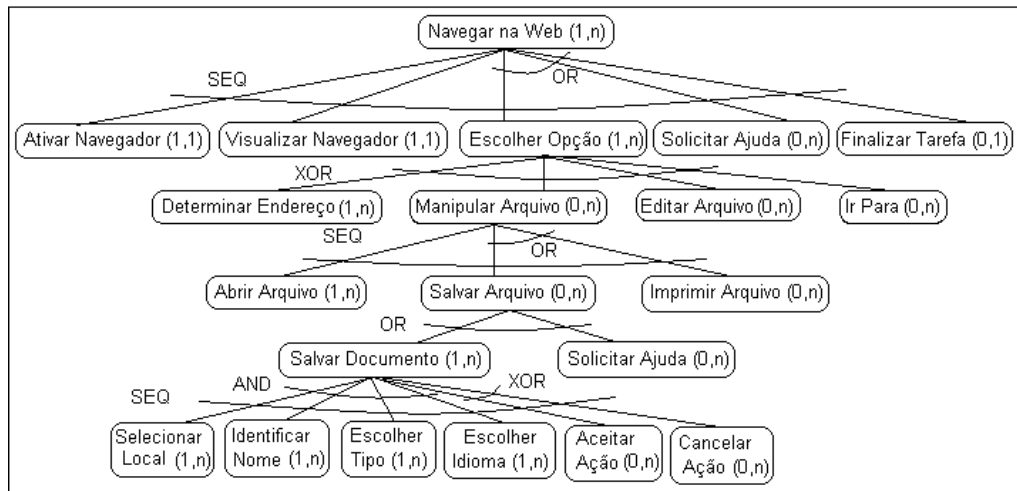


Figura B.8: Sub-Plano Salvar Arquivo de NW

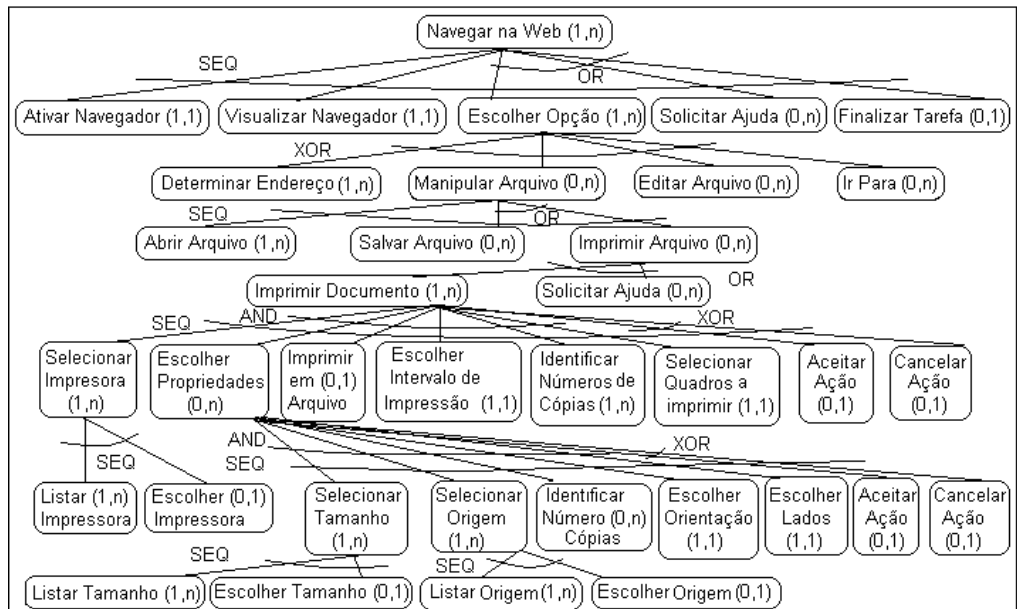


Figura B.9: Sub-Plano Imprimir Arquivo de NW

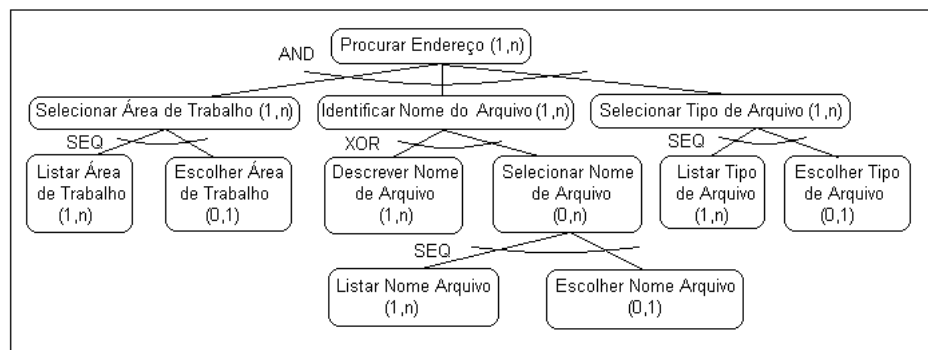


Figura B.10: Sub-Plano Procurar Endereço de NW

B.3 Definição de Cada Classe

Em seguida se ilustra a definição de cada classe da tarefa Navegar na Web a partir da definição das classes que modelam o domínio.

Classe	Plano
Nome	Navegar na Web
Descrição	Usuário navega na Web por meio de um navegador e busca informação
Pré-situação	Situação Inicial
Pós-situação	Situação Objetivo
Ocorrência	(1,n)
Ações	[Ativar Navegador, Visualizar Navegador, Finalizar Tarefa]
Sub-Planos	[Escolher Opção, Solicitar Ajuda]
Como-Realizar	Método de Navegar na Web

Descritor B.1: Classe Plano “Navegar na Web”

Classe	Situação
Nome	Situação Inicial
Descrição	Pré-situação ou estado do mundo antes da realização da tarefa
Objetos	[Usuário, Navegador, Tela]
Restrição	Visível Ícone(Navegador, Tela) AND Disponível(Usuário)
Como-atingir	[]

Descritor B.2: Classe Situação “Situação_Inicial”

Classe	Situação
Nome	Situação Objetivo
Descrição	Objetivo a atingir
Objetos	[Informação]
Restrição	Obtido(Informação)
Como-atingir	Navegar na Web

Descritor B.3: Classe Situação “Situação_Objetivo”

Classe	Método
Nome	Método de Navegar na Web
Descrição	Estratégia para navegar na web
Corpo	SEQ(Ativar Navegador, Visualizar Navegador, OR(Escolher Opção, Solicitar Ajuda, Finalizar Tarefa))

Descritor B.4: Classe Método “Método Navegar na Web”

Classe	Agente
Nome	Usuário
Descrição	Usuário que realiza a tarefa Navegar na Web
Competência	Navegar na Web

Descritor B.5: Classe Agente “Usuário”

Classe	Instrumento
Nome	Navegador
Descrição	Navegador utilizado pelo Usuário para navegar na web
Ator	Usuário
Emprego	Navegar na Web

Descritor B.6: Classe Instrumento “Navegador”

Classe	Ação
Nome	Ativar Navegador
Descrição	Inicializa navegador do Usuário
Pré-situação	Situação Inicial
Pós-situação	Navegador na Tela
Ocorrência	(1,1)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.7: Classe Ação “Ativar Navegador”

Classe	Ação
Nome	Visualizar Navegador
Descrição	Usuário visualiza o navegador na tela
Pré-situação	Navegador na Tela
Pós-situação	Navegador Visto
Ocorrência	While(Navegador na Tela)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.8: Classe Ação “Visualizar Navegador”

Classe	Situação
Nome	Navegador Visto
Descrição	Usuário olha navegador ativado na tela
Objetos	[Usuário, Navegador, Tela]
Restrição	Disponível(Navegador, Tela) AND Disponível(Usuário)
Como-atingir	Visualizar Navegador

Descritor B.9: Classe Pós-situação da Ação “Visualizar Navegador”

Classe	Plano
Nome	Escolher Opção
Descrição	Escolher Opção no Navegador
Pré-situação	Situação Inicial
Pós-situação	Pós-EscolherOpção
Ocorrência	(1,n)
Ações	[]
Sub-Planos	[Determinar Endereço, Manipular Arquivo, Editar Arquivo, Ir Para]
Como-Realizar	Método de Escolher Opção

Descritor B.10: Classe Plano “Escolher Opção”

Classe	Situação
Nome	Pós-EscolherOpção
Descrição	Pós-situação do Plano Escolher Opção
Objetos	[Endereço Navegador, Arquivo, Lista de Páginas]
Restrição	XOR(Determinado(Endereço), Manipulado(Arquivo), Editado(Arquivo), FoiPara(OutraPagina)
Como-atingir	Navegar na Web

Descritor B.11: Classe Pós-situação do Plano “Escolher Opção”

Classe	Método
Nome	Método de Escolher Opção
Descrição	Método de Escolher Opção do Navegador
Corpo	XOR (Determinar Endereço, Manipular Arquivo, Editar Arquivo, Ir Para)

Descritor B.12: Classe Método “Método Escolher Opção”

Classe	Plano
Nome	Determinar Endereço
Descrição	Determinar Endereço do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-DeterminarEndereço
Ocorrência	(1,0)
Ações	[]
Sub-Planos	[Informar Endereço, Selecionar Favoritos, Selecionar Histórico]
Como-Realizar	Método de Determinar Endereço

Descritor B.13: Classe Plano “Determinar Endereço”

Classe	Situação
Nome	Pós-DeterminarEndereço
Descrição	Pós-situação do Plano Determinar Endereço Navegador
Objetos	[Endereço,Favoritos,Histórico]
Restrição	XOR(Informado(Endereço), Selecionado(Favoritos), Selecionado(Histórico))
Como-atingir	Escolher Opção

Descritor B.14: Classe Pós-situação do Plano “Determinar Endereço”

Classe	Método
Nome	Método de Determinar Endereço
Descrição	Método de Determinar Endereço do Navegador
Corpo	XOR (Informar Endereço, Selecionar Favoritos, Selecionar Histórico)

Descritor B.15: Classe Método “Método Determinar Endereço”

Classe	Plano
Nome	Informar Endereço
Descrição	Informar Endereço do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-InformarEndereço
Ocorrência	(1,n)
Ações	[]
Sub-Planos	[Selecionar Endereço, Editar Endereço, Solicitar Ajuda]
Como-Realizar	Método de Informar Endereço

Descritor B.16: Classe Plano “Informar Endereço”

Classe	Situação
Nome	Pós-InformarEndereço
Descrição	Pós-situação do Plano Informar Endereço do Navegador
Objetos	[Endereço]
Restrição	OR(SEQ (Selecionado(Endereço),Editado(Endereço)), Solicitada(Ajuda))
Como-atingir	Determinar Endereço

Descritor B.17: Classe Pos-situação do Plano “Informar Endereço”

Classe	Método
Nome	Método de Informar Endereço
Descrição	Método de Informar Endereço do Navegador
Corpo	OR(SEQ (Selecionar Endereço, Editar Endereço), Solicitar Ajuda)

Descritor B.18: Classe Método “Método de Informar Endereço”

Classe	Plano
Nome	Selecionar Endereço
Descrição	Selecionar Endereço do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-SelecionarEndereço
Ocorrência	(0,n)
Ações	[Listar Endereço, Escolher Endereço]
Sub-Planos	[]
Como-Realizar	Método de Selecionar Endereço

Descritor B.19: Classe Plano “Selecionar Endereço”

Classe	Situação
Nome	Pós-SelecionarEndereço
Descrição	Pós-situação do Plano Selecionar Endereço
Objetos	[Endereço]
Restrição	SEQ (Listado(Endereço), Escolhido(Endereço))
Como-atingir	Informar Endereço

Descritor B.20: Classe Pós-situação do Plano “Selecionar Endereço”

Classe	Método
Nome	Método de Selecionar Endereço
Descrição	Método de Selecionar Endereço do Navegador
Corpo	SEQ (Listar Endereço, Escolher Endereço)

Descritor B.21: Classe Método “Método de Selecionar Endereço”

Classe	Ação
Nome	Listar Endereço
Descrição	Listar Endereço do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-ListarEndereço
Ocorrência	(1,n)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.22: Classe Ação “Listar Endereço”

Classe	Situação
Nome	Pós-ListarEndereço
Descrição	Pós-situação da Ação Listar Endereço Navegador
Objetos	[Endereço]
Restrição	Listado(Endereço)
Como-atingir	Selecionar Endereço

Descritor B.23: Classe Pós-situação da Ação “Listar Endereço”

Classe	Ação
Nome	Escolher Endereço
Descrição	Escolher Endereço do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-EscolherEndereço
Ocorrência	(0,1)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.24: Classe Ação “Escolher Endereço”

Classe	Situação
Nome	Pós-EscolherEndereço
Descrição	Pós-situação da Ação Escolher Endereço do Navegador
Objetos	[Endereço]
Restrição	Escolhido(Endereço)
Como-atingir	Selecionar Endereço

Descritor B.25: Classe Pós-situação da Ação “Escolher Endereço”

Classe	Plano
Nome	Editar Endereço
Descrição	Editar Endereço do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-EditarEndereço
Ocorrência	(0,n)
Ações	[Selecionar Texto Endereço]
Sub-Planos	[Editar Endereço]
Como-Realizar	Método de Editar Endereço

Descritor B.26: Classe Plano “Editar Endereço”

Classe	Situação
Nome	Pós-EditarEndereço
Descrição	Pós-situação do Plano Editar Endereço do Navegador
Objetos	[Texto Endereço]
Restrição	SEQ(Selecionado(Texto Endereço), Editado(Texto Endereço))
Como-atingir	Informar Endereço

Descritor B.27: Classe Pós-situação do Plano “Editar Endereço”

Classe	Método
Nome	Método de Editar Endereço
Descrição	Método de Editar Endereço do Navegador
Corpo	SEQ(Selecionar Texto Endereço, Editar Texto Endereço)

Descritor B.28: Classe Método “Método de Editar Endereço do Navegador”

Classe	Ação
Nome	Selecionar Texto Endereço
Descrição	Selecionar Texto Endereço do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-SelecionarTextoEndereço
Ocorrência	(1,1)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.29: Classe Ação “Selecionar Texto Endereço”

Classe	Situação
Nome	Pós-SelecionarTextoEndereço
Descrição	Pós-situação da Ação Selecionar Texto Endereço do Navegador
Objetos	[Texto Endereço]
Restrição	Selecionado(Texto Endereço)
Como-atingir	Editar Endereço

Descritor B.30: Classe Pós-situação da Ação “Selecionar Texto Endereço”

Classe	Plano
Nome	Editar Texto Endereço
Descrição	Editar Texto do Endereço do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-EditarTextoEndereço
Ocorrência	(1,n)
Ações	[Recortar Texto Endereço, Colar Texto Endereço, Copiar Texto Endereço]
Sub-Planos	[]
Como-Realizar	Método Editar Texto Endereço

Descritor B.31: Classe Plano “Editar Texto Endereço”

Classe	Situação
Nome	Pós-EditarTextoEndereço
Descrição	Pós-situação do Plano Editar Texto do Endereço do Navegador
Objetos	[Texto Endereço]
Restrição	OR(Recortado(Texto Endereço), Colado(Texto Endereço), Copiado(Texto Endereço))
Como-atingir	Editar Endereço

Descritor B.32: Classe Pós-situação do Plano “Editar Texto Endereço”

Classe	Método
Nome	Método Editar Texto Endereço
Descrição	Método Editar Texto Endereço do Navegador
Corpo	OR(Recortar Texto Endereço, Colar Texto Endereço, Copiar Texto Endereço)

Descritor B.33: Classe Método “Editar Texto Endereço”

Classe	Ação
Nome	Recortar Texto Endereço
Descrição	Recortar Texto Endereço do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-RecortarTextoEndereço
Ocorrência	(0,1)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.34: Classe Ação “Recortar Texto Endereço”

Classe	Situação
Nome	Pós-RecortarTextoEndereço
Descrição	Pós-situação da Ação Recortar Texto Endereço do Navegador
Objetos	[Texto Endereço]
Restrição	Recortado(Texto Endereço)
Como-atingir	Editar Texto Endereço

Descritor B.35: Classe Pós-situação da Ação “Recortar Texto Endereço”

Classe	Ação
Nome	Colar Texto Endereço
Descrição	Colar Texto Endereço do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-ColarTextoEndereço
Ocorrência	(0,1)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.36: Classe Ação “Colar Texto Endereço”

Classe	Situação
Nome	Pós-ColarTextoEndereço
Descrição	Pós-situação da Ação Colar Texto Endereço do Navegador
Objetos	[Texto Endereço]
Restrição	Colado(Texto Endereço)
Como-atingir	Editar Texto Endereço

Descritor B.37: Classe Pós-situação da Ação “Colar Texto Endereço”

Classe	Ação
Nome	Copiar Texto Endereço
Descrição	Copiar Texto Endereço do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-CopiarTextoEndereço
Ocorrência	(0,1)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.38: Classe Ação “Copiar Texto Endereço”

Classe	Situação
Nome	Pós-CopiarTextoEndereço
Descrição	Pós-situação da Ação Copiar Texto Endereço Navegador
Objetos	[Texto Endereço]
Restrição	Copiado(Texto Endereço)
Como-atingir	Editar Texto Endereço

Descritor B.39: Classe Pós-situação da Ação “Copiar Texto Endereço”

Classe	Plano
Nome	Selecionar Favoritos
Descrição	Selecionar Favoritos do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-SelecionarFavoritos
Ocorrência	(1,n)
Ações	[Listar Favoritos, Escolher Favoritos]
Sub-Planos	[Solicitar Ajuda]
Como-Realizar	Método de Selecionar Favoritos

Descritor B.40: Classe Plano “Selecionar Favoritos”

Classe	Situação
Nome	Pós-SelecionarFavoritos
Descrição	Pós-situação do Plano Selecionar Favoritos Navegador
Objetos	[Endereço]
Restrição	OR(SEQ (Listado(Favoritos), Escolhido(Favoritos)), Solicitada(Ajuda))
Como-atingir	Determinar Endereço

Descritor B.41: Classe Pós-situação do Plano “Selecionar Favoritos”

Classe	Método
Nome	Método de Selecionar Favoritos
Descrição	Método de Selecionar Favoritos Navegador
Corpo	OR(SEQ (Listar Favoritos, Escolher Favoritos), Solicitar Ajuda)

Descritor B.42: Classe Método “Método Selecionar Favoritos”

Classe	Ação
Nome	Listar Favoritos
Descrição	Listar Favoritos do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-ListarFavoritos
Ocorrência	(1,n)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.43: Classe Ação “Listar Favoritos”

Classe	Situação
Nome	Pós-ListarFavoritos
Descrição	Pós-situação da Ação Listar Favoritos do Navegador
Objetos	[Favoritos]
Restrição	Listado(Favoritos)
Como-atingir	Selecionar Favoritos

Descritor B.44: Classe Pós-situação da Ação “Listar Favoritos”

Classe	Ação
Nome	Escolher Favoritos
Descrição	Escolher Favoritos do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-EscolherFavoritos
Ocorrência	(0,1)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.45: Classe Ação “Escolher Favoritos do Navegador”

Classe	Situação
Nome	Pós-EscolherFavoritos
Descrição	Pós-situação da Ação Escolher Favoritos do Navegador
Objetos	[Favoritos]
Restrição	Escolhido(Favoritos)
Como-atingir	Selecionar Favoritos

Descritor B.46: Classe Pós-situação de “Escolher Favoritos do Navegador”

Classe	Plano
Nome	Selecionar Histórico
Descrição	Selecionar Histórico do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-SelecionarHistorico
Ocorrência	(1,n)
Ações	[Listar Histórico, Escolher Histórico]
Sub-Planos	[Solicitar Ajuda]
Como-Realizar	Método de Selecionar Histórico

Descritor B.47: Classe Plano “Selecionar Histórico do Navegador”

Classe	Situação
Nome	Pós-SelecionarHistorico
Descrição	Pós-situação do Plano Selecionar Histórico Navegador
Objetos	[Histórico, Ajuda]
Restrição	OR(SEQ (Listado(Histórico), Escolhido(Histórico)), Solicitada(Ajuda))
Como-atingir	Determinar Endereço

Descritor B.48: Classe Pós-situação do Plano “Selecionar Histórico do Navegador”

Classe	Método
Nome	Método de Selecionar Histórico
Descrição	Método de Selecionar Histórico Navegador
Corpo	OR(SEQ (Listar Histórico, Escolher Histórico), Solicitar Ajuda)

Descritor B.49: Classe Método “Método Selecionar Histórico”

Classe	Ação
Nome	Listar Histórico
Descrição	Listar Histórico do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-ListarHistorico
Ocorrência	(1,n)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.50: Classe Ação “Listar Histórico do Navegador”

Classe	Situação
Nome	Pós-ListarHistorico
Descrição	Pós-situação da Ação Listar Histórico do Navegador
Objetos	[Histórico]
Restrição	Listado(Histórico)
Como-atingir	Selecionar Histórico

Descritor B.51: Classe Pós-situação da Ação “Listar Histórico do Navegador”

Classe	Ação
Nome	Escolher Histórico
Descrição	Escolher Histórico do Navegador
Pré-situação	Situação Inicial
Pós-situação	Pós-EsHiBro
Ocorrência	(0,1)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.52: Classe Ação “Escolher Histórico”

Classe	Situação
Nome	Pós-EscolherHistorico
Descrição	Pós-situação da Ação Escolher Histórico do Navegador
Objetos	[Histórico]
Restrição	Escolhido(Histórico)
Como-atingir	Listar Histórico

Descritor B.53: Classe Pós-situação da Ação “Escolher Histórico”

Classe	Plano
Nome	Manipular Arquivo
Descrição	Manipular Arquivo do Navegador
Pré-situação	Situação Inicial
Pós-situação	Pós-ManipularArquivo
Ocorrência	(0,n)
Ações	[]
Sub-Planos	[Abrir Arquivo, Salvar Arquivo, Imprimir Arquivo]
Como-Realizar	Método de Manipular Arquivo

Descritor B.54: Classe Plano “Manipular Arquivo”

Classe	Situação
Nome	Pós-ManipularArquivo
Descrição	Pós-situação do Plano Manipular Arquivo do Navegador
Objetos	[Arquivo]
Restrição	SEQ(Aberto(Arquivo), OR(Salvado(Arquivo), Impresso Arquivo))
Como-atingir	Escolher Opção

Descritor B.55: Classe Pós-situação do Plano “Manipular Arquivo”

Classe	Método
Nome	Método de Manipular Arquivo
Descrição	Método de Manipular Arquivo do Navegador
Corpo	SEQ(Abrir Arquivo, OR(Salvar Arquivo, Imprimir Arquivo))

Descritor B.56: Classe Método “Método Manipular Arquivo do Navegador”

Classe	Plano
Nome	Abrir Arquivo
Descrição	Abrir Arquivo do Navegador
Pré-situação	Situação Inicial
Pós-situação	Pós-AbrirArquivo
Ocorrência	(1,n)
Ações	[Cancelar Ação]
Sub-Planos	[Procurar Arquivo]
Como-Realizar	Método de Abrir Arquivo

Descritor B.57: Classe Plano “Abrir Arquivo do Navegador”

Classe	Situação
Nome	Pós-AbrirArquivo
Descrição	Pós-situação do Plano Abrir Arquivo do Navegador
Objetos	[Arquivo, Ação]
Restrição	OR (Procurado(Arquivo), Cancelada(Ação))
Como-atingir	Manipular Arquivo

Descritor B.58: Classe Pós-situação do Plano “Abrir Arquivo”

Classe	Método
Nome	Método de Abrir Arquivo
Descrição	Método de Abrir Arquivo do Navegador
Corpo	OR(Procurar Arquivo, Cancelar Ação)

Descritor B.59: Classe Método “Método de Abrir Arquivo”

Classe	Plano
Nome	Procurar Arquivo
Descrição	Procurar Arquivo do Navegador
Pré-situação	Situação Inicial
Pós-situação	Pós-ProcurarArquivo
Ocorrência	(1,n)
Ações	[Aceitar Ação]
Sub-Planos	[Localizar Endereço Arquivo]
Como-Realizar	Método de Procurar Arquivo

Descritor B.60: Classe Plano “Procurar Arquivo do Navegador”

Classe	Situação
Nome	Pós-ProcurarArquivo
Descrição	Pós-situação do Plano Procurar Arquivo do Navegador
Objetos	[Endereço Arquivo, Ação]
Restrição	SEQ(Localizado(Endereço Arquivo), Aceitada(Ação))
Como-atingir	Abrir Arquivo

Descritor B.61: Classe Pós-situação do Plano “Procurar Arquivo”

Classe	Método
Nome	Método de Procurar Arquivo
Descrição	Método de Procurar Arquivo do Navegador
Corpo	SEQ(Localizar Endereço Arquivo, Aceitar Ação)

Descritor B.62: Classe Método “Método de Procurar Arquivo”

Classe	Plano
Nome	Localizar Endereço Arquivo
Descrição	Localizar Endereço Arquivo do Navegador
Pré-situação	Situação Inicial
Pós-situação	Pós-LocalizarEndereço
Ocorrência	(1,n)
Ações	[]
Sub-Planos	[Informar Endereço, Procurar Endereço]
Como-Realizar	Método de Localizar Endereço Arquivo

Descritor B.63: Classe Plano “Localizar Endereço Arquivo”

Classe	Situação
Nome	Pós-LocalizarEndereço
Descrição	Pós-situação do Plano Localizar Endereço Arquivo do Navegador
Objetos	[Endereço]
Restrição	XOR (Informado(Endereço), Procurado(Endereço))
Como-atingir	Procurar Arquivo

Descritor B.64: Classe Pós-situação do Plano “Localizar Endereço Arquivo”

Classe	Método
Nome	Método de Localizar Endereço Arquivo
Descrição	Método de Localizar Endereço Arquivo do Navegador
Corpo	XOR(Informar Endereço, Procurar Endereço)

Descritor B.65: Classe Método “Método Localizar Endereço Arquivo”

Classe	Plano
Nome	Procurar Endereço
Descrição	Procurar Endereço do Navegador
Pré-situação	Situação Inicial
Pós-situação	Pós-ProcurarEndereço
Ocorrência	(1,n)
Ações	[]
Sub-Planos	[Selecionar Área de Trabalho, Identificar Nome do Arquivo, Selecionar Tipo de Arquivo]
Como-Realizar	Método de Procurar Endereço

Descritor B.66: Classe Plano “Procurar Endereço do Navegador”

Classe	Situação
Nome	Pós-ProcurarEndereço
Descrição	Pós-situação do Plano Procurar Endereço do Navegador
Objetos	[Área de Trabalho, Nome de Arquivo, Tipo de Arquivo]
Restrição	AND (Selecionada(Área de Trabalho), Identificado(Nome de Arquivo), Selecionado(Tipo de Arquivo))
Como-atingir	Localizar Endereço Arquivo

Descritor B.67: Classe Pós-situação do Plano “Procurar Endereço”

Classe	Método
Nome	Método de Procurar Endereço
Descrição	Método de Procurar Endereço do Navegador
Corpo	AND(Selecionar Área de Trabalho, Identificar Nome do Arquivo, Selecionar Tipo de Arquivo)

Descritor B.68: Classe Método “Método de Procurar Endereço do Navegador”

Classe	Plano
Nome	Selecionar Área de Trabalho
Descrição	Selecionar Área de Trabalho do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-SelecionarAreaTrabalho
Ocorrência	(1,n)
Ações	[Listar Área de Trabalho, Escolher Área de Trabalho]
Sub-Planos	[]
Como-Realizar	Método de Selecionar Área de Trabalho

Descritor B.69: Classe Plano “Selecionar Área de Trabalho”

Classe	Situação
Nome	Pós-SelecionarAreaTrabalho
Descrição	Pós-situação do Plano Selecionar Área de Trabalho
Objetos	[Área de Trabalho]
Restrição	SEQ (Listada(Área de Trabalho), Escolhida(Área de Trabalho))
Como-atingir	Procurar Endereço

Descritor B.70: Classe Pós-situação do Plano “Selecionar Área de Trabalho”

Classe	Método
Nome	Método de Selecionar Área de Trabalho
Descrição	Método de Selecionar Área de Trabalho do Navegador
Corpo	SEQ(Listar Área de Trabalho, Escolher Area de Trabalho)

Descritor B.71: Classe Método “Método de Selecionar Área de Trabalho”

Classe	Ação
Nome	Listar Área de Trabalho
Descrição	Listar Área de Trabalho do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-ListarAreaTrabalho
Ocorrência	(1,n)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.72: Classe Ação “Listar Área de Trabalho do Navegador”

Classe	Situação
Nome	Pós-ListarAreaTrabalho
Descrição	Pós-situação da Ação Listar Área de Trabalho do Navegador
Objetos	[Área de Trabalho]
Restrição	Listado(Área de Trabalho)
Como-atingir	Selecionar Área de Trabalho

Descritor B.73: Classe Pós-situação da Ação “Listar Área de Trabalho”

Classe	Ação
Nome	Escolher Area de Trabalho
Descrição	Escolher Área de Trabalho do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-EscolherAreaTrabalho
Ocorrência	(0,1)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.74: Classe Ação “Escolher Área de Trabalho”

Classe	Situação
Nome	Pós-EscolherAreaTrabalho
Descrição	Pós-situação da Ação Escolher Área de Trabalho do Navegador
Objetos	[Área de Trabalho]
Restrição	Escolhido(Área de Trabalho)
Como-atingir	Listar Área de Trabalho

Descritor B.75: Classe Pós-situação da Ação “Escolher Área de Trabalho”

Classe	Plano
Nome	Identificar Nome de Arquivo
Descrição	Identificar Nome de Arquivo do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-IdentificarNomeArquivo
Ocorrência	(1,n)
Ações	[Descrever Nome de Arquivo, Selecionar Nome de Arquivo]
Sub-Planos	[]
Como-Realizar	Método de Identificar Nome de Arquivo

Descritor B.76: Classe Plano “Identificar Nome de Arquivo”

Classe	Situação
Nome	Pós-IdentificarNomeArquivo
Descrição	Pós-situação do Plano Identificar Nome de Arquivo
Objetos	[Nome do Arquivo]
Restrição	SEQ (Descrito(Nome de Arquivo), Selecionado(Nome de Arquivo))
Como-atingir	Procurar Endereço

Descritor B.77: Classe Plano “Identificar Nome do Arquivo”

Classe	Método
Nome	Método de Identificar Nome do Arquivo
Descrição	Método de Identificar Nome do Arquivo do Navegador
Corpo	XOR(Descrever Nome de Arquivo, Selecionar Nome de Arquivo)

Descritor B.78: Classe Método “Método de Identificar Nome do Arquivo”

Classe	Ação
Nome	Descrever Nome de Arquivo
Descrição	Descrever Nome de Arquivo do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-DescreverNomeArquivo
Ocorrência	(1,n)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.79: Classe Ação “Descrever Nome de Arquivo”

Classe	Situação
Nome	Pós-DescreverNomeArquivo
Descrição	Pós-situação da Ação Descrever Nome de Arquivo do Navegador
Objetos	[Nome de Arquivo]
Restrição	Descrito(Nome de Arquivo)
Como-atingir	Identificar Nome do Arquivo

Descritor B.80: Classe Pós-situação da Ação “Descrever Nome do Arquivo”

Classe	Plano
Nome	Selecionar Nome de Arquivo
Descrição	Selecionar Nome de Arquivo do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-SelecionarNomeArquivo
Ocorrência	(0,n)
Ações	[Listar Nome de Arquivo, Escolher Nome de Arquivo]
Sub-Planos	[]
Como-Realizar	Método de Selecionar Nome de Arquivo

Descritor B.81: Classe Plano “Selecionar Nome de Arquivo do Navegador”

Classe	Situação
Nome	Pós-SelecionarNomeArquivo
Descrição	Pós-situação do Plano Selecionar Nome de Arquivo
Objetos	[Nome de Arquivo]
Restrição	SEQ (Listado(Nome de Arquivo), Escolhido(Nome de Arquivo))
Como-atingir	Identificar Nome do Arquivo

Descritor B.82: Classe Pós-situação do Plano “Selecionar Nome de Arquivo”

Classe	Método
Nome	Método de Selecionar Nome de Arquivo
Descrição	Método de Selecionar Nome de Arquivo do Navegador
Corpo	SEQ(Listar Nome de Arquivo, Escolher Nome de Arquivo)

Descritor B.83: Classe Metodo “Método de Selecionar Nome de Arquivo”

Classe	Ação
Nome	Listar Nome de Arquivo
Descrição	Listar Nome de Arquivo do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-ListarNomeArquivo
Ocorrência	(1,n)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.84: Classe Ação “Listar Nome de Arquivo”

Classe	Situação
Nome	Pós-ListarNomeArquivo
Descrição	Pós-situação da Ação Listar Nome de Arquivo do Navegador
Objetos	[Nome de Arquivo]
Restrição	Listado(Nome de Arquivo)
Como-atingir	Selecionar Nome de Arquivo

Descritor B.85: Classe Pós-situação da Ação “Listar Nome de Arquivo”

Classe	Ação
Nome	Escolher Nome de Arquivo
Descrição	Escolher Nome de Arquivo do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-EscolherNomeArquivo
Ocorrência	(0,1)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.86: Classe Ação “Escolher Nome de Arquivo”

Classe	Situação
Nome	Pós-EscolherNomeArquivo
Descrição	Pós-situação da Ação Escolher Nome de Arquivo do Navegador
Objetos	[Nome de Arquivo]
Restrição	Escolhido(Nome de Arquivo)
Como-atingir	Listar Nome de Arquivo

Descritor B.87: Classe Pós-situação da Ação “Escolher Nome de Arquivo”

Classe	Plano
Nome	Selecionar Tipo de Arquivo
Descrição	Selecionar Tipo de Arquivo do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-SelecionarTipoArquivo
Ocorrência	(1,n)
Ações	[Listar Tipo de Arquivo, Escolher Tipo de Arquivo]
Sub-Planos	[]
Como-Realizar	Método de Selecionar Tipo de Arquivo

Descritor B.88: Classe Plano “Selecionar Tipo de Arquivo”

Classe	Situação
Nome	Pós-SelecionarTipoArquivo
Descrição	Pós-situação do Plano Selecionar Tipo de Arquivo
Objetos	[Tipo de Arquivo]
Restrição	SEQ (Listado(Tipo de Arquivo), Escolhido(Tipo de Arquivo))
Como-atingir	Procurar Endereço

Descritor B.89: Classe Plano “Selecionar Tipo de Arquivo”

Classe	Método
Nome	Método de Selecionar Tipo de Arquivo
Descrição	Método de Selecionar Tipo de Arquivo do Navegador
Corpo	SEQ(Listar Tipo de Arquivo, Escolher Tipo de Arquivo)

Descritor B.90: Classe Método “Método de Selecionar Tipo de Arquivo”

Classe	Ação
Nome	Listar Tipo de Arquivo
Descrição	Listar Tipo de Arquivo do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-ListarTipoArquivo
Ocorrência	(1,n)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.91: Classe Ação “Listar Tipo de Arquivo do Navegador”

Classe	Situação
Nome	Pós-ListarTipoArquivo
Descrição	Pós-situação da Ação Listar Tipo de Arquivo do Navegador
Objetos	[Tipo de Arquivo]
Restrição	Listado(Tipo de Arquivo)
Como-atingir	Selecionar Tipo de Arquivo

Descritor B.92: Classe Pós-situação da Ação “Listar Tipo de Arquivo do Navegador”

Classe	Ação
Nome	Escolher Tipo de Arquivo
Descrição	Escolher Tipo de Arquivo do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-EscolherTipoArquivo
Ocorrência	(0,1)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.93: Classe Ação “Escolher Tipo de Arquivo”

Classe	Situação
Nome	Pós-EscolherTipoArquivo
Descrição	Pós-situação da Ação Escolher Tipo de Arquivo do Navegador
Objetos	[Tipo de Arquivo]
Restrição	Escolhido(Tipo de Arquivo)
Como-atingir	Listar Tipo de Arquivo

Descritor B.94: Classe Pós-situação da Ação “Escolher Tipo de Arquivo”

Classe	Plano
Nome	Salvar Arquivo
Descrição	Salvar Arquivo do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-SalvarArquivo
Ocorrência	(0,n)
Ações	[]
Sub-Planos	[Salvar Documento, Solicitar Ajuda]
Como-Realizar	Método de Salvar Arquivo

Descritor B.95: Classe Plano “Salvar Arquivo”

Classe	Situação
Nome	Pós-SalvarArquivo
Descrição	Pós-situação do Plano Salvar Arquivo do Navegador
Objetos	[Documento, Ajuda]
Restrição	OR (Salvado(Documento), Solicitada(Ajuda))
Como-atingir	Manipular Arquivo

Descritor B.96: Classe Pós-situação do Plano “Salvar Arquivo”

Classe	Método
Nome	Método de Salvar Arquivo
Descrição	Método de Salvar Arquivo do Navegador
Corpo	OR(Salvar Documento, Solicitar Ajuda)

Descritor B.97: Classe Método “Método de Salvar Arquivo”

Classe	Plano
Nome	Imprimir Arquivo
Descrição	Imprimir Arquivo do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-ImprimirArquivo
Ocorrência	(0,n)
Ações	[]
Sub-Planos	[Imprimir Documento, Solicitar Ajuda]
Como-Realizar	Método de Imprimir Arquivo

Descritor B.98: Classe “Plano Imprimir Arquivo”

Classe	Situação
Nome	Pós-ImprimirArquivo
Descrição	Pós-situação do Plano Imprimir Arquivo do Navegador
Objetos	[Documento, Ajuda]
Restrição	OR (Impresso(Documento), Solicitada(Ajuda))
Como-atingir	Manipular Arquivo

Descritor B.99: Classe Pós-situação do Plano “Imprimir Arquivo”

Classe	Método
Nome	Método de Imprimir Arquivo
Descrição	Método de Imprimir Arquivo do Navegador
Corpo	OR(Imprimir Documento, Solicitar Ajuda)

Descritor B.100: Classe Método “Método de Imprimir Arquivo”

Classe	Plano
Nome	Editar Arquivo
Descrição	Editar Arquivo do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-EditarArquivo
Ocorrência	(0,n)
Ações	[Selecionar Texto, Recortar Texto, Copiar Texto, Colar Texto]
Sub-Planos	[]
Como-Realizar	Método de Editar Arquivo

Descritor B.101: Classe Plano “Editar Arquivo”

Classe	Situação
Nome	Pós-EditarArquivo
Descrição	Pós-situação do Plano Editar Arquivo do Navegador
Objetos	[Texto]
Restrição	SEQ(Selecionado(Texto), OR(Recortado(Texto), Copiado(Texto), Colado(Texto)))
Como-atingir	Escolher Opção

Descritor B.102: Classe Pós-situação do Plano “Editar Arquivo”

Classe	Método
Nome	Método de Editar Arquivo
Descrição	Método de Editar Arquivo do Navegador
Corpo	SEQ(Selecionar Texto, OR(Recortar Texto, Copiar Texto, Colar Texto))

Descritor B.103: Classe Método “Método de Editar Arquivo”

Classe	Plano
Nome	Ir Para
Descrição	Ir Para do Navegador
Pré-situação	Situação_Inicial
Pós-situação	Pós-IrPara
Ocorrência	(0,n)
Ações	[Voltar para a Página Anterior, Avançar para Página Seguinte, Deslocar-se para a Página Inicial]
Sub-Planos	[]
Como-Realizar	Método de Ir Para

Descritor B.104: Classe Plano “Ir Para”

Classe	Situação
Nome	Pós-IrPara
Descrição	Pós-situação do Plano Ir Para do Navegador
Objetos	[Lista de Páginas]
Restrição	XOR(Voltou(para a Página Anterior), Avançou(para Página Seguinte), Deslocou-se para a Página Inicial)
Como-atingir	Escolher Opção

Descritor B.105: Classe Pós-situação do Plano “Ir Para”

Classe	Método
Nome	Método de Ir Para
Descrição	Método de Ir Para do Navegador
Corpo	XOR(Voltar para a Página Anterior, Avançar para Página Seguinte, Deslocar-se para a Página Inicial)

Descritor B.106: Classe Método “Método de Ir Para”

Classe	Plano
Nome	Solicitar Ajuda
Descrição	Solicitar Ajuda do Navegador
Ocorrência	(0,n)
Pré-situação	Situação Inicial
Pós-situação	Pós-SolicitarAjuda
Ações	Exibir Texto
Sub-Planos	[Pesquisar Palavra-Chave ,Selecionar Tópicos, Selecionar Índice]
Como-Realizar	[Método de Exibir Texto, Método de Pesquisar Palavra-Chave , Método de Selecionar Tópicos, Método de Selecionar Índice]

Descritor B.107: Classe Plano “Solicitar Ajuda”

Classe	Situação
Nome	Pós- SolicitarAjuda
Descrição	Pós-situação do Plano Solicitar Ajuda do Navegador
Objetos	[Texto, Palavra-Chave, Tópicos, Índice]
Restrição	XOR(Exibido(Texto), Pesquisada(Palavra-Chave), Selecionado(Tópicos), Selecionado(Índice))
Como-atingir	Navegar na Web

Descritor B.108: Classe Pós-situação do Plano “Solicitar Ajuda”

Classe	Método
Nome	Método de Exibir Texto
Descrição	Método de Exibir Texto do Navegador
Corpo	Exibir Texto

Descritor B.109: Classe Método “Método Exibir Texto”

Classe	Método
Nome	Método de Pesquisar Palavra-Chave
Descrição	Método de Pesquisar Palavra-Chave do Navegador
Corpo	Pesquisar Palavra-Chave

Descritor B.110: Classe Método “Método de Pesquisar Palavra-Chave”

Classe	Método
Nome	Método de Selecionar Tópicos
Descrição	Método de Selecionar Tópicos do Navegador
Corpo	Selecionar Tópicos

Descritor B.111: Classe Método “Método de Selecionar Tópicos”

Classe	Método
Nome	Método de Selecionar Índice
Descrição	Método de Selecionar Índice do Navegador
Corpo	Selecionar Índice

Descritor B.112: Classe Método “Método de Selecionar Índice”

Classe	Ação
Nome	Finalizar Tarefa
Descrição	Finalizar Tarefa do Usuário
Pré-situação	Situação_Inicial
Pós-situação	Pós-FinalizarTarefa
Ocorrência	(1,1)
Agente	Usuário
Instrumento	Navegador
Status	Não automática

Descritor B.113: Classe Ação “Finalizar Tarefa”

Classe	Situação
Nome	Pós-FinalizarTarefa
Descrição	Pós- situação da Ação Finalizar Tarefa
Objetos	[Usuário, Navegador]
Restrição	TarefaFinalizada(Usuário, Navegador)
Como-atingir	Navegar na Web

Descritor B.114: Classe Pós-situação da Ação “Finalizar Tarefa”

Referências Bibliográficas

[Bisseret 1979]

Bisseret, A., *Introductory Elements to Ergonomics Research in Man-machine Systems*, Trends in Man-Machine Communications (Ed. Lienard, J.S.), Rocquencourt: INRIA, 1979.

[Card 1983]

Card S., Moran T. e Newell A., *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates, Publ., 1983.

[Cibys 1996]

Cibys W. A., *Ergonomia e Usabilidade de Software*, Labutil, Universidade Federal de Santa Catarina, Florianópolis, Brasil, 1996.

[Falzon 1989]

Falzon P., *Ergonomie Cognitive du Dialogue*, Presses Universitaires de Grenoble, 1989.

[Gamboa 1997]

Gamboa F. e 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.

[Graesser 1980]

Graesser A. C., Kowalsky D. J. e Smith D. A.. Memory for Typical and Atypical Actions in Scripted Activities, *Journal, Exper, Psychology: Human Learning and Memory*, vol. 6, 5, pp., 503-515.

[Green 1985]

Green M. W., *The Design of User Graphical Interfaces*, Ph. D. Thesis., Tech. Report CSRI-70, Computer Systems Research Institute, University of Toronto, Canada, April 1995.

[Haan 1992]

de 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.

[Haan 1999]

de Haan G., All Your Ever Wanted to Know about Extended Task-Action Grammar but Were Afraid to Ask, <http://home.worldonline.nl/~gdehaan/etag.html>, 1999.

[Hammouche 1993]

Hammouche H., *De la Modélisation des Tâches à la Spécification d'Interfaces Utilisateur*, Thèse de Docteur. Centre d'Estudes de la Navigation Aérienne et Institut National de Recherche en Informatique et en Automatique INRIA, France, 1993.

[Heemann 1997]

Heemann V., *Curso de Ergonomia em Sistemas de Informação*, Campina Grande, Julho de 1997.

[Hix 1993]

Hix D. e Hartson H. R., *Developing User Interfaces*, John Wiley & Sons, Inc., U.S.A., 1993.

[Johnson 1988]

Johnson P., Johnson H., Waddington R. e Shouls A. Task-Related Knowledge Structures: *Analysis, Modelling and Application*, Queen Mary College, University of London. 1988.

[Johnson 1991]

Johnson H. e Johnson P., *Task Knowledge Structures: Psychological Basis and Integration Into System Design*, Acta Psychologica 78, pp 3-26. Department of Computer Science, Queen Mary College, University of London. 1991.

[Johnson-Laird 1981]

Johnson-Laird P. N., *Mental Models in Cognitive Science*, Perspectives on Cognitive Science (Ed. Norman, D.A), Ablex, pp. 147-192. 1981.

[Kessel 1995]

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.

[Kieras 1984]

Kieras D.E. e Bovair S., *The Role of a Mental Model in Learning to Operate a Device*, Cognitive Science, vol. 8, No. 3. 1984.

[Kieras 1985]

Kieras D. E. e Polson P.G., *An Approach to the Formal Analysis of User Complexity*, Int. Journal of Man-Machine Studies vol. 22, pp. 365-394, 1985.

[Markopoulos 1992]

Markopoulos P., Pycock J., Wilson S. and Johnson P., *Adept – A Task Based Design Environment*, Queen Mary and Westfield College, 1992.

[Markopoulos 1997]

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.

[Medeiros 1995a]

Medeiros H. e Rousselot F., *Acquisition et Modélisation de Concepts Dynamiques: Le Système TAME*, Rapport ERIC R0102-96, Strasbourg, França, 1995.

[Medeiros 1995b]

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.

[Medeiros 1995c]

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.

[Minsky 1975]

Minsky M., *A Framework for Representing Knowledge*, The Psychology of Computer Vision. Winston, P., (ed), pp. 211-277. McGraw-Hill, New York, 1975.

[Moran 1981]

Moran T., *The Command Language Grammar*, A Representation for the User Interface of Interactive Computer Systems, International Journal of Man-Machine Studies, Vol 15, pp. 3-50, 1981.

[Nanard 1990]

Nanard J., *La Manipulation Directe en Interface Homme-Machine*, Thèse de Docteur, Université Montpellier II, France, 1990.

[Newell 1984]

Newell G. E., *Learning from Writing in Two Content Areas: A Case Study of Protocol Analysis*, Research in the Teaching of English, 18,3, Oct 84, pp. 265-287.

[Nielsen 1986]

Nielsen J., *A Virtual Protocol for Computer-Human Interaction*, Int. Journal of Man-Machine Studies, 24, pp.301-312. 1986.

[Nielsen 1990]

Nielsen J., *A Meta-Model for Interacting with Computers*, Interacting with Computers 2, 2, pp. 147-160.

[Norman 1983]

Norman D. A., *Some Observations on Mental Models. Mental Models*, 7-14. Editado por Dedre Gentner & Albert L. Stevens, Lawrence Erlbaum Associates, Hillsdale, New Jersey. 1983.

[Norman 1986]

Norman D. A. e Drapper S. W., *User Centered System Design*, Lawrence Erlbaum Associates, Publishers. 1986.

[Pierret 1989]

Pierret-Golbreich C., Delouis I. e Scapin D. L., *Un Outil D'Acquisition et de Representation des Taches Oriente-Objet*, Unité de Recherche, Inria-Rocquencourt, France 1989

[Rasmussen 1980]

Rasmussen, J., *The Human as a Systems Component*, Human Interaction with Computers (Ed. Smith, H.T. et Green, T.R.G.), Academic Press, 67-96. 1980.

[Richard 1990]

Richard J. F., *Les Activités Mentales, Comprendre, Raisonner, Trouver des Solutions*, Paris, A. Colin. 1990.

[Sacerdoti 1974a]

Sacerdoti E. D., *Planning in a Hierarchy of Abstraction Spaces*, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California, U.S.A., 1974.

[Sacerdoti 1974b]

Sacerdoti E. D., Informação referenciada de [http://www-cs-students.stanford.edu/~pdoyle/quailuail/notes/pdoyle/search.html#ABSTRIP S](http://www-cs-students.stanford.edu/~pdoyle/quailuail/notes/pdoyle/search.html#ABSTRIP_S), 1974.

[Sacerdoti 1975]

Sacerdoti E. D., *The Nonlinear Nature of Plans*, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California, U.S.A., 1975.

[Scapin 1987]

Scapin D. L., *Guide Ergonomique de Conception des Interfaces Homme-Machine*, Rapport INRIA No. 77, France, 1987.

[Scapin 1988]

Scapin D. L., *Vers des Outils Formels de Description des Taches Orientes Conception D'Interfaces*, Unité de Recherche, INRIA, Rocquencourt, France, 1988.

[Scapin 1989]

Scapin D. L. e Christine Pierret-Golbreich, *Towards a Method for Task Description: MAD*, Unité de Recherche, INRIA, Rocquencourt, France, 1989.

[Sebillote 1987]

Sebillote S., *La Coception de Scénarios Interactifs: Analyse de L'Activité*, Le Travail Humain, Tome 50, No. 4. Projet de Psychologie Ergonomique pour L'Informatique, INRIA, Domaine de Voluceau, Rocquencourt, Le Chesnay Cedex, France 1987.

[Sebillote 1988a]

Sebillote S., *Théories et Methodologies – Décrire des Taches Selon les Objectifs des opérateurs, De l'Interview a la Formalization*, Unité de Recherche, INRIA, Rocquencourt, France, 1988.

[Sebillote 1988b]

Sebillote S., *Hierarchical Planning as Method for Task Analysis: the example of office task analysis*, Behaviour and Information Technology, 1988, vol. 7, No. 3, pp. 275-293. INRIA, Rocquencourt, Le Chesnay-Cedex, France, 1988.

[Sebillote 1991]

Sebillote S., *Théories et Méthodologies, Décrire des Taches Selon les Objectives des Opérateurs, de l'Interview a la Formalisation*, Projet de Psychologie Ergonomique pour l'Informatique, INRIA, Domaine de Voluceau, Rocquencourt, Le Chesnay Cedex, France, 1991.

[Sebillote 1994a]

Sebillote S. e Alonso B., *Description MAD de la tâche de "contrôle aérien" exécutée par deux contrôleurs*, Projet de Psychologie Ergonomique pour l'Informatique, INRIA, Rocquencourt, Août, 1994.

[Sebillote 1994b]

Sebillote S., Alonso B., Fallah D., Hamouche H., Scapin D.L., e Varnild E., *Note de Recherche Concernant le Formalisme MAD*, Projet de Psychologie Ergonomique pour l'Informatique, INRIA, Ronquencourt, Novembre, 1994.

[Sebillote 1994c]

Sebillote S., *Méthodologie Pratique d'Analyse de la Tâche en Vue de l'Extraction de Caractéristiques Pertinentes pour la Conception d'Interfaces*, Rapport de Recherche No. 163, INRIA, Mars, 1994.

[Sebillote 1994d]

Sebillote S. e Fallah D., *Description MAD d'une tâche complexe: "Resource un incendie en mer"*, Rapport de Contrat, Étude réalisée dans le cadre d'un projet ESPRIT:INTUITIVE P6593, INRIA, Roquencourt, Mai, 1994.

[Sebillote 1994e]

Sebillote S. e Scapin D. L., *From Users' Task Knowledge to High-Level Interface Specification*, International Journal of Human-Computer Interaction 6(1) pp. 1-15, INRIA, Ronquencourt, Le Chesnay, Cedex, France, 1994.

[Sebillote 1995b]

Sebillote S., *Task TD-5: Methodology Guide to Task Analysis with the Goal of Extracting Relevant Characteristics for Interfaces*, Esprit 3 Projet: P6593, "INTUITIVE", INRIA Ronquencourt, April 1995.

[Senach 1990]

Senach B., *Evaluation de l'ergonomie des interfaces homme-machine: Une Revue de la Littérature*, Rapport de Recherche, INRIA, No. 1180, Mars, 1990.

[Shneiderman 1998]

Shneiderman B., *Designing the User Interface – Strategies for effective Humman-Computer Interaction*, Addison Wesley Publishing Co., 3rd Edition, 1998.

[Silva 1999]

da Silva J. C., *Aquisição de Conhecimentos e Manutenção para uma sociedade de Agentes Tutores Artificiais*, Dissertação de mestrado, Universidade Federal da Paraíba, COPIN, 1999.

[Wilson 1992]

Wilson S., Johnson P., Kelly C., Cunningham .J e Markopoulos P., *Beyond Hacking: A Model Based Approach to User Interface Design*, Department of Computer Science, Queen Mary and Westfield College, University of London, Inglaterra, 1992.

[Young 1981]

Young, R.M. *The Machine Inside the Machine: User's Models of Pocket Calculators*, Int. Journal Man-Machine Studies, vol 15, 51-85, 1981.

Bibliografia

[Bastien 1992]

Bastien J.M.C e Scapin D. L., *A Validation of Ergonomic Criteria for the Evaluation of Human – Computer Interfaces*, INRIA, France, 1992.

[Bastien 1995]

Bastien J.M.C. e Scapin D. L., *Evaluating a User Interface With Ergonomic Criteria*, INRIA, France, 1995.

[Coutaz 1990]

Coutaz J., *Interfaces Homme – Ordinateur*, Bordas, Paris, France, 1990.

[Cox 1993]

Cox K. e Walker D., *User-Interface*, Prentice Hall. Singapore 1993.

[Guinn 1998]

Guinn C. I., *An Analysis of Initiative Selection in Collaborative Task-Oriented Discourse*, *User Modeling and User-Adapted Interaction Vol 8*, pp. 255-314, Kluwer Academic Publishers, Netherlands, 1998.

[Hackos 1998]

Hackos J. T. e Redish J. C., *User and Task Analysis for Interface*, John Wiley and Sons, Inc., 1998.

[Jakobsson 1999]

Jakobsson M., Analysis and Design of User Interfaces. <http://www.uwasa.fi/~mj/hci/hci7.html>, Department of Information Technology and Production Economics, Faculty of Industrial Management, University of Vaasa, Finland 1999.

[Kolski 1993]

Kolski C., *Ingénierie des Interfaces Homme-Machine*, Editions Hermès, Paris 1993.

[Lula 1992]

Lula B., *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.

[Moço 1996]

Moço S., *O Uso de Cenários como uma Técnica de Apoio para Avaliações Ergonômicas de Softwares Interativos*, Dissertação de mestrado. Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção, Florianópolis, 1996.

[Norton 1991]

Norton P. e Heid J., *Desvendando o Macintosh*, Editora Campus, Rio de Janeiro, Brasil, 1991.

[Normand 1992]

Normand V., *Le Modèle SIROCO: de la Spécification Conceptuelle des Interfaces Utilisateur à leur Réalisation*, Thèse de Docteur. Université Joseph Fourier, Grenoble I, França, 1992.

[Pierret 1990]

Pierret-Golbreich C. e Delouis I., *Task Centered Representation for Expert Systems at the Knowledge Level*, Laboratoire de Recherche en Informatique, Equipe Intelligence Artificielle et Systèmes d'Inférences, Université Paris Sud, Orsay Cedex, France 1990.

[Preece 1993]

Preece J., *A Guide to Usability*, Addison-Wesley Publishing Company, Great Britain, 1993.

[Queiroz 1998]

de Queiroz J. E. R. e Turnell. M. F. Q. V., *Avaliando a Avaliação: Um SIG como Estudo de Caso*, Relatório Técnico RT00170 COPELE/CCT/UFPB, Campina Grande, Julho, 1998.

[Queiroz 1999]

de Queiroz J. E. R., *Estudo da Relação entre a Usabilidade de Interfaces Computacionais e Fatores de Avaliação*, (Exame de Qualificação), COPELE/CCT/UFPB, Campina Grande, Janeiro, 1999.

[Redmond 1995]

David Redmond-Pyle e Alan Moore, *Graphical User Interface Design and Evaluation*, Prentice Hall, 1995.

[Sebillote 1986]

Sebillote S.. *La Planification Hiérarchique comme Méthode d'Analyse de la Tâche*, Analyse de Taches de Bureau, Research Report 599, INRIA, Rocquencourt, Le Chesnay, France, 1986.

[Sebillote 1995c]

Sebillote S. e Fallah D., *Task TD-2: Task Description "Resolve the Emergency Situation: Fire"*, Raport de Contrat, Esprit 3 Projet: P6593. "INTUITIVE", INRIA Ronquencourt, April, 1995.

[Siochi 1989]

Siochi A. C. e Hartson H. R., *Task-Oriented Representation of Asynchronous User Interfaces*, Department of Computer Science, Chi'89 Proceedings, Virginia, U.S.A., May , 1989.

[Treu 1994]

Treu S., *User Interface Design*, University of Pittsburgh, Plenum Press, New York 1994.

[Turnell 1998]

Turnell M.F.Q.V., *Notas de Aula – Conceitos e Projeto de Interfaces Usuário – Computador*, Laboratório de Interfaces Homem – Máquina, DEE – CCT/UFPB, 1998.

[Waern 1990]

Waern Y., *On the Dynamics of Mental Models*, *Mental Models and Human-Computer Interaction 1*, Editado por D. Ackermann e M.J. Tauber (Editors), Elsevier Science Publishers B. V. North-Holland, 1990.