

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

CENTRO DE CIÊNCIAS E TECNOLOGIA

COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

DISSERTAÇÃO DE MESTRADO

PROPOSTA DE SUPORTE COMPUTACIONAL AO MCI

DANIEL SCHERER

**CAMPINA GRANDE
FEVEREIRO - 2004**

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

CENTRO DE CIÊNCIAS E TECNOLOGIA

COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

PROPOSTA DE SUPORTE COMPUTACIONAL AO MCI

Dissertação submetida à Coordenação de Pós-Graduação em Informática do Centro de Ciências e Tecnologia da Universidade Federal de Campina Grande como requisito parcial para obtenção do grau de mestre em informática (MSc).

DANIEL SCHERER

PROF^A MARIA DE FÁTIMA QUEIROZ VIEIRA TURNELL, PHD.

PROF^A FRANCILENE PROCÓPIO GARCIA, DSc.

(ORIENTADORAS)

ÁREA DE CONCENTRAÇÃO: CIÊNCIA DA COMPUTAÇÃO

LINHA DE PESQUISA: ENGENHARIA DE SOFTWARE

CAMPINA GRANDE - PARAÍBA

FEVEREIRO DE 2004

SCHERER, Daniel

S326P

Proposta de Suporte Computacional ao MCI

Dissertação (mestrado), Universidade Federal de Campina Grande, Centro de Ciências e Tecnologia, Coordenação de Pós-Graduação em Informática, Campina Grande - Paraíba, março de 2004.

176 p. Il.

Orientadoras: Maria de Fátima Queiroz Vieira Turnell, PhD
Francilene Procópio Garcia, DSc

Palavras-Chave

- 1 - Interface Homem-Máquina
- 2 - Métodos de Concepção de Interfaces
- 3 - Método para Concepção de Interfaces (MCI)
- 4 - Formalismos
- 5 - Ferramentas

CDU - 519.683B

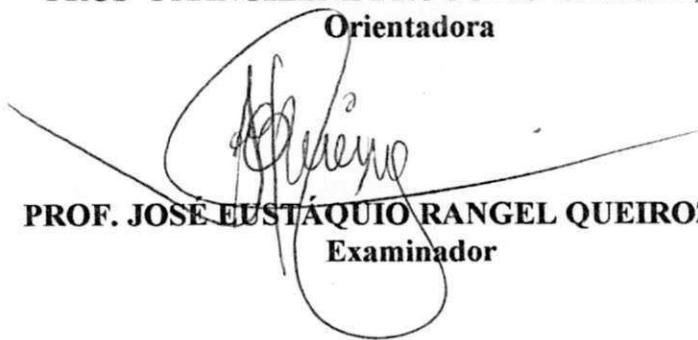
“PROPOSTA DE SUPORTE COMPUTACIONAL AO MÉTODO MCI”

DANIEL SCHERER

DISSERTAÇÃO APROVADA EM 26.02.2004

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


PROFª FRANCILENE PROCÓPIO GARCIA, D.Sc
Orientadora


PROF. JOSÉ EUSTÁQUIO RANGEL QUEIROZ, D.Sc
Examinador


PROF. FERNANDO DA FONSECA DE SOUZA, Ph.D
Examinador

CAMPINA GRANDE – PB

Ao meu pai e minha mãe.

Agradecimentos

Aos meus pais Rudi e Tecla Scherer pelo incentivo e apoio durante todo este período. Por todas as discussões e preocupações em conversas por telefone em cada domingo que passou.

À minha professora e orientadora Maria de Fátima Queiroz Vieira Turnell por toda dedicação, apoio, compreensão e discussões na realização deste trabalho.

À orientadora Francilene Procópio por toda a preocupação quanto à metodologia, confecção e correção do trabalho.

À minha namorada Thiciany pelo companheirismo e amizade durante todo este período.

A todos os colegas e amigos do mestrado, em especial: Cássio, Cidinha, César, Cláudia, Amâncio, Scaico e Zé Alves.

Às funcionárias da Copin: Aninha e Vera, pela compreensão e auxílio nos mais diversos momentos.

Aos meus amigos que mesmo de longe torceram por mim, em especial: Letícia e LF Garcia, Raul, Natanael e Iky.

À CAPES, sem cujo apoio não teria sido possível a conclusão deste trabalho.

Enfim, a todos aqueles que contribuíram para que esta etapa fosse concluída, o meu muito obrigado.

Sumário

SUMÁRIO	VII
LISTA DE FIGURAS.....	XII
LISTA DE QUADROS.....	XV
RESUMO.....	XVIII
ABSTRACT.....	XIX
CAPÍTULO 1 - INTRODUÇÃO.....	1
1.1 CONTEXTUALIZAÇÃO SOBRE IHC.....	1
1.2 TRABALHOS RELACIONADOS	2
1.3 FOCO DO TRABALHO.....	3
1.4 METODOLOGIA	4
1.5 OBJETIVOS	4
1.6 ORGANIZAÇÃO DO DOCUMENTO	4
CAPÍTULO 2 - CONCEPÇÃO DE INTERFACES	6
2.1 INTRODUÇÃO.....	6
2.2 FORMALISMOS PARA A MODELAGEM DA TAREFA	6
2.3 MÉTODOS DE CONCEPÇÃO	9
2.3.1 ADEPT	10
2.3.2 TRIDENT.....	12
2.3.3 MEDITE	14
2.4 MCI.....	17
2.4.1 <i>Análise de Requisitos</i>	17
2.4.1.1 Análise e Modelagem do Perfil do Usuário.....	18
2.4.1.2 Análise e Modelagem do Contexto	19
2.4.1.3 Análise e Modelagem da Tarefa	20
2.4.1.4 Avaliação do Modelo da Tarefa.....	21

2.4.2 <i>Concepção do Modelo da Interação</i>	21
2.4.2.1 <i>Análise e Modelagem da Interação</i>	21
2.4.2.2 <i>Avaliação do Modelo da Interação</i>	23
2.4.2.3 <i>Representação do Cenário</i>	23
2.4.2.4 <i>Construção do Modelo de Navegação</i>	24
2.4.2.5 <i>Avaliação do Modelo de Navegação</i>	24
2.4.3 <i>Concepção e Avaliação do Protótipo</i>	25
2.4.3.1 <i>Geração do Protótipo</i>	25
2.4.3.2 <i>Avaliação do Protótipo</i>	26
2.5 <i>CONSIDERAÇÕES</i>	29
CAPÍTULO 3 - SUPORTES COMPUTACIONAIS.....	30
3.1 <i>INTRODUÇÃO</i>	30
3.2 <i>CRITÉRIOS</i>	30
3.3 <i>AMBIENTES</i>	32
3.3.1 <i>JMVCase</i>	32
3.3.1.1 <i>Processo</i>	32
3.3.1.2 <i>Considerações</i>	34
3.3.2 <i>DIGBE (Dynamic Interaction Generation for Building Environments)</i>	34
3.3.2.1 <i>Processo</i>	35
3.3.2.2 <i>Considerações</i>	37
3.3.3 <i>INGRID (INteractive GRaphical Interface Designer)</i>	38
3.3.3.1 <i>Processo</i>	39
3.3.3.2 <i>Considerações</i>	40
3.3.4 <i>Teallach</i>	40
3.3.4.1 <i>Processo</i>	42
3.3.4.2 <i>Considerações</i>	43
3.3.5 <i>CTTE + Petshop</i>	44
3.3.5.1 <i>Processo</i>	44
3.3.5.2 <i>Considerações</i>	48
3.3.6 <i>TeleUSE</i>	48
3.3.6.1 <i>Processo</i>	48
3.3.6.2 <i>Considerações</i>	50
3.4 <i>RELAÇÃO ENTRE AMBIENTES E AS ETAPAS DO MCI</i>	50

3.5 FERRAMENTAS	51
3.5.1 <i>Euterpe</i>	52
3.5.1.1 Funcionalidades	52
3.5.1.2 Considerações	53
3.5.2 <i>TAMOT</i>	53
3.5.2.1 Funcionalidades	55
3.5.2.2 Considerações	55
3.5.3 <i>iTAOS</i>	56
3.5.3.1 Funcionalidades	57
3.5.3.2 Considerações	57
3.5.4 <i>CTTE</i>	58
3.5.4.1 Funcionalidades	59
3.5.4.2 Considerações	59
3.5.5 <i>DENIM</i>	60
3.5.5.1 Funcionalidades	61
3.5.5.2 Considerações	62
3.5.6 <i>Gartner Web Evaluation Tool</i>	62
3.5.6.1 Funcionalidades	63
3.5.6.2 Considerações	63
3.5.7 <i>Design/CPN</i>	64
3.5.7.1 Funcionalidades	64
3.5.7.2 Considerações	65
3.5.8 <i>OpUS (Opinião do Usuário de Software)</i>	66
3.5.8.1 Funcionalidades	67
3.5.8.2 Considerações	68
3.5.9 <i>DePerUSI (Delineamento do Perfil do Usuário de Sistemas Interativos)</i> ..	68
3.5.9.1 Funcionalidades	69
3.5.9.2 Considerações	70
3.5.10 <i>TERESA (Transformation Environment for inteRactivE Systems representAtions)</i>	70
3.5.10.1 Funcionalidades	71
3.5.10.2 Considerações	72
3.5.11 <i>CADP (Caesar/Aldebaran Development Package)</i>	73
3.5.11.1 Funcionalidades	73

3.5.12 <i>Considerações</i>	74
3.6 RELAÇÃO ENTRE ETAPAS DO MCI E FERRAMENTAS	75
3.6.1 <i>Considerações</i>	75
CAPÍTULO 4 - FORMAS DE INTEGRAÇÃO	79
4.1 INTRODUÇÃO.....	79
4.2 AAIML	82
4.3 AUIML	82
4.4 XUL	83
4.5 UIML	84
4.6 XIML	85
4.7 CONSIDERAÇÕES	86
CAPÍTULO 5 - APLICAÇÃO DO MCI E DE SEU SUPORTE COMPUTACIONAL A UM ESTUDO DE CASO	88
5.1 APRESENTAÇÃO DO SUPORTE COMPUTACIONAL PROPOSTO	88
5.2 ESTUDO DE CASO	90
5.3 APLICAÇÃO DO SUPORTE COMPUTACIONAL AO ESTUDO DE CASO.....	91
5.3.1 <i>Análise de Requisitos</i>	92
5.3.1.1 <i>Análise e Modelagem do Perfil do Usuário</i>	92
5.3.1.2 <i>Análise e Modelagem do Contexto</i>	96
5.3.1.3 <i>Análise, Modelagem e Avaliação do Modelo da Tarefa</i>	96
5.3.2 <i>Concepção do Modelo de Interação</i>	103
5.3.2.1 <i>Representação dos Cenários</i>	103
5.3.2.2 <i>Construção do Modelo de Navegação</i>	105
5.3.2.3 <i>Avaliação do Modelo de Navegação</i>	106
5.3.2.4 <i>Análise, Modelagem e Avaliação do Modelo da Interação</i>	107
5.3.3 <i>Concepção e Avaliação do Protótipo</i>	112
5.3.3.1 <i>Geração do Protótipo</i>	112
5.3.3.2 <i>Avaliação do Protótipo</i>	112
5.4 INTEGRAÇÃO.....	115
5.4.1 <i>Integração através de um conjunto fixo de ferramentas</i>	115
5.4.2 <i>Integração através de um canal de comunicação XIML</i>	116
CAPÍTULO 6 - CONCLUSÕES.....	120

6.1 DISCUSSÃO DOS RESULTADOS.....	120
6.1.1 <i>Suportes Computacionais</i>	121
6.1.2 <i>Suporte Computacional Proposto</i>	122
6.1.2.1 Etapas não contempladas por ferramental.....	123
6.1.2.2 Ferramental indicado na proposta.....	123
6.1.3 <i>Escopo da Solução Proposta</i>	124
6.2 CONSIDERAÇÕES FINAIS	125
6.3 PROPOSTAS DE CONTINUIDADE.....	126
REFERÊNCIAS BIBLIOGRÁFICAS	127
BIBLIOGRAFIA COMPLEMENTAR.....	136
ANEXO A - OUTROS SUPORTES	139
ANEXO B - FORMALISMOS PARA A MODELAGEM DA TAREFA	141
CTT.....	141
DIANE+.....	144
GTA.....	146
MAD.....	149
MAD*.....	151
TAOS.....	153
TKS.....	156

Lista de figuras

Figura 1 - Processo de Interação Homem-Máquina	1
Figura 2 - Processo de projeto do ADEPT [Wils95]	10
Figura 3 - Editor do Modelo do Usuário [Wils93]	11
Figura 4 - Editor de Tarefas [Wils93]	11
Figura 5 - Editor de Modelo Abstrato da Interface [Wils93]	12
Figura 6 - Visão geral de TRIDENT [Boda94]	12
Figura 7 - Visão parcial da árvore de seleção dos objetos abstratos [Vand93]	13
Figura 8 - Editor de apresentação [Vand93]	14
Figura 9 - Metodologia MEDITE [Guer02]	15
Figura 10 - Árvore de tarefas MAD*	15
Figura 11 - Etapas do MCI	18
Figura 12 - Árvore MAD utilizada no MCI	20
Figura 13 - Modelagem de sistemas através da ferramenta JavaRC [Barr99]	33
Figura 14 - Integração dos componentes da ferramenta [Barr99]	33
Figura 15 - Modelo de Tarefa e Interação - TIM [Penn00]	36
Figura 16 - Geração da interface do usuário Técnico [Penn00]	37
Figura 17 - Representação gráfica do Modelo 4D [Guim91]	38
Figura 18 - Interface Organizer [Guim91]	39
Figura 19 - Display Editor [Guim91]	39
Figura 20 - Representação da arquitetura com os modelos [Grif01]	41
Figura 21 - Método Teallach [Barc99]	41

Figura 22 - Editores Teallach [Barc99]	43
Figura 23 - Framework de integração CTTE e PetShop [Nava01]	45
Figura 24 - Árvore de tarefas gerada no CTTE.....	46
Figura 25 - Arquitetura do ambiente PetShop [Nava01]	46
Figura 26 - Editor de Correspondência [Nava01]	47
Figura 27 - Diagrama do TeleUSE.....	49
Figura 28 - VIP (<i>Visual Interactive Presentation</i>)	49
Figura 29 - Euterpe	52
Figura 30 - TAMOT	54
Figura 31 - Tamot e o ambiente ISOLDE.....	55
Figura 32 - iTAOS	57
Figura 33 - CTTE	58
Figura 34 - DENIM	61
Figura 35 - Gartner Web Evaluation Tool [Gart01].....	63
Figura 36 - Design/CPN.....	65
Figura 37 - OpUS	66
Figura 38 - DePerUSI	69
Figura 39 - TERESA	71
Figura 40 - Diversidade de dispositivos.....	79
Figura 41 - Arquitetura WEB [W3C00]	81
Figura 42 - Estrutura da Linguagem XIML [Puer03].....	85
Figura 43 - Método MCI em função das ferramentas.....	88
Figura 44 - Estratégia de Integração entre o Método MCI e Ferramentas de Suporte.....	89
Figura 45 - Ferramentas, Etapas e Processos do MCI.....	91
Figura 46 - Sub-árvore Estudar (versão 1).....	97
Figura 47 - Sub-árvore Estudar (versão 2).....	97

Figura 48 - Sub-árvore Estudar (versão 3).....	98
Figura 49 - Sub-árvore Estudar (versão 4).....	98
Figura 50 - Sub-árvore Estudar (versão final).....	99
Figura 51 - Relatório <i>Task Model Statistics</i>	101
Figura 52 - <i>Task Model Compare</i> entre os modelos.....	103
Figura 53 - Cenário da Tarefa Estudar através de Use Case.....	104
Figura 54 - Modelo de Navegação em CPN	106
Figura 55 - Tela do <i>Presentation Task Sets Generation</i>	109
Figura 56 - Geração do Protótipo	111
Figura 57 - Confirm Panel	112
Figura 58 - MCI com XIIML	117
Figura 59 - Árvore de tarefas CTT	142
Figura 60 - Formalismo Diane+ [Tarb96]	145
Figura 61 - Árvore de tarefas Diane+	145
Figura 62 - Ontologia GTA [Weli98].....	146
Figura 63 - Árvore de tarefas GTA	148
Figura 64 - Árvore de tarefas MAD	150
Figura 65 - Árvore de tarefas MAD*	151
Figura 66 - Conceitos MAD* [Hamm95].....	151
Figura 67 - Meta Modelo TAOS [Mede03].....	153
Figura 68 - Árvore tarefas TAOS	154
Figura 69 - Estrutura abstrata TKS [Mark97].....	157

Lista de Quadros

Quadro 1 - Relação comparativa entre Formalismos e Operadores	7
Quadro 2 - Facilidades x Formalismos	8
Quadro 3 - Descritor de Tarefas	16
Quadro 4 - Lista de objetos e ações do domínio da tarefa.....	21
Quadro 5 - Lista associativa entre domínio da tarefa e da interface.....	23
Quadro 6 - ISO 9241.....	26
Quadro 7 - Etapas do Ensaio de Usabilidade típico	28
Quadro 8 - Template para registro das informações apuradas para um suporte computacional	31
Quadro 9 - Resumo sobre JMVCASE.....	34
Quadro 10 - Resumo sobre DIGBE.....	37
Quadro 11 - Resumo sobre INGRID	40
Quadro 12 - Resumo sobre Teallach	44
Quadro 13 - Resumo sobre CTTE + Petshop.....	48
Quadro 14 - Resumo sobre TeleUSE.....	50
Quadro 15 - Relação entre Ambientes e as Etapas do MCI.....	51
Quadro 16 - Resumo sobre Euterpe.....	53
Quadro 17 - Resumo sobre TAMOT	56
Quadro 18 - Resumo sobre iTAOS.....	58
Quadro 19 - Resumo sobre CTTE	60
Quadro 20 - Resumo sobre DENIM.....	62

Quadro 21 - Resumo sobre Gartner Web Evaluation Tool.....	64
Quadro 22 - Resumo sobre Design/CPN	66
Quadro 23 - Resumo sobre OpUS	68
Quadro 24 - Características relativas ao usuário [Quei01]	69
Quadro 25 - Resumo sobre DePerUSI.....	70
Quadro 26 - Resumo sobre TERESA	72
Quadro 27 - Resumo sobre CADP	74
Quadro 28 - Relação entre Etapas do MCI e as Ferramentas	76
Quadro 29 - Quadro comparativo das linguagens de marcação.....	87
Quadro 30 - Questões propostas para o Estudo de Caso em questão.....	93
Quadro 31 - Formulário FLPU	95
Quadro 32 - Informações iniciais para a modelagem da tarefa.....	97
Quadro 33 - Informações finais da modelagem da tarefa	100
Quadro 34 - <i>Reachability Analysis</i> com definição de tarefa Inicial e Final.....	100
Quadro 35 - <i>Reachability Analysis</i> com definição de tarefa Inicial, Final e Intermediária ...	101
Quadro 36 - Comparativo entre as opções de modelagem da navegação.....	106
Quadro 37 - Modelo de Interação da Tarefa Estudar	107
Quadro 38 - Modelo da interação da Tarefa Estudar através do CTTE	108
Quadro 39 - Conjunto inicial de PTS	110
Quadro 40 - Conjunto PTS após aplicação de critério	110
Quadro 41 - Questionário OpUS	113
Quadro 42 - Relação entre Etapas do Ensaio e Materiais utilizados.....	115
Quadro 43 - Código XML iTAOS x CTTE.....	118
Quadro 44 - Ambientes.....	121
Quadro 45 - Ferramentas.....	122
Quadro 46 - Outros suportes apurados	139

Quadro 47 - Atributos do CTT	142
Quadro 48 - Atributos Diane+	145
Quadro 49 - Descritor da Tarefa	148
Quadro 50 - Descritor do Objeto	148
Quadro 51 - Descritor do Papel	149
Quadro 52 - Descritor do Agente	149
Quadro 53 - Descritor do Evento	149
Quadro 54 - Descritores da Tarefa	150
Quadro 55 - Descritores MAD*	152
Quadro 56 - Descritor de tarefas complexas	154
Quadro 57 - Descritor da tarefa elementar ou ação	155
Quadro 58 - Descritor de pré e pós-situação	155
Quadro 59 - Descritor do método	155
Quadro 60 - Descritor do agente	155
Quadro 61 - Descritor da ferramenta	155

Resumo

Este trabalho propõe uma abordagem de suporte computacional para a concepção e o projeto de interfaces usuário-computador no âmbito do método de concepção e avaliação MCI, baseada na utilização de um *framework* que integra ferramentas diversas a partir de um padrão de integração.

A proposta se baseia: no estudo comparativo de formalismos para a modelagem da tarefa; na análise de métodos de concepção de interfaces e de seus suportes computacionais e na análise de padrões de integração de ferramentas computacionais.

A proposta de suporte computacional foi validada a partir de um estudo de caso apresentado e discutido no texto desta dissertação.

Abstract

This work proposes an approach when offering computational support for the conception and design of user interfaces, in the context of the MCI (Method for the Conception of Interfaces). This approach is based on a framework which integrates already available tools by adopting an integration standard

The proposal is fundamented on: a comparative study of task modelling formalisms; study if it developed initiating for the comparative degree between formalisms used for task modelling; the analysis of interface conception methods and respective supporting tools; and on the analysis of available standards to integrate computational tools.

The proposed computational support was validated through a case study which is presented and discussed in the text of this dissertation.

Capítulo 1 - Introdução

A concepção de interfaces, por muito tempo, tem sido relegada a um segundo plano, tanto em termos de projeto quanto em termos de desenvolvimento. Entretanto, cada vez mais se observa a importância que as interfaces possuem para um sistema, seja por representarem uma parte significativa dos custos de desenvolvimento das aplicações [Somm03], seja por haver “um consenso entre os desenvolvedores de software de que a qualidade do desempenho do usuário está ligada à qualidade de sua interface com o sistema” [Cybi98].

O presente capítulo se inicia por uma breve contextualização sobre a Interface Homem-Computador (IHC), seguida pela discussão dos trabalhos relacionados, o foco da presente dissertação, a metodologia adotada e a apresentação da organização deste documento.

1.1 CONTEXTUALIZAÇÃO SOBRE IHC

As últimas décadas têm sido palco de grandes avanços na área de desenvolvimento de software e, principalmente nos últimos tempos, a concepção da interface tem recebido grande atenção.

Myers, em [Myer96], cita como exemplo deste avanço o uso do Microsoft Windows 95 ou o desenvolvimento da *World Wide Web* que tem como berço os campos universitários e as pesquisas da área de IHC e as tecnologias de hipertexto.

Sommerville, em [Somm03], afirma que “o bom projeto da interface com o usuário é fundamental para o sucesso de um sistema”, podendo inclusive determinar seu sucesso ou fracasso do mesmo. A Figura 1 representa o processo de interação entre o usuário e o sistema, onde pode ser observada a importância fundamental que a interface possui sobre o sistema.

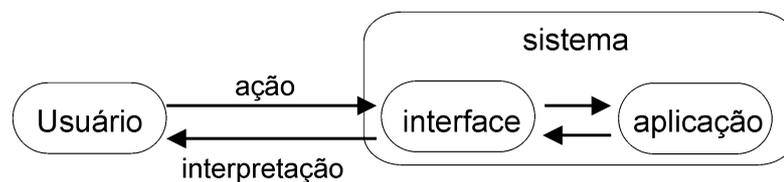


Figura 1 - Processo de Interação Homem-Máquina

O usuário, por sua vez, anseia que o sistema seja de fácil aprendizado, fácil e eficiente de usar e que seja útil para atingir seus objetivos, o que segundo a parte 11 do padrão internacional ISO 9241 pode ser conceituado como usabilidade, ou seja, “um produto pode ser usado por um usuário específico para alcançar um determinado objetivo, de forma efetiva, eficiente e satisfatória para um contexto específico de uso” [ISO98].

Guerrero [Guer02] apresenta os seguintes tópicos, como sendo os principais norteadores na tarefa de obter maior qualidade na interface:

- facilitar o desenvolvimento da interface através de software prototipador e validando-as através de um processo consistente de avaliação;
- utilizar métodos de concepção que possibilitem o desenvolvimento de interfaces a partir do conhecimento do usuário e de sua tarefa; ou
- utilizar um misto das duas formas citadas anteriormente.

Buscando estar em sintonia com as necessidades do usuário e as possibilidades da tecnologia há a área de Interação Homem-Computador, cujo objetivo é fornecer explicações e previsões para fenômenos da interação usuário-sistema, além de resultados práticos para o projeto da interface do usuário [Souz99].

Sendo uma área multidisciplinar, a IHC abrange disciplinas como Ciência da Computação, Psicologia Cognitiva, Psicologia Social e Organizacional, Ergonomia, Lingüística, Inteligência Artificial, Filosofia, Sociologia, Antropologia, Engenharia e *Design* [Souz99].

1.2 TRABALHOS RELACIONADOS

Conceber interfaces para sistemas computacionais consiste em atividade extensa e exaustiva. Uma das formas de facilitar os processos vem a ser a utilização de suportes computacionais. ADEPT¹ e TRIDENT² são algumas das metodologias de concepção de interfaces que já possuem suportes computacionais desenvolvidos, entretanto indisponíveis para uso externo aos grupos de desenvolvimento.

¹ ADEPT: *Advanced Design Environment for Prototyping with Tasks.*

² TRIDENT: *Tools foR an Interactive Development EnvironmeNT.*

Outro projeto mais recente é o Projeto CTTE + Petshop que, através da parceria entre o CNUCE-CNR³ e o LIHS⁴, aborda a integração de duas notações diferentes: modelagem da tarefa com o *ConcurTaskTrees* (CTT) e modelagem de sistemas com o *Interactive Cooperative Objects* (ICO) [Sy00] para, assim, suportar o desenvolvimento de sistemas interativos. Porém apenas parte das ferramentas utilizadas encontra-se disponível para livre uso.

O Projeto CAMELEON é um consórcio iniciado em 2001 entre diversas instituições⁵ com a proposta de desenvolvimento de aplicações que forneçam suporte ao projeto e desenvolvimento de sistemas interativos de software sensíveis ao contexto (*highly-usable context-sensitive interactive software systems*). Entretanto, apesar de passados dois anos, há pouca informação disponível.

Há também grupos como o W3C e associados, que têm pesquisado com o propósito de propor uma linguagem padrão que suporte a concepção de interfaces, tais como: AUIML⁶, UIML⁷ ou XIML⁸. Entretanto, a grande maioria destes padrões não possui suporte computacional associado e os que possuem o disponibilizam apenas mediante pagamento de licenças de uso.

1.3 FOCO DO TRABALHO

Observa-se pelos esforços mais recentes que há uma tendência natural (I) para a integração direta entre ferramentas já desenvolvidas (Projeto CTTE + PetShop) [Nava01] ou (II) pela concepção de interfaces padronizadas de troca de dados (AUIML, UIML, XIML) [Nune02, Abra99, Puer03].

³ CNUCE - CNR - <http://giove.cnuce.cnr.it/>

⁴ LIHS - <http://lis.univ-tlse1.fr/>

⁵ I.S.T.I - C.N.R., IS3-GICE, Motorola Technical Centre Research Italy, Université Catholique de Louvain, Université Joseph Fourier.

⁶ AUIML - *Abstract User Interface Markup Language*.

⁷ UIML - *User Interface Markup Language*.

⁸ XIML - *eXtensible Interface Markup Language*.

Nesta mesma linha, o foco do presente trabalho concentra-se na proposição (I) de um conjunto de ferramentas para apoio computacional ao Método para Concepção de Interfaces (MCI)⁹ e (II) de um padrão que auxilie a integração entre as ferramentas.

1.4 METODOLOGIA

Nas últimas décadas, diversas ferramentas e ambientes integrados foram desenvolvidos com o objetivo de fornecer, direta ou indiretamente, suporte à concepção de interfaces.

Propor um conjunto de ferramentas que possa fornecer suporte ao MCI exige amplo levantamento, a iniciar por ferramentas desenvolvidas no âmbito do Grupo de Interfaces Homem-Máquina (GIHM), passando por ferramentas e ambientes disponibilizados por grupos de pesquisas pertencentes ou não à área de IHC e incluindo ambientes comerciais.

O mesmo caminho de pesquisa teve que ser trilhado para o levantamento e escolha da linguagem a ser utilizada como padrão de integração pelo MCI.

Um estudo complementar, sobre métodos de concepção e formalismos associados, se fez necessário para dar o embasamento teórico visando apoiar as referidas escolhas.

1.5 OBJETIVOS

O principal objetivo deste trabalho é propor um suporte computacional que facilite o projeto e desenvolvimento de interfaces com base no método de concepção e avaliação MCI.

Como objetivos específicos, enumeram-se:

- proposição de um suporte computacional flexível a partir de ferramentas compatíveis com o método e de livre uso;
- proposição de um padrão de integração das ferramentas em um arcabouço.

1.6 ORGANIZAÇÃO DO DOCUMENTO

O presente documento está organizado em seis capítulos, sendo este o primeiro.

⁹ MCI - Metodologia desenvolvida no âmbito do Grupo de Interfaces Homem-Máquina (GIHM) da Universidade Federal de Campina Grande (UFCG).

O Capítulo 2 apresenta, inicialmente, um breve comparativo entre formalismos utilizados para a modelagem da tarefa, seguido pela descrição de métodos de concepção de interfaces, concluindo com a apresentação detalhada do MCI.

No Capítulo 3, são discutidos os suportes computacionais apurados na pesquisa, iniciando pelos critérios de análise de adequação aos propósitos deste trabalho, seguidos pela abordagem de ambientes integrados e de ferramentas computacionais. Tanto para ambientes quanto para ferramentas são apresentados quadros comparativos entre suas características e aquelas necessárias ao MCI. O capítulo conclui com considerações acerca das ferramentas mais adequadas para adoção.

O Capítulo 4 apresenta o levantamento de padrões de integração e discute dentre eles qual o mais indicado para adoção.

No Capítulo 5 é apresentado o suporte computacional proposto para o MCI, bem como o estudo de caso abordado para validá-lo.

Finalizando, no Capítulo 6, é apresentada a discussão dos resultados obtidos, bem como as propostas de continuidade.

Capítulo 2 - Concepção de Interfaces

2.1 INTRODUÇÃO

Como já exposto, a interface usuário-aplicação constitui parte fundamental do sistema. Assim sendo, durante o desenvolvimento das aplicações, deve ser dada a devida atenção a sua concepção.

O presente capítulo inicia pela apresentação de um estudo comparativo entre alguns formalismos para a modelagem da tarefa, uma vez que este é um componente fundamental dos métodos de concepção de interfaces centrados no usuário. O capítulo conclui com a descrição de alguns métodos de concepção, para então, com maior ênfase, expor o Método para Concepção de Interfaces (MCI).

2.2 FORMALISMOS PARA A MODELAGEM DA TAREFA

A modelagem da tarefa é considerada por diversos autores como um dos pontos mais relevantes no projeto de interfaces para sistemas interativos, uma vez que:

- representa ações e seus relacionamentos através de atividades lógicas [Pate02];
- possibilita a obtenção de requisitos de forma mais rápida, os quais são mais fáceis de verificar e validar [Laue03];
- o modelo obtido, geralmente, é de compreensão mais simples e fácil tanto para usuários quanto para a equipe de desenvolvimento.

Diversos formalismos e metodologias foram desenvolvidos na intenção de padronizar o processo de modelar tarefas, dentre os quais: CTT, Diane+, GTA, MAD, MAD*, TAOS e TKS, os quais são detalhados no Anexo B - Formalismos para a modelagem da tarefa.

Segundo [Weli03], a análise e modelagem da tarefa têm por objetivos:

- documentar e comunicar conhecimento entre projetistas;
- analisar o trabalho e encontrar gargalos;

- estruturar as idéias do projetista;
- possibilitar a discussão de aspectos da tarefa entre membros da equipe de projeto;
- possibilitar a proposta de mudanças no processo;
- possibilitar a comparação de alternativas entre projetistas e clientes.

Uma vez que a escolha do formalismo possui implicações significativas no projeto, a escolha inapropriada pode implicar em resultados insatisfatórios ou até na irrelevância das informações apuradas.

Através do Quadro 1 pode-se observar a relação entre os formalismos estudados e os operadores tipicamente oferecidos.

Quadro 1 - Relação comparativa entre Formalismos e Operadores

	CTT	Diane+	GTA	MAD	MAD*	TAOS	TKS
1. Execução de mais de uma tarefa (AND) com Independência de ordem	=	AND	AnyOrder	-	E	AND	-
2. Execução de zero OR mais tarefas	-	OU	Any	-	OU	OR	-
3. ALT / XOR	[]	XOR	Choice	ALT	ALT	XOR	[]
4. Execução em seqüência	>>	-	Successive	SEQ	SEQ	SEQ	>>
5. Execução paralela intercalada	>	-	-	PAR	PAR	PAR	
6. Execução simultânea com troca de informação		-	-	-	SIM	SIM	
7. Execução concorrente e independente		-	-	SIM	-	-	
8. Desativação de execução	[>	-	-	-	-	-	[>
9. Execução em Seqüência com passagem de informação	[]>>	-	-	-	-	-	-
10. Execução em laço	T*	x	T*	BOUCLE	-	-	-
11. Execução opcional	[]	x	-	FAC	-	-	-

-: não possui este operador.

Segue uma breve explicitação dos operadores temporais:

1. **E**, execução de todas as tarefas sem ordem definida;
2. **OU**, deve-se executar ao menos uma tarefa do subconjunto;
3. **ALT**, equivalente ao **OU exclusivo**, ou seja, apenas uma tarefa do subconjunto é executada;
4. **SEQ**, execução das tarefas seguindo uma seqüência rígida;
5. **PAR**, intercalação da execução das tarefas;
6. **SIM**, tarefas executadas de forma sincronizada devido à troca de informações;

7. **Concorrência independente**, tarefas executadas concorrentemente e de forma independente uma da outra;
8. **Desativação (Deactivation)**, quando a execução de T2 provoca a interrupção definitiva de T1;
9. **Seqüência com passagem de informação**, tarefas são executadas em seqüência, porém com transferência de informação de T1 para T2;
10. **Iteração**, ou seja, a tarefa executa repetitivamente até que outra tarefa a interrompa;
11. **Facultativa**, quando a execução da tarefa é opcional;

Alguns autores [Mark94, Hamm95, Bent01] apontam que uma quantidade reduzida de operadores pode resultar em ambigüidades no modelo da tarefa, o que, por sua vez, pode implicar sérios problemas de compreensão por parte da equipe de desenvolvimento. Observando-se sob esta óptica, destaca-se o formalismo CTT, com sua ampla variedade de operadores.

Bentley et al. [Bent01] realizam uma análise dos requisitos para alguns modelos da tarefa onde, além de relacionar os formalismos sob o aspecto dos operadores temporais também os relacionam sob outros aspectos. O Quadro 2 foi elaborado a partir da análise apresentada em [Bent01] acrescida de informações apuradas nas referências que embasaram a redação do Anexo B - Formalismos para a modelagem da tarefa, e do cruzamento destes dados com as informações sobre as ferramentas que apóiam a adoção destes formalismos, sempre que disponíveis.

Quadro 2 - Facilidades x Formalismos

	CTT	Diane+	GTA	MAD	MAD*	TAOS	TKS	
Representação de pré-condições	X	X	X	X	X	X	?	
Representação de pós-condições	-	-	X	X	X	X	?	
Estimativa de tempo para execução de (sub) tarefas	X	X	X	?	X	X	?	
Distingue executores da tarefa	X	X	X	-	X	X	?	
Lista objetos utilizados pela tarefa	X	X	X	X	X	X	X	
Modela tarefas cooperativas	X	-	X	X	X	X	X ₄	
Documentação	Figura da árvore	X	-	X	?	?	X ₂	?
	Texto dos descritores	X ₃	X	X	?	?	-	?
Apoio computacional	X	X	X ₁	í	í	X	í	
Simulador	X	-	-	?	?	-	?	
Multiplataforma	X	-	-	?	?	X	?	

-: não aplicável;

?: impossível determinar devido a indisponibilidade da ferramenta e/ou informações;

- i: publicamente indisponível;
- x₁: disponível apenas para uso não comercial;
- x₂: opção disponível, porém executada de forma insatisfatória;
- x₃: gera apenas uma versão HTML da árvore de tarefas;
- x₄: em fase de estudo e extensão do formalismo;

2.3 MÉTODOS DE CONCEPÇÃO

O encontro de 1968 na cidade alemã de Garmisch abriu espaço para que cientistas e autoridades da área tecnológica debatessem a, assim chamada, “Crise do Software”. Na época, observou-se que havia um crescente uso e, conseqüentemente, desenvolvimento de sistemas informáticos e que este era realizado de forma “quase artística”. A falta de um padrão de concepção acarretava (I) um alto grau de atrasos nos planos de implementação, (II) erros (*bugs*) de software e (III) crescente dificuldade de se realizar a manutenção destas aplicações. Assim, inferiu-se que era necessário o uso de processos de software que, segundo Sommerville, consistem em “um conjunto de atividades e resultados associados que levassem à produção de um produto de software”.

A concepção da interface, por sua vez, por muito tempo foi considerada como sendo uma atividade de segunda ordem, pois o que realmente importava eram as questões relacionadas às funcionalidades. Entretanto, aos poucos, os profissionais perceberam que interfaces difíceis de serem usadas e aprendidas, normalmente, resultavam em alta taxa de erros cometidos pelo usuário. Percebeu-se que havia a necessidade de um projeto claro e consistente da interface da aplicação, visando incutir-lhe o maior grau possível de facilidade de uso e de aprendizado.

Apesar da existência de diversos métodos de concepção, aqueles **baseados em modelos** são, atualmente, os mais adotados. Tal abordagem busca explicitar os resultados da análise e projeto na forma de modelos declarativos. Deste modo, para cada etapa do processo são obtidos modelos que, geralmente, resultam em um ou mais artefatos. Destas metodologias, aquelas que centram o foco no usuário são as mais indicadas, por incluírem no processo de desenvolvimento um representante da comunidade que utilizará o software e portanto o compreende melhor.

Dentre as metodologias baseadas em modelos, citam-se: ADEPT [Mark92], TRIDENT [Boda94], MEDITE¹⁰ [Guer02] e MCI¹¹ [Turn00].

¹⁰ MEDITE: MAD* + EDITOR + ERGONOMIA.

¹¹ MCI: Método para Concepção de Interfaces.

2.3.1 ADEPT

Concebido através da parceria entre Queen Mary e Westfield College, British Aerospace Plc, British Maritime Technology Ltd. e MJC [Mark92], o ADEPT consiste em um ambiente de concepção de interfaces baseado na tarefa e centrado no usuário.

Observa-se na Figura 2 o processo de projeto, bem como os modelos da metodologia ADEPT, cabendo observar que os fluxos em linha cheia correspondem aos caminhos executados através da ferramenta *ADEPT Demonstrator*.

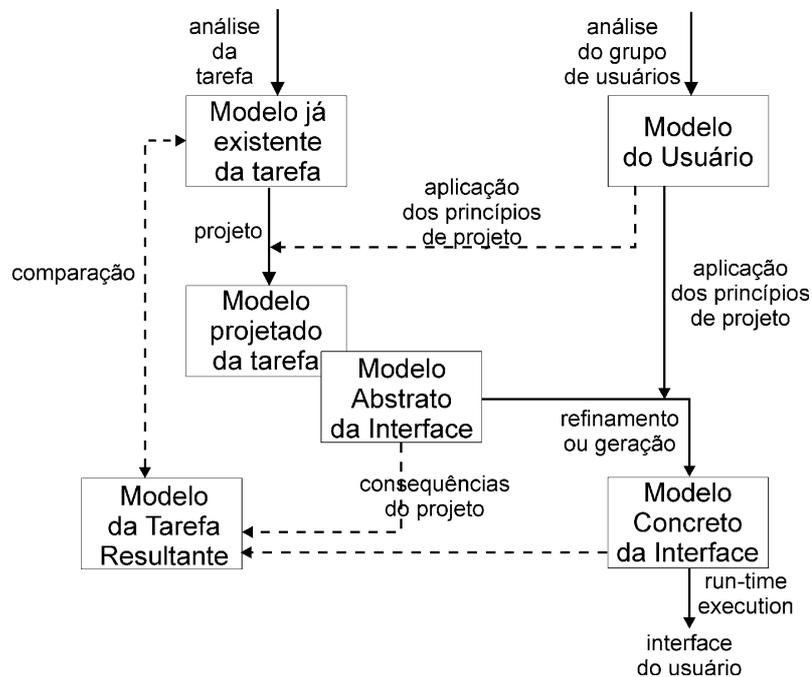


Figura 2 - Processo de projeto do ADEPT [Wils95]

Segundo os autores, em [Wils95], o ambiente possibilita ao projetista partir da análise da tarefa, realizar o projeto da tarefa e convergir para a concepção da interface. Para isto, oferece editores, interpretadores e geradores que suportam a metodologia.

O ambiente, denominado *ADEPT Demonstrator*, possui:

- editor do modelo do usuário (Figura 3), o qual permite concentrar informações acerca de habilidades, conhecimentos e estilos de processamento da informação;
- editor de tarefas (Figura 4), que utiliza uma versão modificada do Task Knowledge Structures (TKS) e permite modelar as tarefas diretamente ou através de menus.
- editor de Modelo Abstrato da Interface (Figura 5), que possibilita a transição do modelo da tarefa para a implementação. Provê uma descrição, em alto nível da interface;

- Modelo Concreto da Interface, consiste de uma descrição da interface independente da plataforma, o modelo é gerado a partir do interpretador, disponível no ambiente.

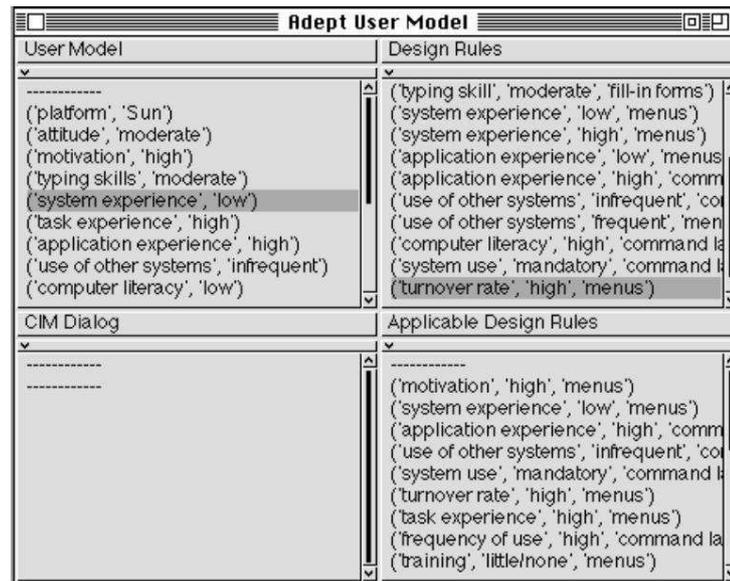


Figura 3 - Editor do Modelo do Usuário [Wils93]

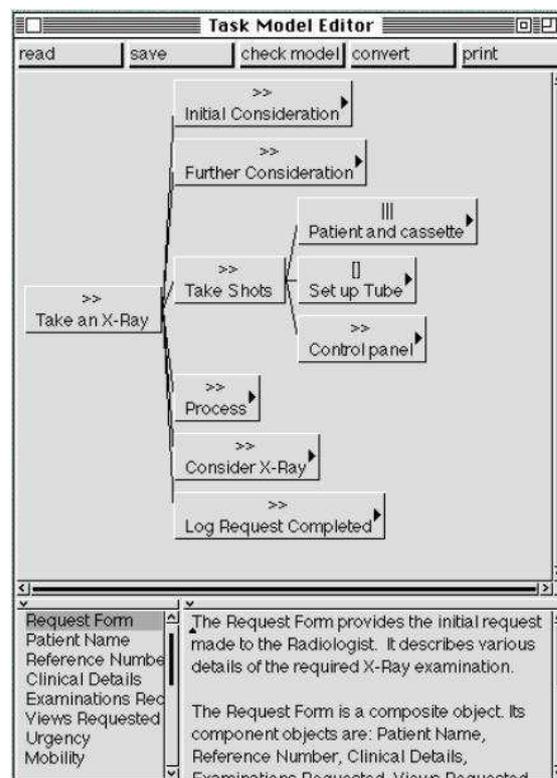


Figura 4 - Editor de Tarefas [Wils93]

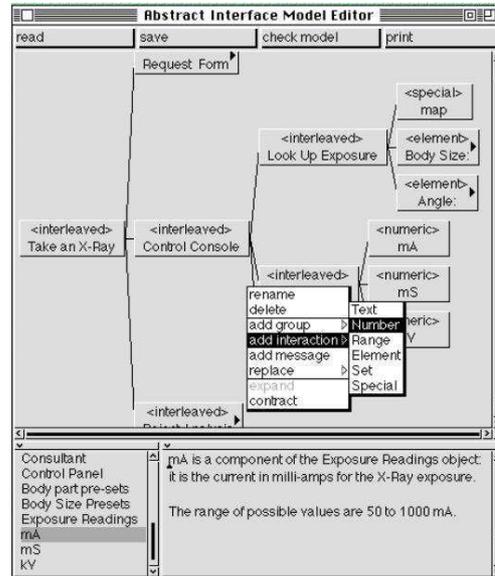


Figura 5 - Editor de Modelo Abstrato da Interface [Wils93]

2.3.2 TRIDENT

Projeto iniciado em 1990 e mantido pelo Instituto de Informática da Facultés Notre-Dame de la Paix à Namur, visando desenvolver aplicações interativas através de uma metodologia suportada por um ambiente computacional [Boda94, Boda95].

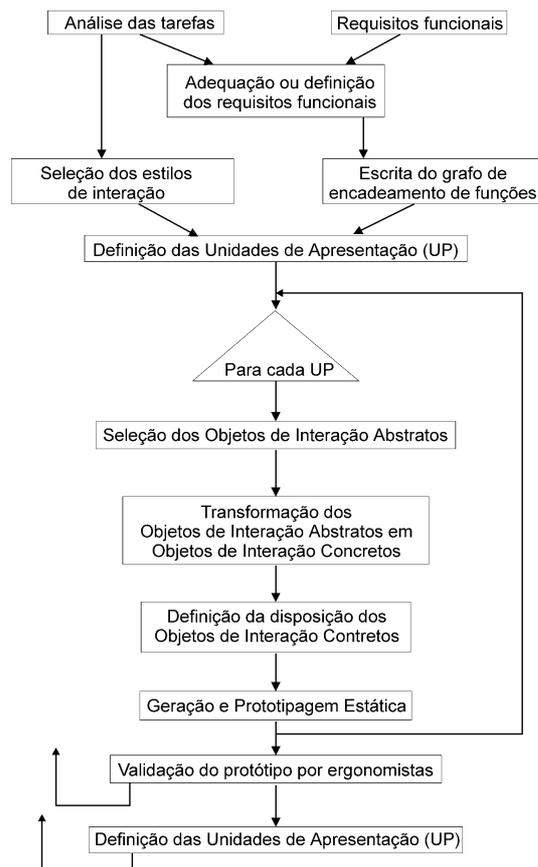


Figura 6 - Visão geral de TRIDENT [Boda94]

O fluxo apresentado na Figura 6 representa a visão geral da metodologia. Através do cruzamento de informações advindo da análise da tarefa e dos requisitos funcionais, gera-se o ACG (*Activity Chaining Graph*) - Grafo de Encadeamento de Funções.

O ACG, por sua vez, descreve o fluxo de informações entre as funções do domínio da aplicação. Estas são necessárias para a execução dos objetivos de tarefa levantados na análise da tarefa.

Enquanto que o ACG caracteriza-se por representar o sistema de forma dissociada do tipo de usuário, o processo de escolha dos estilos de interação, embasa-se justamente nos estereótipos dos futuros utilizadores da aplicação.

Do casamento das informações sobre estilos de interação com aquelas advindas do ACG, definem-se as unidades de apresentação. Segundo apontado em [Boda94], uma UP consiste em “um mundo decomposto de entradas/visualizações contidos em janelas (não necessariamente todas apresentadas simultaneamente) visando executar uma subtarefa da tarefa interativa satisfazendo ergonomicamente o usuário”.

A etapa seguinte consiste em selecionar os objetos de interação abstratos (Figura 7), que é, segundo os autores, assistido por um sistema especialista que permite a definição automática ou semi-automatizada.

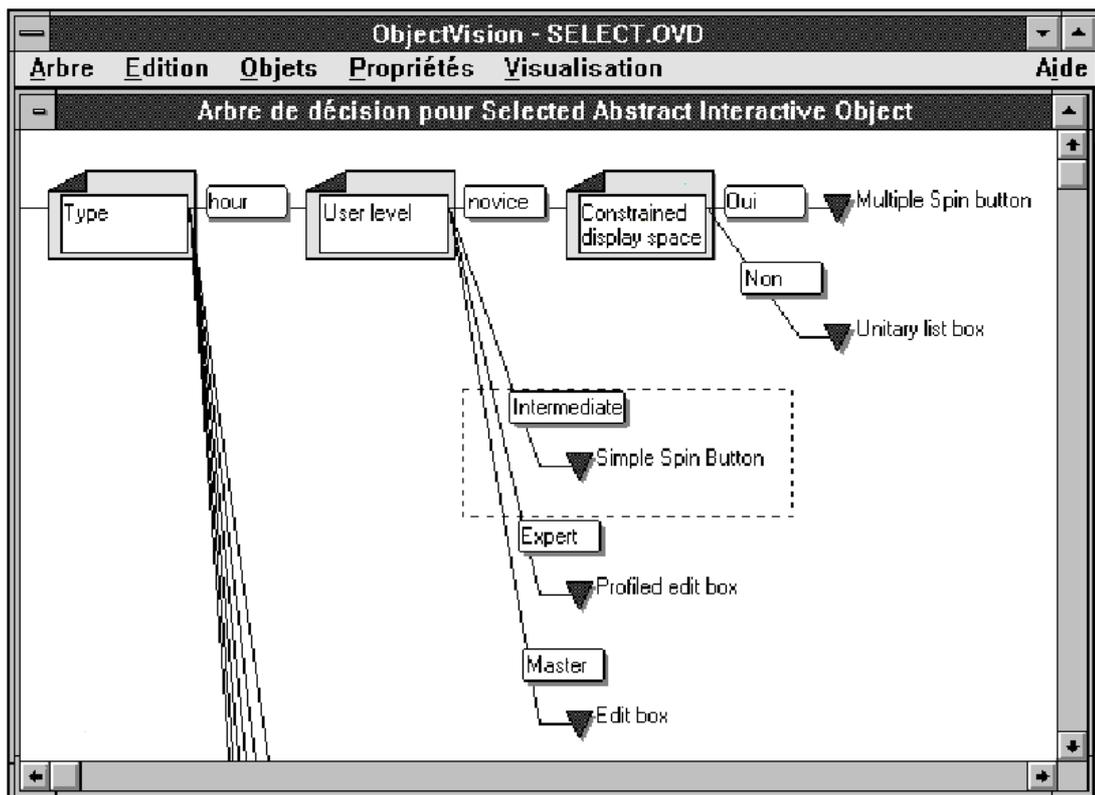


Figura 7 - Visão parcial da árvore de seleção dos objetos abstratos [Vand93]

A transformação dos objetos abstratos em concretos leva em consideração a plataforma de desenvolvimento, tendo sido observado pelos autores em [Boda94] que, nesta conversão, podem ocorrer “degradações” do que se esperava pelos objetos abstratos e o que se obteve com os objetos concretos, porém ressalta que isto ocorre, normalmente devido a deficiências das plataformas as quais podem não possuir o objeto concreto desejado.

A definição da disposição dos objetos concretos permite ao projetista, através de editor gráfico (Figura 8), retrabalhar a apresentação de forma a melhor adaptá-la às necessidades do usuário.

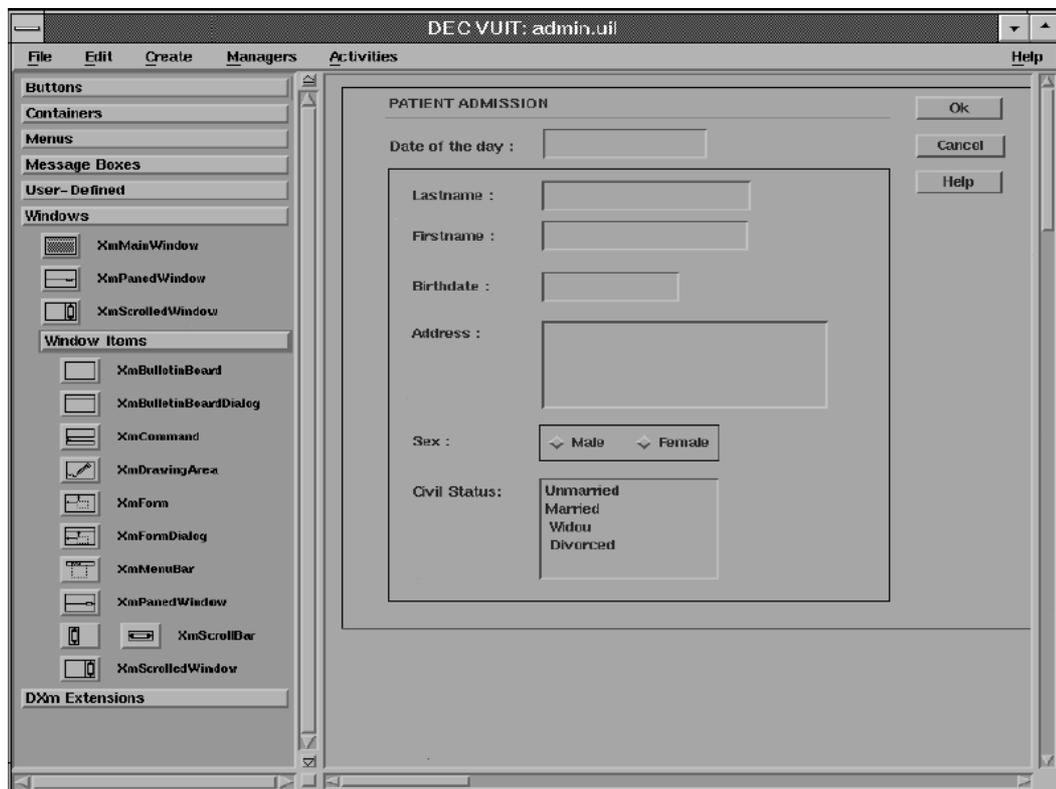


Figura 8 - Editor de apresentação [Vand93]

Os passos seguintes consistem na geração do protótipo e na sua validação por projetistas e usuários.

2.3.3 MEDITE

Desenvolvida por Guerrero [Guer02], consiste em uma metodologia adotada no âmbito do GIHM, a qual busca preencher uma lacuna no desenvolvimento de software, no que diz respeito à especificação de interfaces com base em critérios ergonômicos.

O princípio desta metodologia é guiar o desenvolvedor, experiente ou não em ergonomia, passo a passo na especificação das interfaces. Para tanto, propõe etapas bem

definidas com modelos e artefatos específicos. A Figura 9 apresenta a metodologia com os círculos representando os processos e os retângulos representando os produtos gerados e os modelos utilizados.

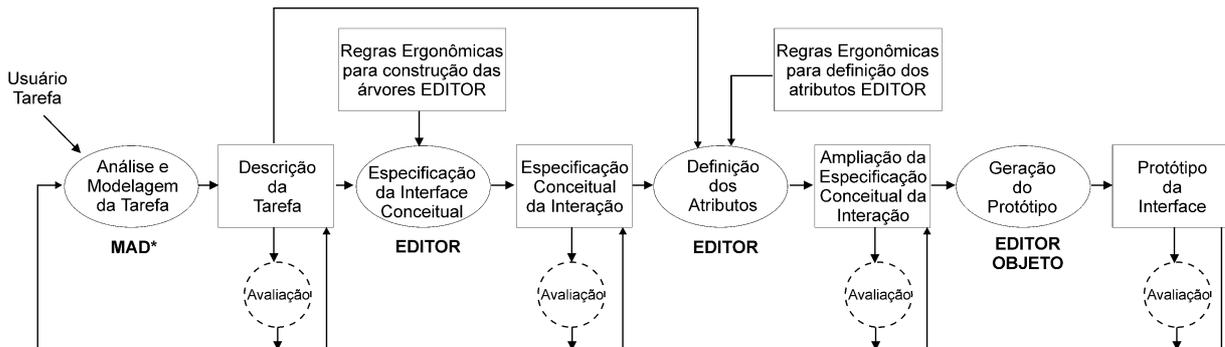


Figura 9 - Metodologia MEDITE [Guer02]

Partindo da análise e modelagem da tarefa busca-se identificar os objetivos do sistema a ser desenvolvido. Esta etapa embasa-se no formalismo MAD*, o qual é composto por uma representação hierárquica da árvore de tarefas (Figura 10) associado a descritores (Quadro 3) [Guer02].

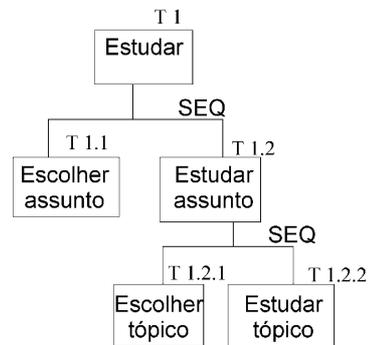


Figura 10 - Árvore de tarefas MAD*

A etapa seguinte (especificação conceitual da interface) parte do modelo da tarefa e, através do auxílio de regras ergonômicas, gera-se um novo modelo de árvore de tarefas baseada no formalismo EDITOR¹² que possui as seguintes informações:

- *apresentação* complementa-se com informações sobre localização, formato, tamanho de fonte, background, entre outros;
- *abstração* complementa-se com dados sobre o domínio da aplicação;
- *controle* com a definição do encadeamento de diálogo.

¹² EDITOR [Lula92] é um modelo de arquitetura que alia aspectos remarcáveis do modelo de Seeheim às vantagens de um modelo multi-agente [Guer02].

Quadro 3 - Descritor de Tarefas

T1 “ESTUDAR” (Nível intermediário)	
Corpo da Unidade-Tarefa	
Corpo da Tarefa	
Número da tarefa	T1
Nome	Estudar
Objetivo	Estudar o tema contido no Tutorial sobre Máquina de Turing
Modo	Facultativa
Ator	Aluno (alunos de teoria da computação)
Prioridade	3 (0 a 3)
Interruptibilidade	Interruptível com reinício em curso
Estado de execução	-----
Condições de entrada da tarefa	
Estado do mundo inicial	Ver Estado do Mundo Inicial da T 0 Ícone Estudar
Pré-condições / condições desencadeantes	Ver Pré-condições da T 0 Ícone Estudar disponível = verdade
Condições de saída da tarefa	
Estado do mundo final	<i>Browser</i> , Tutorial, Aluno, Conhecimento ou Informação (assuntos e tópicos)
Pós-condições / resultados da execução	<i>Browser</i> utilizado = verdade Tutorial consultado = verdade Aluno satisfeito = verdade Conhecimento ou Informação adquiridos (assuntos e tópicos estudados = verdade).
Decomposição da Unidade-Tarefa	
Relação	Seqüencial (SEQ)
Lista de subtarefas	T 1.1 - Escolher assunto T 1.2 - Estudar assunto
Nível superior	T 0 - Consultar tutorial
Tarefa Elementar	Não
Atributos orientados à concepção da interface	
Restrição de diálogo	-----
Tipo	Mental e sensório-motor
Modalidade	Interativa
Centralidade	Importante Frequência: elevada Entidades importantes: <i>Browser</i> , tutorial, aluno, conhecimento
Papel do operador	Aluno Experiência com a tarefa: iniciante, média [iniciante, média e alta] Experiência com o sistema: elementar e médio [elementar, médio e <i>expert</i>] Competência: selecionar um ícone entre vários, navegar na Internet
Meios de interação	Interação icônica

Segundo Guerrero [Guer02], a geração do protótipo pode ser realizada de forma automática através do uso das árvores EDITOR. Isto é possível em virtude do “modelo de arquitetura espelhar exatamente o modelo conceitual de interação”.

A etapa de avaliação apesar de exposta como uma quinta etapa, na verdade está distribuída em todas as etapas anteriores, consistindo em uma avaliação de cada artefato gerado de cada etapa (Figura 9) Observa-se na Figura 9 que, em caso de necessidade, pode-se retornar não somente à etapa avaliada como retornar a etapas anteriores, havendo portanto um desenvolvimento iterativo. No entanto, na descrição do método, não é definido como e em que bases se processa esta avaliação.

2.4 MCI

Metodologia desenvolvida também no âmbito do GIHM e já largamente utilizada em vários projetos e trabalhos de dissertações, caracteriza-se por ser iterativa, centrada no usuário, utiliza-se de modelos formais e, assim como MEDITE, objetiva o desenvolvimento de produtos interativos baseados em critérios ergonômicos e princípios de usabilidade.

O MCI possui as seguintes características [Turn00]:

- utiliza a noção de artefato, de forma que ao término de cada etapa há a geração de um produto;
- utiliza uma abordagem incremental, de forma que os resultados de uma etapa constituem artefatos de entrada para etapas posteriores;
- o ciclo de concepção é centrado na avaliação, assim, os resultados de cada etapa são submetidos à avaliação antes de constituírem a entrada da etapa seguinte.

A Figura 11 apresenta as etapas constituintes do método bem como a seqüência em que ocorrem, havendo em seguida, a explicitação das etapas e subetapas do método.

2.4.1 Análise de Requisitos

Esta etapa se subdivide em três subetapas as quais consistem em: apurar informações sobre o usuário, sobre o contexto de uso e sobre a tarefa que será realizada. Este levantamento visa responder três questões:

- **quem** é o usuário do produto;
- **onde** e sob **quais condições** o produto será utilizado;

- **o que** precisa ser feito, ou seja, quais tarefas devem ser executadas.

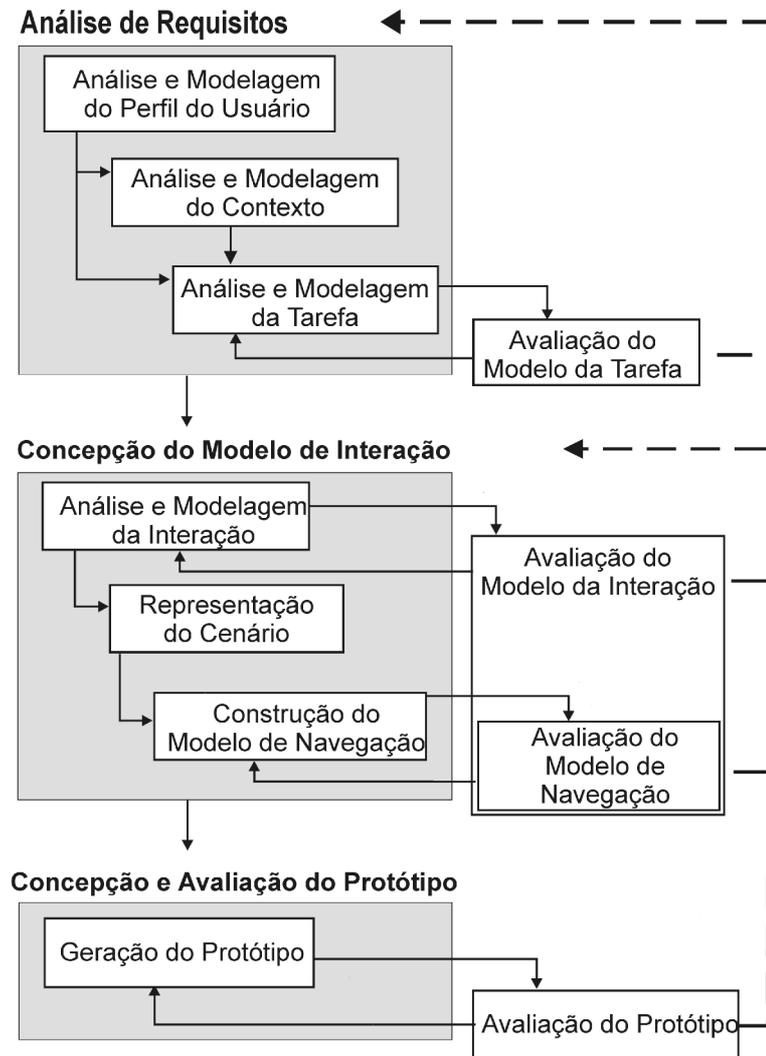


Figura 11 - Etapas do MCI

2.4.1.1 Análise e Modelagem do Perfil do Usuário

O levantamento do perfil do usuário objetiva explicitar quais são as características do usuário do sistema. Para tal, diversos atributos que representam o usuário podem ser apurados, tais como:

- limitações;
- habilidades;
- preferências;
- interesses.

Segundo Turnell [Turn00] a escolha dos atributos e a atribuição de um grau de importância são tarefas do projetista e a apuração destas informações deve estar apoiada em:

- conhecimentos científicos;
- opiniões obtidas do usuário;
- dados mensurados ou observados pelo próprio projetista.

Para esta subetapa, o MCI apóia-se no uso de duas ferramentas:

- **Formulário para o Levantamento do Perfil do Usuário (FLPU)** [Turn00];
- **Formulário para Delineamento do Perfil do Usuário (DePerUSI)** [Quei01].

Ambos consistem em questionários. No primeiro caso, é um questionário a ser preenchido pelo projetista com informações sobre o usuário, as quais podem ter sido colhidas com o usuário a partir de entrevistas ou observação. No segundo caso, o questionário deve ser preenchido por grupo representativo dos usuários.

A segunda abordagem, segundo Queiroz [Quei01], apesar de apresentar algumas desvantagens apresenta diversos atrativos, dentre os quais cita-se:

- possibilidade de focalizar o tema de interesse;
- possibilidade de realizar análises estatísticas sobre os escores produzidos;
- possibilidade de aplicação rápida o que permite redução de custos envolvidos;
- possibilidade de apurar grande quantidade de dados;
- possibilidade de garantir o anonimato aos respondentes.

2.4.1.2 Análise e Modelagem do Contexto

O levantamento do contexto segue linha semelhante àquela do levantamento do perfil do usuário, explicitando **onde** e sob **quais condições** o produto deverá ser usado.

Informações como: luminosidade do ambiente; nível de periculosidade; e nível de *stress* que o ambiente proporciona são algumas das informações de interesse nesta subetapa.

Atualmente, não há uma ferramenta específica para esta subetapa. Entretanto, devido à similaridade com a etapa anterior, infere-se que possam ser utilizados os mesmos instrumentos daquela, após uma devida adequação.

2.4.1.3 Análise e Modelagem da Tarefa

Atividade que envolve a explicitação sobre o **que** será feito pelo usuário. A tarefa, segundo Turnell [Turn00], “é aquilo que o usuário deve realizar durante a preparação, operação e manutenção de um sistema”.

No MCI, durante o levantamento de tarefas, o projetista busca compreender:

- as tarefas a serem realizadas
- as seqüências de ações necessárias para realizar cada uma das tarefas;
- da importância que cada tarefa possui;
- qual o papel do usuário na realização da tarefa;
- quais pontos podem ser automatizados de forma a aumentar a produtividade, eficiência e qualidade da interação.

Bem como procurar analisar as tarefas, de forma a:

- determinar o grau de dificuldade em se realizar cada tarefa (segundo ponto de vista do usuário);
- determinar a gravidade da consequência dos erros quando da execução de cada tarefa.

O modelo da tarefa é construído organizando-se as informações de forma hierárquica, iniciando pela tarefa objetivo (a de nível mais alto), chegando-se às tarefas mais elementares (no nível das folhas). Para tanto, o MCI apóia-se no formalismo MAD ou em qualquer outro formalismo que como o MAD adote a representação arbórea das tarefas. A Figura 12 apresenta um exemplo de árvore utilizada no MCI. Cabe observar que o modelo de tarefas utilizado pelo MCI atribui números a cada tarefa, a exemplo do que é feito no MAD*.

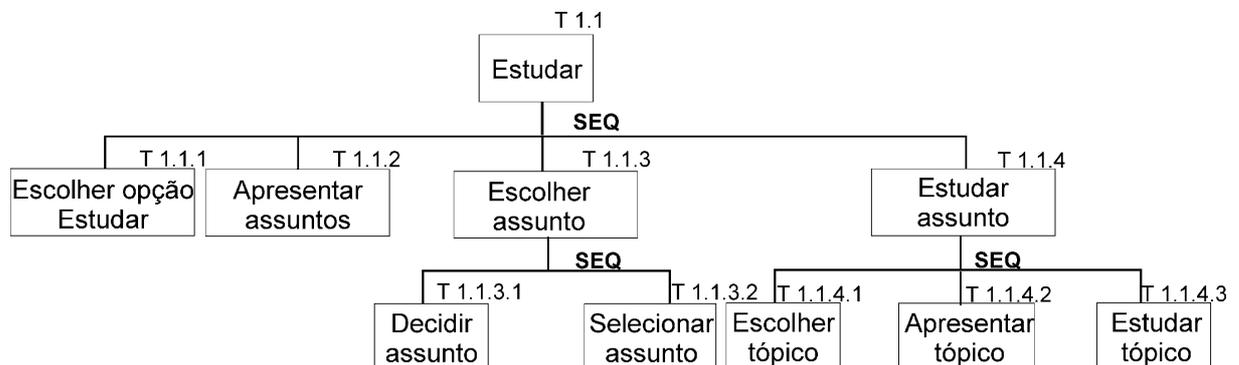


Figura 12 - Árvore MAD utilizada no MCI

A análise e modelagem da tarefa, além de representar a atividade do usuário, também apóia as atividades de (I) especificação do sistema, (II) projeto da interface e (III) elaboração do material de treinamento [Turn00]. Entretanto, apesar de sua importância no processo, esta etapa não possui apoio computacional adequado no MCI. Dos suportes computacionais analisados, a elaboração do material de treinamento é a atividade com menor suporte.

2.4.1.4 Avaliação do Modelo da Tarefa

Tendo sido concluído o modelo da tarefa, cabe ao projetista, conjuntamente com o usuário, validá-lo observando se está completo, verificando se todas as tarefas e subtarefas foram contempladas, assim como a relação entre elas. Esta análise, ao ser realizada nos estágios iniciais do projeto, reduz sensivelmente incoerências nos requisitos levantados.

2.4.2 Concepção do Modelo da Interação

Esta etapa consiste em propor uma concepção de interface considerando requisitos apurados e critérios ergonômicos. Esta concepção consiste de um modelo de apresentação e de um modelo de navegação, ambos construídos para um ou mais cenários escolhidos pelo projetista de acordo com a relevância para o projeto.

2.4.2.1 Análise e Modelagem da Interação

Etapa I - Levantamento dos objetos e ações das tarefas

De posse das características do usuário, da tarefa e do contexto e de um conhecimento ergonômico implícito, compõe-se inicialmente um quadro no qual são colocados dados oriundos do modelo da tarefa (lista de objetos e ações) e, se for relevante, acrescenta-se informações sobre o grau de complexidade de cada tarefa (ver Quadro 4).

Quadro 4 - Lista de objetos e ações do domínio da tarefa

Identificação da Tarefa	Modelo da Tarefa		
	Ação	Objeto	Complexidade
T 1.1.1	Selecionar	Opções	Média
T 1.1.2	Apresentar	Opções	Média
T 1.1.3.2	Selecionar	Opções	Média

Construído o quadro, reagrupam-se os elementos em **ordem alfabética de ações** e/ou dos **objetos**, de forma que o projetista possa obter informações sobre a realização de um

mesmo tipo de ação sobre objetos diferentes e sobre quais ações são realizadas sobre um mesmo tipo de objeto, além de poder verificar frequência de realização destas ações.

Esta análise aliada, a informações sobre o perfil do usuário e sobre o contexto, possibilita ao projetista avaliar e inferir acerca da complexidade de cada tarefa.

Etapa II - Mapeamento do Domínio da Tarefa para o Domínio da Interface

Esta etapa consiste na concepção da interface, propriamente dita. Nela o projetista associa aos elementos identificados no modelo da tarefa: os objetos que os representarão na interface; decidindo qual o estilo de interação a ser adotado, os dispositivos de interação que serão utilizados pelo usuário para manipular estes objetos e as metáforas adotadas na sua representação.

Esta decisão é fundamentada no conhecimento advindo dos artefatos gerados nas etapas anteriores: perfil do usuário, modelo da tarefa e descrição do contexto associado a princípios ergonômicos heurísticos do projetista ou embasado em documentos de referência como guias de estilo, diretrizes de projetos ou normas internacionais (ex.: Guia de Smith & Mosier, ISO 9241, *Ergonomic Criteria for the Evaluation of Human Computer Interfaces* [Turn00]).

- **estilos de interação**, associando-os a cada tarefa (ex.: seleção por menu, comando, formulário, entre outros);
- **dispositivos de interação**, associando-os a manipulação dos objetos e realização das ações (ex.: mouse, caneta ótica, outros);
- **metáforas e manipuladores (objeto da interface)**, proposição dos objetos e ações da interface que concretizarão os objetos e ações da tarefa (ex.: metáfora *desktop* com os objetos característicos desta metáfora: cesta de lixo, caixa de janela, caixa de diálogo, barra de menu).

Como resultado, obtém-se o conteúdo ilustrado no Quadro 5. Nele, cada **ação e objeto** do domínio da tarefa se encontra associado a um **estilo de interação, objeto da interface e agrupamento visual** do domínio da interface.

A realização desta etapa também não possui apoio computacional ficando, portanto, sob responsabilidade do projetista a construção dos quadros de mapeamento. Aqui seria interessante poder contar com um mecanismo de apoio a decisão que pudesse guiar projetistas inexperientes com a escolha das regras ergonômicas adequadas ao contexto do projeto.

Quadro 5 - Lista associativa entre domínio da tarefa e da interface

Identificação da Tarefa	Modelo da Tarefa			Modelo da Interação		
	Ação	Objeto	Complexidade	Estilo de Interação	Objeto da Interface	Agrupamento Visual
T 1.1.1	Selecionar	Opções	Média	Menu	Menu Principal opção “Estudar”	Janela Principal
T 1.1.2	Apresentar	Opções	Média	Agrupar	Lista de assuntos disponíveis	Janela Estudar_Assunto
T 1.1.3.12	Selecionar	Opções	Média	Menu	Caixa de seleção simples	Janela Estudar_Assunto

2.4.2.2 Avaliação do Modelo da Interação

A avaliação do modelo da interação pode ser realizada segundo critérios ergonômicos [Turn00], tais como:

- existência de caminhos alternativos para a realização de uma mesma tarefa;
- coerência do projeto visual com as necessidades e/ou anseios do cliente/usuário;
- coerência quanto aos níveis de profundidade da navegação.

Observa-se que assim como a escolha das ações e objetos do domínio da interface, também a avaliação dos mesmos embasa-se no uso de regras ergonômicas e/ou guias de estilo.

2.4.2.3 Representação do Cenário

Desenvolver sistemas centrados no usuário exige esforço tanto da equipe de desenvolvimento quanto do usuário. Para possibilitar a participação do usuário nas decisões, o MCI propõe a prototipação para **cenários de uso** que representem situações de maior interesse para discussão com o usuário. Em geral, são escolhidos cenários que representem as tarefas mais complexas ou mais suscetíveis a erros do usuário.

Sendo constituído por situações familiares ao usuário, possibilita-se uma abertura de comunicação entre projetista e cliente, para que se possa analisar e validar os modelos gerados.

Segundo Turnell [Turn00], a introdução de casos de uso no MCI teve como propósitos:

- permitir uma melhor captura do uso do sistema a ser desenvolvido;
- ampliar o canal de comunicação projetista-usuário de forma a garantir a construção da interface desejada.

Dentre os diversos modelos existentes para a geração de cenários, o MCI destaca:

- *Use-cases*, técnica baseada em cenário para a obtenção de requisitos utilizada pela notação *Unified Modeling Language* (UML) e que em sua forma mais simples permite identificar agentes envolvidos em uma interação além de especificar o tipo de interação [Somm03];
- *MSC* ou *Message Sequence Chart*, consiste em uma forma visual de apresentar as informações, tendo em conta sua ordem de execução. Cenários expostos através do MSC podem ser mais bem compreendidos pelo usuário por serem mais intuitivos para este.

2.4.2.4 Construção do Modelo de Navegação

A navegação é um dos componentes mais críticos do projeto da interface. Particularmente quando os sistemas são críticos do ponto de vista do tempo de execução e tem restrições de navegação associadas à segurança no acesso às funcionalidades do sistema.

O modelo de navegação, consiste em uma representação formal (*statecharts*, Redes de Petri, entre outras) das situações de transição da interface projetada de maneira que independa de dispositivos de interação e sobre estas aplicar os cenários elaborados em etapa anterior [Sous99]. A representação formal permite uma avaliação do modelo baseada em suas propriedades e apoiada por ferramentas antes que seja iniciada a implementação. Para tanto, é de grande interesse que o formalismo utilizado possibilite representar graficamente o modelo em questão, além de possuir um embasamento matemático e apoio computacional.

Tais características, aliadas a um conjunto pré-definido e/ou programável de possibilidades de análise formais e quantitativas, podem ser encontradas no formalismo de Redes de Petri Coloridas (CPN) e seu apoio computacional **Design/CPN** [Mura89].

Entretanto, além da ferramenta computacional associada ter seu uso restrito ao ambiente Linux, Sousa aponta que os modelos gerados sob este formalismo, normalmente, são dependentes de contexto, o que tende a dificultar a reutilização do modelo gerado para contextos diferentes [Sous99].

2.4.2.5 Avaliação do Modelo de Navegação

Os pontos a serem avaliados no modelo de navegação estão diretamente relacionados ao contexto de uso da aplicação, porém além de verificar a completa execução dos cenários gerados, pode-se inferir alguns requisitos ergonômicos, tais como:

- existência de caminhos alternativos para execução de uma mesma tarefa, tanto extensos (usuário leigo) quanto otimizados (usuário *expert*);
- existência de bloqueios na navegação, ou seja, se existem pontos nos quais o usuário não possui opções de navegação;
- existência de reversibilidade quando assim for demandado, ou seja, permite-se ao usuário retornar ações (*undo*);
- fácil acesso à ajuda;
- fácil acesso à saída do sistema.

2.4.3 Concepção e Avaliação do Protótipo

Mesmo em desenvolvimentos centrados no usuário, pode haver incoerências advindas, do fato do cliente não conseguir prever, com absoluta certeza, quais os impactos da adoção de um sistema computadorizado no dia-a-dia de trabalho.

Muitas das dúvidas dos usuários podem ser sanadas pela utilização de um protótipo, que, segundo Sommerville [Somm03], corresponde a “uma versão inicial de um sistema de software, que é utilizada para mostrar conceitos, experimentar opções de projeto e, em geral, para conhecer mais sobre os problemas e suas possíveis soluções”.

2.4.3.1 Geração do Protótipo

Como pode ser observado na Figura 11, a geração do protótipo visa concretizar a concepção da interface com base no conjunto de artefatos apurados nas etapas anteriores do MCI. Apesar dos esforços despendidos na modelagem da tarefa e da interação, estas informações, no MCI, não estão organizadas de forma a serem diretamente mapeadas para uma linguagem de programação. A explicitação da concepção é muito discursiva e dependente dos conhecimentos e interpretações do desenvolvedor.

Souza [Sous99] propõe o uso da ferramenta **Devguide** para a concepção da interface, em virtude desta oferecer “elementos pré-definidos para a construção das interfaces”. Entretanto além desta ferramenta não estar publicamente disponível, a mesma possui aplicação restrita ao ambiente UNIX.

2.4.3.2 Avaliação do Protótipo

Objetivando mensurar a usabilidade do produto que resultará da concepção, deve-se proceder a avaliação de um protótipo construído com base nesta concepção. Esta avaliação pode ser feita através de três formas distintas:

- inspeção de conformidade a padrões;
- mensuração do desempenho do usuário durante o uso do produto; e
- sondagem da satisfação do usuário quanto ao produto.

Inspeção de conformidade a padrões

Inspeccionar um produto segundo padrões consiste em verificar o seu grau de conformidade frente a um conjunto de regras.

Quando da utilização de guias de estilos de empresas como Sun e Microsoft ou uso de normas de instituições, subentende-se que o produto avaliado será usado predominantemente por um grupo específico de usuários e/ou em um escopo definido (como produtos a serem utilizados sobre o ambiente Windows).

Já no caso do uso do Padrão ISO 9241 como norteador da inspeção, subentende-se uma abrangência internacional de uso do produto. O ISO 9241 possui 17 partes, sendo que as 9 primeiras dizem respeito a aspectos do hardware, enquanto que as demais abordam os aspectos relativos a software. O Quadro 6 consiste de extrato de material apresentado por Queiroz em [Quei01] e resume sucintamente as 17 partes do padrão ISO 9241.

Quadro 6 - ISO 9241

Parte	Conteúdo
1	Apresentação das partes do padrão, de alguns princípios básicos e recomendações de uso do padrão e seus relatórios.
2	Apresentação de orientações sobre a especificação de requisitos de tarefas.
3	Apresentação das especificações dos requisitos de projeto de hardware de visualização usados em escritórios, além da proposição para o teste de desempenho do usuário.
4	Apresentação da especificação de características ergonômicas de teclado alfanumérico.
5	Apresentação das especificações de requisitos ergonômicos para estações de trabalho com terminais de visualização.
6	Apresentação de especificações de requisitos ergonômicos para ambientes de trabalho.
7	Apresentação de especificações e detalhamentos de métodos de mensuração da ofuscação e de reflexos produzidos nas telas de dispositivos de visualização.
8	Apresentação de especificações dos requisitos ergonômicos para terminais de visualização em cores.
9	Apresentação das especificações dos requisitos ergonômicos para dispositivos de entrada, além de proposição de procedimento de teste de desempenho.

10	Exposição dos princípios ergonômicos de alto nível aplicáveis ao projeto de diálogos usuário-computador.
11	Apresentação de orientações acerca da especificação ou avaliação da usabilidade em termos de medidas de desempenho e satisfação do usuário. Explicação de como a usabilidade de um produto pode ser especificada e avaliada como parte de um sistema de qualidade.
12	Apresentação dos aspectos ergonômicos específicos envolvidos na representação e apresentação das informações em terminais de visualização.
13	Apresentação de recomendações para o projeto e a avaliação de atributos de orientação do usuário de interfaces de software.
14	Apresentação de recomendações para o projeto ergonômico de diálogos via menus.
15	Apresentação de recomendações para o projeto ergonômico de diálogos via linguagem de comandos.
16	Apresentação de recomendações para o projeto ergonômico de diálogos via manipulação direta.
17	Apresentação de recomendações para o projeto ergonômico de diálogos via preenchimento de formulários.

Queiroz [Quei01] ressalta que, para cada parte do padrão ISO 9241¹³, existem listas de inspeção (*checklists*) estruturadas em formato tabular. Assim, o **projetista** possui suporte na inspeção da aplicabilidade das recomendações do ISO 9241 a projetos específicos e, aos **avaliadores**, permite-se verificar a adoção, pelos projetistas, das recomendações existentes no padrão. Este têm sido o padrão adotado nas avaliações dos produtos concebidos com o MCI.

Mensuração do desempenho do usuário

Consiste em estudar um processo interativo usuário-computador específico em condições “reais” ou “controladas”. A mensuração do desempenho do usuário, segundo Queiroz [Quei01], coloca-se como sendo “o método mais fundamental e, em alguns aspectos, insubstituível”.

O Ensaio de Usabilidade pode ser realizado tanto no próprio ambiente de trabalho do usuário quanto em ambientes especialmente concebidos para esta função, tais como laboratórios de testes de usabilidade.

Uma vez iniciado o ensaio, cabem aos especialistas em interfaces coletarem dados relacionados à facilidade/dificuldade de interação entre a amostra da comunidade usuária e a aplicação em teste.

Queiroz [Quei01] apresenta uma síntese relacionando etapas e suas descrições, reproduzida no Quadro 7.

¹³ Partes 14, 15, 16 e por conseguinte a 17.

Quadro 7 - Etapas do Ensaio de Usabilidade típico

Etapa	Descrição
Preparação	- Verificação de todas as condições de teste (sala, sistema computacional, material necessário em geral, instruções e instrumentos de questionamento) antes da introdução do usuário de teste no ambiente.
Introdução	- Recepção do usuário e explanação breve do(s) propósito(s) do ensaio. - Introdução do(s) procedimento(s) de teste. - Distribuição das instruções e esclarecimento de dúvidas.
Teste	- Observação imparcial de todos os eventos interativos usuário-sistema. - Anotação, quando necessária, de detalhes pertinentes ao contexto avaliatório. - Diálogo limitado ao estritamente necessário com apenas um dos avaliadores, conforme acordo pré-fixado.
Questionamento/ Análise de Resultados	- Apresentação do mecanismo de questionamento (questionário, verbalização de procedimentos ou outra estratégia pré-definida) ao usuário de teste. - Associação de todas as informações coletadas (arquivos armazenados no sistema, anotações, questionários e/ou outro meio) ao usuário correspondente. - Elaboração imediata de um breve relatório do ensaio. - Elaboração do relatório final.

Sondagem da satisfação do usuário

Consiste na consulta ao usuário sobre a usabilidade do produto. Uma forma de obter estas informações consiste na utilização de estratégias envolvendo medições psicofisiológicas objetivas (ex.: dilatação de pupila, ritmo cardíaco, condutividade da pele, entre outros exames). Apesar de ser uma alternativa, ela torna-se inviável na medida em que exige equipamentos, amostragens do usuário e outras abordagens que podem ser intimidadoras, deixando o usuário tenso e pouco à vontade para utilizar o produto [Quei01].

Outra alternativa vem a ser o uso de questionários como forma de sondar a satisfação subjetiva do usuário. Ainda segundo Queiroz [Quei01], quando observado apenas o ponto de vista de um usuário, pode-se ter a aquisição de informações puramente subjetivas, porém quando ponderadas as opiniões de um grupo mais amplo de usuários, pode-se inferir sobre um contexto mais amplo e objetivo. O MCI utiliza-se do questionário OpUS (Opinião do Usuário de Software) para realizar a sondagem da satisfação do usuário.

Ferramentas de coleta de dados

Inicialmente, cita-se o uso do **FAIUnix** [Sous99] como ferramenta de apoio à avaliação de interfaces para o ambiente UNIX, permitindo a coleta de informações estáticas¹⁴ e dinâmicas¹⁵. Entretanto, assim como a ferramenta citada para a prototipação, também esta não está publicamente disponível, além de ser para uso restrito a ambiente UNIX.

¹⁴ Informações coletadas sem a participação do usuário [Sous99].

¹⁵ Informações coletadas a partir da interação do usuário com o protótipo [Sous99].

Nesta mesma linha, tem-se as ferramentas **FAIdos** e **FAIwin** que foram desenvolvidas por alunos que integraram o GIHM e objetivavam a coleta de informações dos testes de usabilidade.

2.5 CONSIDERAÇÕES

O presente capítulo abordou quatro métodos de concepção, sendo dois desenvolvidos no GIHM (MEDITE e MCI) e os métodos ADEPT e TRIDENT.

Os métodos ADEPT e TRIDENT possuem suporte computacional associado. Porém, mesmo sendo metodologias amplamente conhecidas pela comunidade de IHC, somente os grupos de pesquisas que os desenvolveram possuem acesso a este suporte computacional. Outro ponto desfavorável é a dificuldade na obtenção de informações, tanto sobre os formalismos utilizados quanto sobre as etapas constituintes dos métodos.

As metodologias do GIHM, MEDITE e MCI não oferecem ferramental que apóie suas etapas. Por outro lado, as informações sobre suas etapas podem ser obtidas facilmente. Ambas fazem uso de regras ergonômicas, porém MEDITE de forma explícita e o MCI de forma implícita (heurística) ou explícita (uso de padrões, diretrizes, entre outros). Por outro lado, MEDITE não explicita as etapas de Análise e Modelagem do Perfil do Usuário e do Contexto, enquanto que o MCI o faz. Este aspecto é importante para o presente trabalho, uma vez que a apuração de suportes computacionais exige que o método explicita claramente todas as etapas como o faz o MCI.

O levantamento e análise comparativa dos formalismos para a modelagem da tarefa se fez necessário, uma vez que o MCI adota o formalismo MAD e este não oferece ferramental de suporte. Cabia então pesquisar outros formalismos e respectivas ferramentas. Do estudo realizado, conclui-se que diversos formalismos podem ser utilizados. No entanto, o mais recomendado é o CTT, em virtude do seu suporte ferramental mais estável e da sua ampla variedade de operadores temporais, como será discutido no próximo capítulo.

Capítulo 3 - Suportes Computacionais

3.1 INTRODUÇÃO

O desenvolvimento de suportes computacionais visando reduzir a carga de trabalho do ser humano, suprir deficiências ou mesmo buscando a automatização de processos já se tornou uma constante nos meios acadêmicos e comerciais. Entretanto, quando o assunto é o uso de computadores e, conseqüentemente, o desenvolvimento de suportes computacionais que serão utilizados por pessoas, o caminho parece estar apenas começando.

O presente capítulo inicia com a apresentação dos critérios que nortearam o levantamento realizado, seguido pela apresentação do conjunto apurado.

Tendo em vista simplificar a classificação, bem como permitir maior abrangência de material apurado, optou-se por subdividir o conjunto em dois, a citar:

- Ambientes: fornecem apoio a um número mais amplo de etapas, bem como à busca pela obtenção de um produto final concreto;
- Ferramentas: fornecem apoio a um número reduzido de etapas.

Ao término da apresentação de cada grupo, apresentam-se considerações e quadros comparativos entre estes suportes e as etapas do MCI.

Conclui-se o capítulo com a apresentação dos suportes mais indicados para adoção e considerações a respeito.

3.2 CRITÉRIOS

A determinação de critérios possibilita tanto uma simplificação como uma maior concentração de esforços em torno de um foco específico.

Tal concentração de esforços é de extrema importância quando o conjunto de dados é aquele composto por sistemas informáticos.

Focalizar esforços em aparatos desenvolvidos para a área de interface homem-máquina já reduz expressivamente o campo de busca. Entretanto, optou-se por uma maior especificidade.

Sendo assim, elencou-se o seguinte conjunto de critérios norteadores para as ferramentas e ambientes pesquisados:

- deve estar disponível para uso imediato;
- deve, preferencialmente, ser de livre uso (*freeware*);
- deve ser compatível com o método MCI;
- deve disponibilizar meios que facilitem a integração com outras ferramentas, a exemplo de XML como meio de persistência dos artefatos de entrada e saída.

Outras características desejadas são:

- deve ser multiplataforma; e
- a depender da função realizada, deve ser apoiada por notação de valor reconhecido.

As informações obtidas da pesquisa foram sintetizadas em um formato tabular, cujo *template* é ilustrado no Quadro 8.

Quadro 8 - Template para registro das informações apuradas para um suporte computacional

Nome da Ferramenta	Nome da ferramenta.
Autor ou Grupo de Pesquisa Responsável	Responsável pela autoria e manutenção da ferramenta.
Status / Versão	Referente ao status atual da ferramenta.
Licença	Tipo de licença de uso associada à ferramenta.
Idioma	Idioma em que a ferramenta se apresenta.
Plataforma	Qual a plataforma de funcionamento da ferramenta.
Linguagem em que foi desenvolvida	Qual a linguagem de programação utilizada no desenvolvimento.
Notação associada	Quando for o caso, qual a notação associada à ferramenta.
Forma de entrada de dados	Qual(is) formatos de dados são aceitos pela ferramenta.
Forma de saída de dados	Qual(is) formatos de dados são fornecidos pela ferramenta.
Site associado	Endereço eletrônico disponível sobre a ferramenta.
Material bibliográfico associado	Lista de referências bibliográficas sobre a ferramenta.

Quanto ao campo **Status / Versão**, cabe salientar que serão utilizadas as seguintes classificações possíveis:

- *Indisponível para aquisição e uso*, quando se encontra em desenvolvimento ou já foi desenvolvido, porém não foi disponibilizado;

- *Disponível para uso*, encontra-se disponível, porém não explicita a atual versão;
- *Disponibilidade parcial das ferramentas*, o referido suporte possui algumas ferramentas disponíveis e outras indisponíveis;
- *Número da versão*, explicita a versão atual, neste caso é apresentado o número da versão disponível.

Campos que estiverem marcados por “-”, subentende-se que as referidas informações estão indisponíveis.

3.3 AMBIENTES

Como anteriormente exposto, os aparatos computacionais aqui agrupados possuem como características marcantes (I) o apoio amplo às etapas de concepção e (II) a geração de protótipo ao final do processo.

3.3.1 JMVCase

Segundo Barrère et al. [Barr99], o ambiente Java-MVCase “suporta a especificação de requisitos de sistemas em diferentes métodos orientados a objetos” com a “implementação automática desses sistemas em Java”.

Entretanto, tanto o suporte à especificação de requisitos quanto à geração automática do protótipo são realizados através de três ferramentas subordinadas ao ambiente principal, que são:

- **JavaRC**, responsável pela especificação de requisitos;
- **Draco**, responsável pela geração automática do sistema;
- **Visual Café dbDE**, utilizada para a construção das interfaces do sistema.

3.3.1.1 Processo

O passo inicial da ferramenta consiste na especificação e modelagem dos requisitos através da ferramenta JavaRC. Segundo exposto em [Barr99], o ambiente disponibiliza três métodos de modelagem: Coad/Yourdon, Fusion ou UML, bastando ao projetista escolher o que mais lhe agrada.

Segundo os autores, um ponto de destaque para a ferramenta JavaRC (Figura 13) consiste na conversão de modelos para outros métodos, ou seja, o modelo gerado segundo o método Fusion pode ser convertido para UML¹⁶.

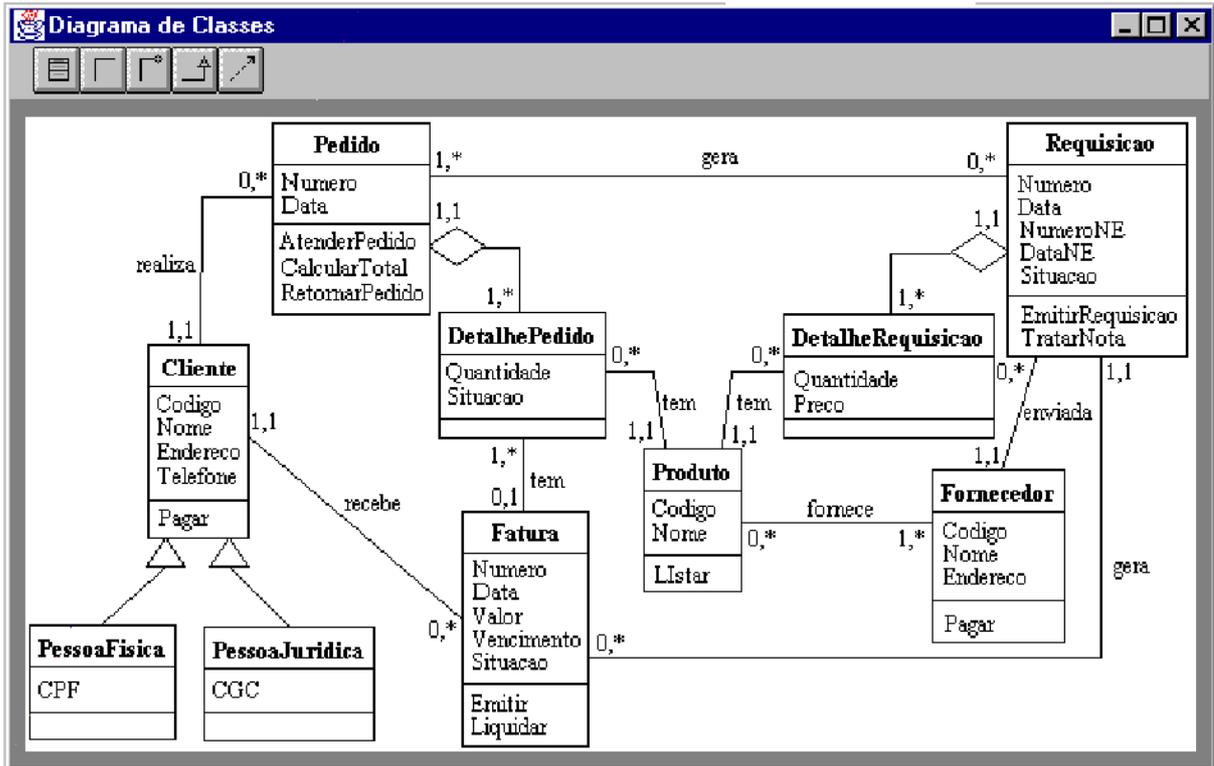


Figura 13 - Modelagem de sistemas através da ferramenta JavaRC [Barr99]

Os sistemas dos modelos gerados (Figura 14 - parte 1) são armazenados segundo uma linguagem baseada na Representação Canônica (RC), enquanto a descrição do comportamento dos objetos é armazenada segundo a Linguagem Baseada de Ensino (LBE).

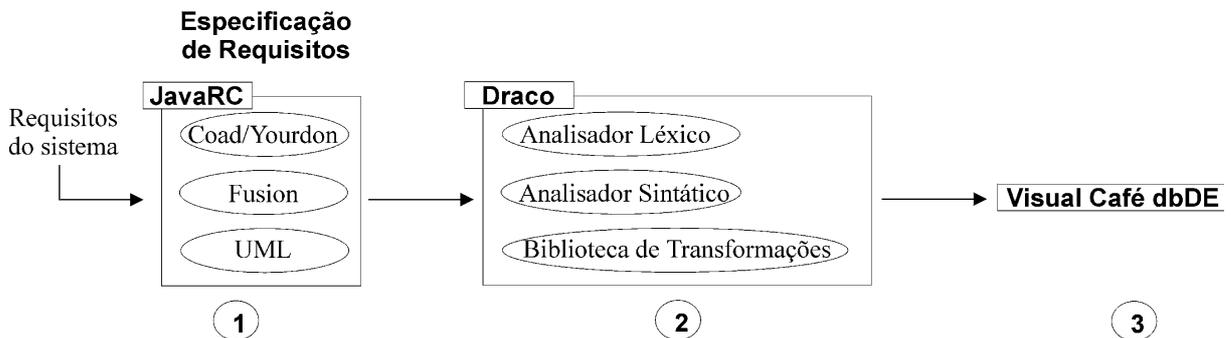


Figura 14 - Integração dos componentes da ferramenta [Barr99]

¹⁶ UML: Unified Modeling Language.

A ferramenta Draco (Figura 14 - parte 2) consiste em um sistema de transformações de software orientado a domínios e é responsável pela implementação automática do sistema a partir das descrições geradas na etapa anterior.

O sistema parte das descrições RC e LBE e, através da aplicação de analisadores léxicos e sintáticos, são obtidas novas descrições que, por sua vez, são transformadas em código Java e especificações SQL.

Por fim, utiliza-se a ferramenta Visual Café dbDE (Figura 14 - parte 3) para a concepção da interface, que possui todo o aparato necessário.

3.3.1.2 Considerações

Apesar do ambiente apresentado em [Barr99] parecer bastante promissor, este não se encontra disponível para aquisição e uso. Quanto às ferramentas apresentadas, tanto JavaRC quanto Draco seguem a linha do ambiente e não estão disponíveis. Já a ferramenta Visual Café dbDE é de cunho comercial, ficando, portanto, fora dos critérios pré-estabelecidos.

Outro ponto desfavorável ao ambiente é quanto à inexistência de um método de concepção que o suporte, o que implica sobrecarga de responsabilidades na equipe de desenvolvimento.

Quadro 9 - Resumo sobre JMVCASE

Nome da Ferramenta	JMVCASE
Autor ou Grupo de Pesquisa Responsável	Tathiana da Silva Barrére Departamento de Computação Universidade Federal de São Carlos
Status / Versão	Indisponível para aquisição e uso
Licença	-
Idioma	-
Plataforma	-
Linguagem em que foi desenvolvida	-
Notação associada	Orientada a Objetos
Forma de entrada de dados	-
Forma de saída de dados	-
Site associado	-
Material bibliográfico associado	[Barr99]

3.3.2 DIGBE (Dynamic Interaction Generation for Building Environments)

Protótipo desenvolvido na Universidade de Minnesota e que objetiva automatizar interfaces de sistemas baseando-se no dispositivo selecionado pelo usuário e adaptando a

interação em tempo real. Segundo os autores [Penn00], o protótipo está habilitado a gerar automaticamente interfaces para monitores (CRT) ou *handheld Palm*.

Na realidade, o protótipo apresentado no artigo [Penn00] baseia-se no DIG (*Dynamic Interaction Generation*) sendo o DIGBE uma extensão do primeiro.

A razão principal de uso do DIG concentra-se no projeto de interfaces de controles digitais, nos quais, segundo Penner e Steinmetz [Penn00], os modelos de domínios com informações sobre tipos de dados e regras são possíveis de serem construídos devido a seu largo uso e, conseqüentemente, extenso arcabouço de dados já existentes.

DIGBE, portanto, “é a implementação de um conjunto de automação modular que automaticamente e dinamicamente concebem e criam interfaces com o usuário para grandes sistemas e as associa com sistemas de controle” [Penn00].

3.3.2.1 Processo

Sendo a ferramenta um automatizador de criação de interfaces, seu núcleo de processo se encontra no conhecimento que os Agentes de Interação possuem sobre o projeto de dinâmica de interação. Este “conhecimento” está contido no Modelo de Tarefa e Interação (*Task and Interaction Model - TIM*).

O TIM, por sua vez, é composto por cinco níveis, organizados de forma hierárquica. A Figura 15 apresenta um subconjunto de um Gerenciador de Sistemas modelado em TIM.

Os níveis existentes são:

- **aplicações**, tais como Gerenciadores de Sistemas, que por sua vez são compostos de
 - **tarefas**, tal como Monitoramento, que são compostos de
 - **subtarefas**, tal como Seleção, e estas subtarefas são compostas de
 - **elementos**, como Conjuntos Discretos, que são compostos de
 - **primitivas**, que são as unidades básicas da interação, tais como Seletores ou Entradas de Textos.

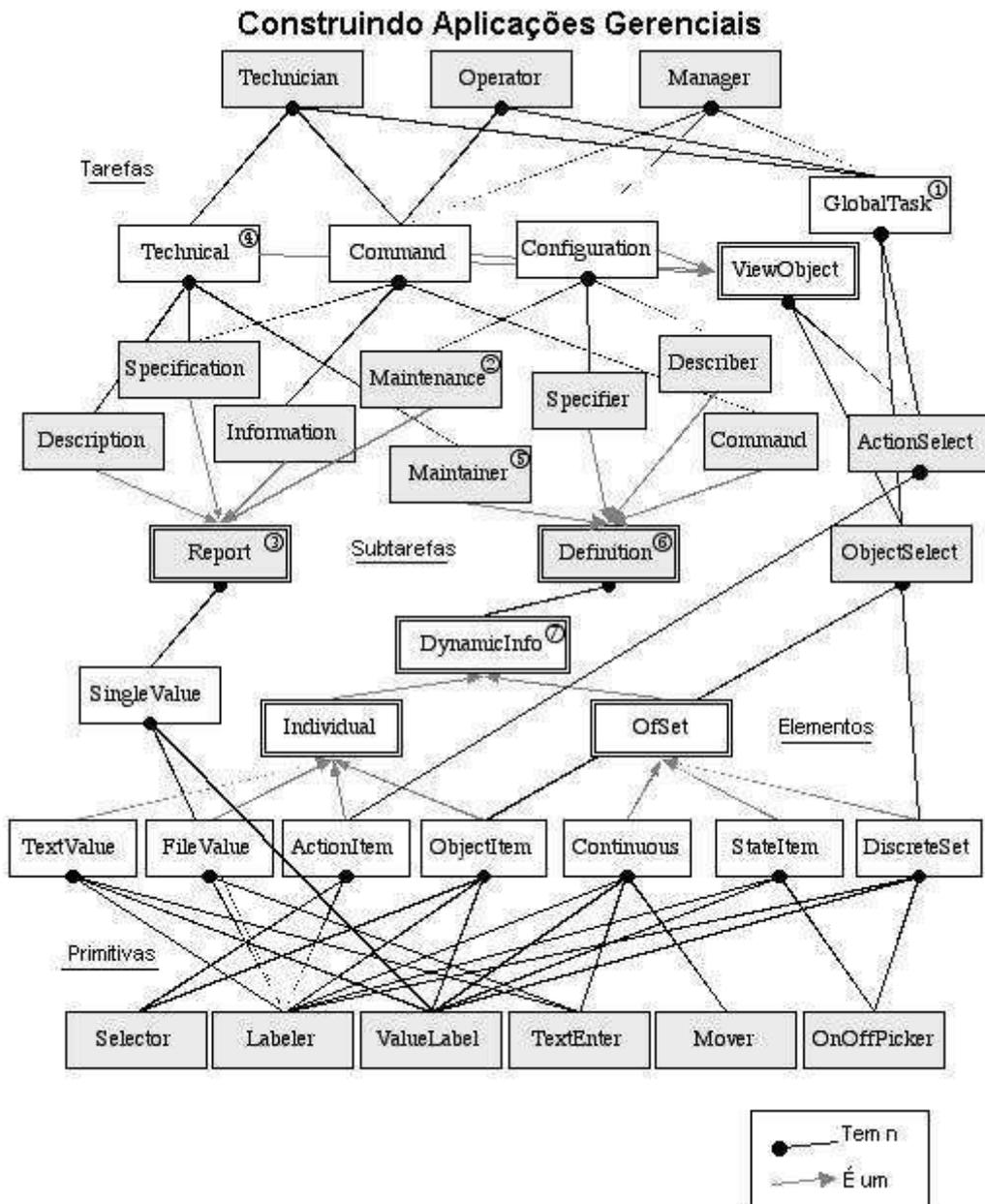


Figura 15 - Modelo de Tarefa e Interação - TIM [Penn00]

Penner e Steinmetz [Penn00] indicam que o TIM é um sistema produtivo de autocomposição, de forma a tornar possível a criação ou modificação em tempo real do modo ou de parte da interface de controle. Pode-se observar este processo na Figura 16, onde é representada a forma como é gerada a interface que será utilizada por um técnico.

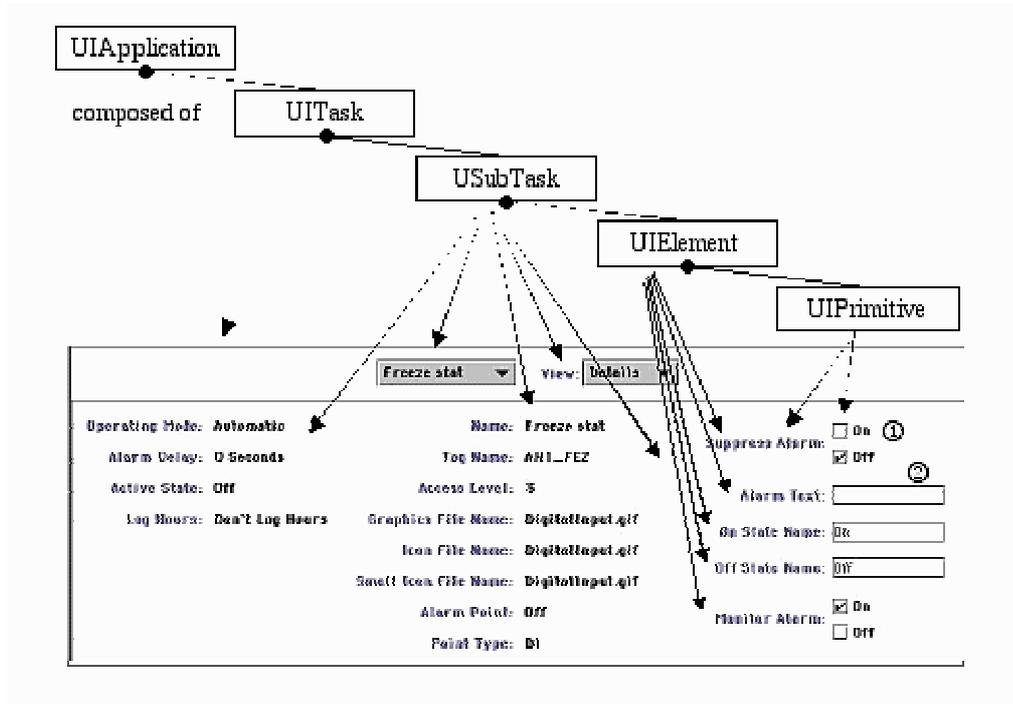


Figura 16 - Geração da interface do usuário Técnico [Penn00]

3.3.2.2 Considerações

Apesar da escassez de informações disponível em [Penn00], pode-se inferir que o ambiente DIGBE embasa-se no Modelo de Domínio [Penn00] e para funcionar corretamente deve estar instalado e em contato direto com os todos os dados do sistema. Somente assim será possível gerar dinamicamente as interfaces para cada tipo de usuário.

Subtende-se, portanto, que o ambiente seja uma espécie de interpretador.

Porém não há informações acerca de quais formalismos estão associados ao ambiente, assim como não são apresentadas informações acerca das ferramentas que o constituem.

Quadro 10 - Resumo sobre DIGBE

Nome da Ferramenta	DIGBE
Autor ou Grupo de Pesquisa Responsável	Universidade de Minnesota
Status / Versão	Indisponível para aquisição e uso
Licença	-
Idioma	-
Plataforma	-
Linguagem em que foi desenvolvida	-
Notação associada	-
Forma de entrada de dados	-
Forma de saída de dados	-
Site associado	-
Material bibliográfico associado	[Penn00]

3.3.3 INGRID (*INteractive GRaphical Interface Designer*)

Ferramenta desenvolvida na linguagem C++ pelo INESC¹⁷, objetivando a construção interativa de interfaces com o usuário [Carr90, Guim91], baseada em dois modelos fundamentais:

- um modelo global para a aplicação;
- um modelo para os componentes da interface com o usuário.

O modelo global para a aplicação segue a linha da independência de diálogo, de forma que interface com o usuário e a aplicação sejam componentes distintos.

Já o modelo da interface refere-se aos aspectos funcionais inerentes à interface, através do modelo 4D (*Display, Data, Dialogue e Driver*) (ver Figura 17) que se assemelha à sistemática existente no modelo PAC [Cout87, Cout90] ou do modelo EDITOR [Lula92].

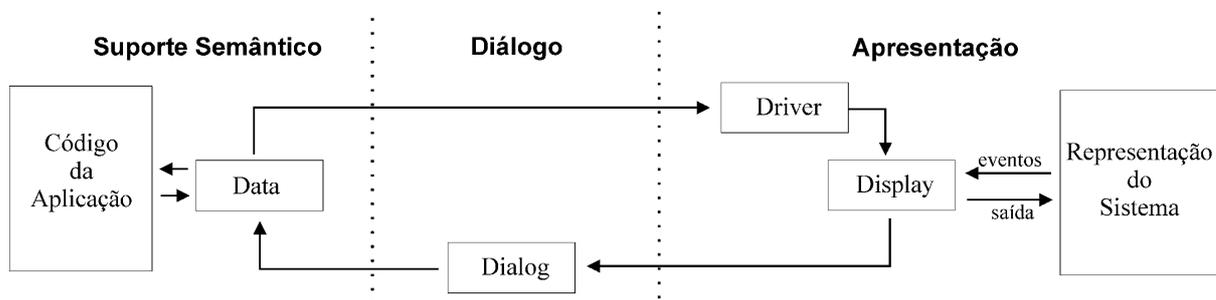


Figura 17 - Representação gráfica do Modelo 4D [Guim91]

Componentes do modelo:

- **Display**, composto pela apresentação gráfica (entrada e saída) da interface, tais como botões, menus e *scrollbars*.
- **Data**, composto pelos tipos abstratos de informações da aplicação, tais como inteiros, arquivo ou lista. Segundo os autores, o objetivo da existência de objetos **Data** é prover uma conexão clara entre a interface do usuário e as partes computacionais, além de prover informação semântica à interface do usuário.
- **Dialog**, contém a estrutura sintática da interação entre usuário e computador.
- **Driver**, é um conversor entre os dados de *Data* e *Display*.

¹⁷ INESC: Instituto de Engenharia de Sistemas e Computadores.

3.3.3.1 Processo

O ambiente INGRID tem como tela inicial o *Interface Organizer* (Figura 18). Nesta tela o projetista tem acesso aos objetos da interface, bem como pode gerenciar o projeto.

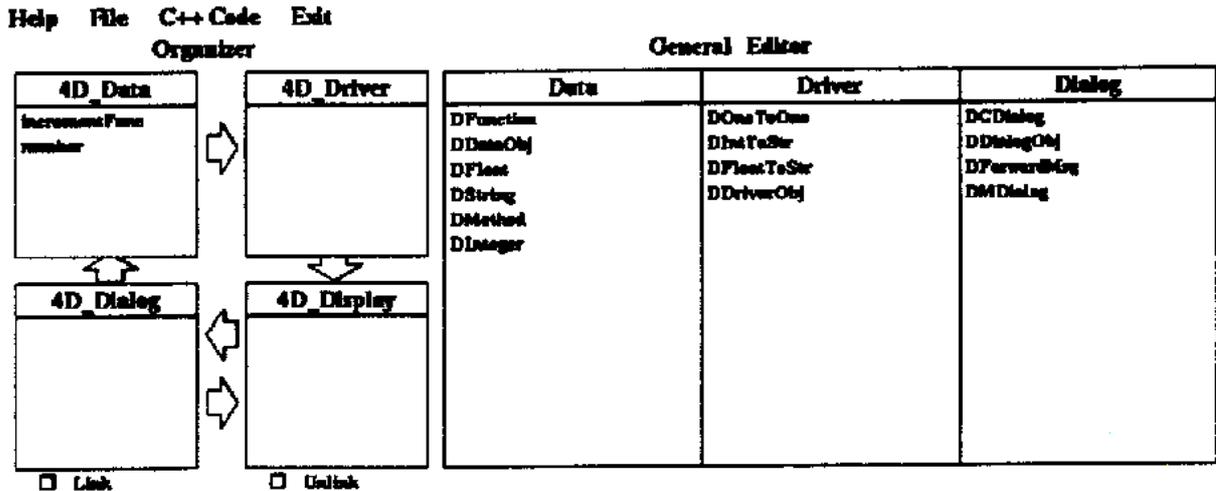


Figura 18 - Interface Organizer [Guim91]

Já o *Display Editor* (Figura 19) permite ao projetista controlar propriedades de telas como posição, tamanho, textos ou *bitmaps*.

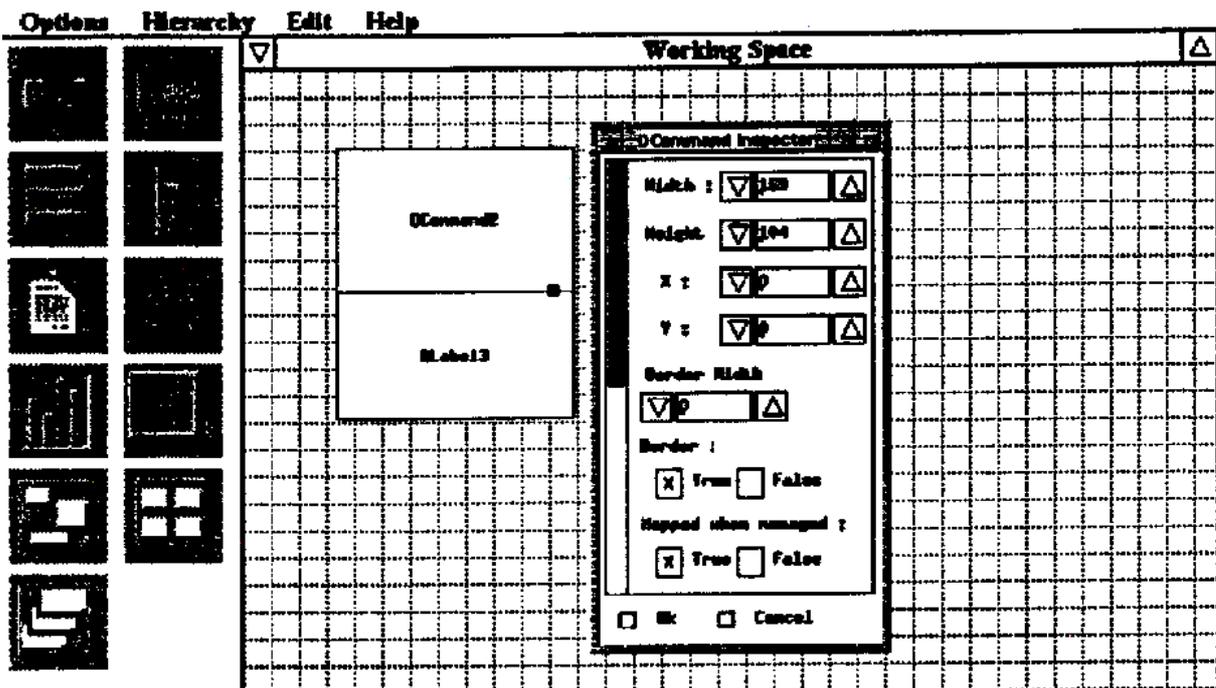


Figura 19 - Display Editor [Guim91]

Há ainda três editores: *Data*, *Driver* e *Dialogue Editor* que são utilizados para associar os tipos de dados, instanciar *driver objects* e auxiliar na definição dos diálogos de interação, respectivamente.

3.3.3.2 Considerações

Como pontos favoráveis da ferramenta citam-se o uso de uma metodologia similar ao PAC e o uso da abordagem de independência do diálogo.

Entretanto, o ambiente não está associado diretamente a nenhum método de concepção, bem como não está disponível para aquisição.

Quadro 11 - Resumo sobre INGRID

Nome da Ferramenta	INGRID
Autor ou Grupo de Pesquisa Responsável	INESC
Status / Versão	Indisponível para aquisição e uso
Licença	-
Idioma	-
Plataforma	-
Linguagem em que foi desenvolvida	C++
Notação associada	-
Forma de entrada de dados	-
Forma de saída de dados	-
Site associado	-
Material bibliográfico associado	[Carr90, Guim91]

3.3.4 Teallach

De um modo geral, aplicações que envolvem a interação com bases de dados, podem ser caracterizadas por processarem e analisarem modelos ricos em informações e com grandes volumes de dados [Barc99]. O Teallach é fruto da parceria entre Universidades¹⁸ e Empresas¹⁹ e consiste em um sistema desenvolvido em Java objetivando fornecer suporte ao desenvolvimento de interfaces para objetos de base de dados, através de “um método flexível de *design* de forma que os *designers* não fiquem restritos a apenas uma estratégia de desenvolvimento” [Barc99].

O ambiente, através de três modelos, possibilita o desenvolvimento de interfaces.

- **Modelo do Domínio**, descreve a aplicação em termos de dados e operações;
- **Modelo da Tarefa**, descreve o que o usuário pode realizar em termos da dinâmica e processamento das informações;

¹⁸ University of Glasgow, Manchester University, Napier University.

¹⁹ British Geological Survey, Criterion Software, IBM, ICL.

- **Modelo da Apresentação**, indica como a interface será apresentada para o usuário.

Segundo exposto pelos autores em [Grif01], “cada um dos modelos define uma visão da informação requerida para a geração da interface para uma aplicação em particular”.

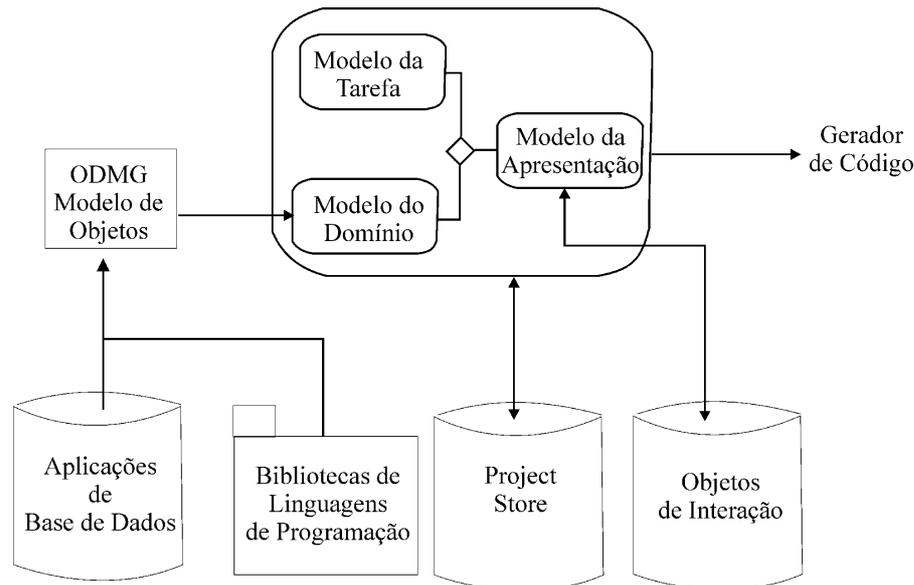


Figura 20 - Representação da arquitetura com os modelos [Grif01]

Em [Grif01], é apresentado um modelo de arquitetura e sua relação com os modelos utilizados no ambiente (Figura 20).

Na Figura 21 observa-se o método apresentado em [Barc99], no qual, os fluxos em linha cheia representam as ações do usuário e os fluxos tracejados representam as ações do sistema. Toda descrição de fluxo que contenha “*”, indica que a ação é repetitiva.

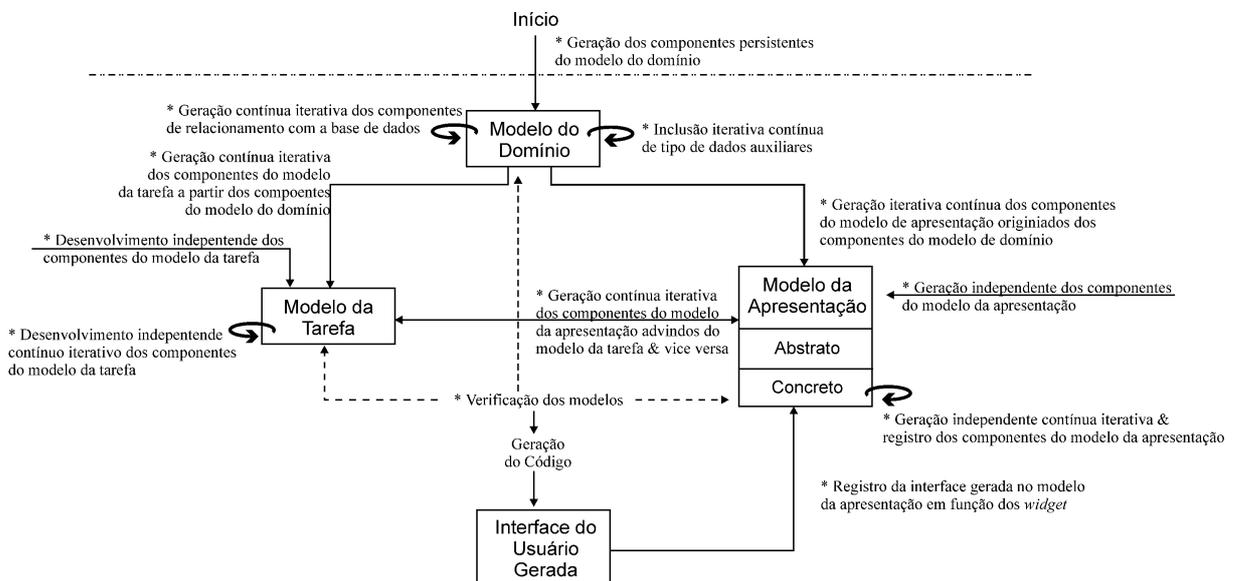


Figura 21 - Método Teallach [Barc99]

3.3.4.1 Processo

Modelo do Domínio

Criado em 1991, o *Object Database Management Group* (ODMG) busca a padronização de Sistemas de Gestão de Bases de Dados Orientados por Objetos (SGBDOO) objetivando expandir o mercado de aplicação destes sistemas. O Grupo pretende obter uma descrição padronizada composta por quatro componentes:

- Modelo Objeto;
- Linguagem de Especificação;
- Linguagem de Consulta;
- Definição de conexões com linguagens de programação.

O Teallach utiliza este conjunto de modelos como ponto de partida na criação das interfaces. Observa-se na Figura 22 - A, a tela do editor do modelo do domínio, a qual é subdividido em quatro painéis:

- **Persistent**, representa os tipos de dados persistentes;
- **Database**, representa funcionalidades genéricas da base de dados, tais como gerenciamento de transações e pesquisas (*query*);
- **Auxiliary**, prove funcionalidades arbitrárias a serem usadas no desenvolvimento da interface;
- **Java API**, biblioteca de classes visando expandir as funcionalidades no projeto da interface.

Modelo da Tarefa

O editor (Figura 22 - B), presente na ferramenta, permite a construção e edição de modelos hierárquicos de tarefas, com os objetivos e sub-objetivos do usuário são representados em ordem temporal de execução.

- Segundo apresentado em [Barc99], inicialmente o *designer* constrói uma especificação em alto nível das tarefas, para em seguida acrescentar as ações e interações entre tarefas. Desta forma, há uma representação de como a aplicação processa a informação e como o usuário final e a aplicação interagem.

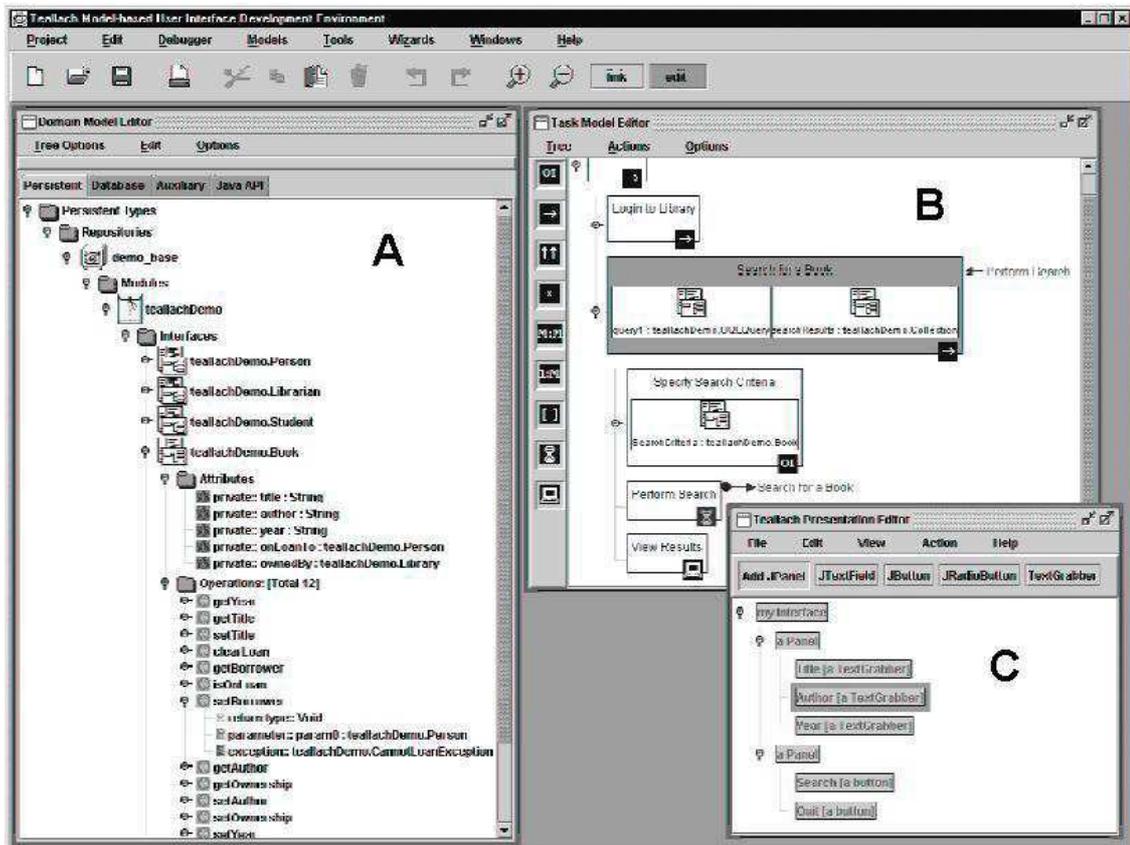


Figura 22 - Editores Teallach [Barc99]

Modelo da Apresentação

A ferramenta disponibiliza um editor de apresentação (Figura 22 - C) com dois níveis de abstração:

- *abstract presentation model* (modelo abstrato), onde são definidas categorias genéricas de *widgets*;
- *concrete presentation model* (modelo concreto), onde é realizada a definição de *widgets* concretos, com base nos apresentados pela categoria selecionada no modelo abstrato.

Segundo o material de referência o ambiente ainda permite pré-visualizar (*preview*) a interface criada.

3.3.4.2 Considerações

A abordagem adotada pelo ambiente permite obter, através de informações existentes nas bases de dados baseadas em ODMG, uma interface completa e funcional.

Segundo exposto pelos desenvolvedores, primou-se por um ambiente que não engessasse a concepção, assim, inicia-se pelo modelo de domínio e as etapas seguintes podem ser tanto

a modelagem da tarefa quanto da apresentação. Embora esta liberdade seja útil para equipes de desenvolvimento experientes, pode ser inconveniente para equipes inexperientes, uma vez que o processo não delimita uma seqüência definida de etapas em direção à obtenção da interface final.

Outro ponto desfavorável diz respeito aos modelos utilizados, excetuando-se o modelo do domínio, os demais não possuem um formalismo associado. Quanto às ferramentas citadas, todas são de uso exclusivo do grupo que as desenvolveu.

Quadro 12 - Resumo sobre Teallach

Nome da Ferramenta	Teallach
Autor ou Grupo de Pesquisa Responsável	Parceria entre Universidades e Empresas
Status / Versão	Indisponível para aquisição e uso
Licença	-
Idioma	Inglês
Plataforma	-
Linguagem em que foi desenvolvida	Java
Notação associada	ODMG ²⁰ (Modelo do Domínio) Modelos hierárquicos para tarefa
Forma de entrada de dados	-
Forma de saída de dados	-
Site associado	http://www.dcs.gla.ac.uk/research/teallach/
Material bibliográfico associado	[Barc99, Grif99, Grif01]

3.3.5 CTTE + Petshop

Fruto da parceria entre CNUCE-CNR e LIHS (Universidade de Toulouse 1) o ambiente é baseado em duas notações distintas (CTT²¹ para a modelagem da tarefa e ICO²² para a modelagem do sistema) e objetiva o suporte ao desenvolvimento de sistemas interativos.

3.3.5.1 Processo

Uma vez que tanto a notação CTT quanto a ICO já possuem suporte computacional individual (CTTE e PetShop, respectivamente), uma ferramenta integradora foi desenvolvida pelo LIHS (Figura 23) [Nava01].

²⁰ ODMG: *object database standard*.

²¹ CTT: ConcurTaskTrees.

²² ICO: *Interactive Cooperative Objects*.

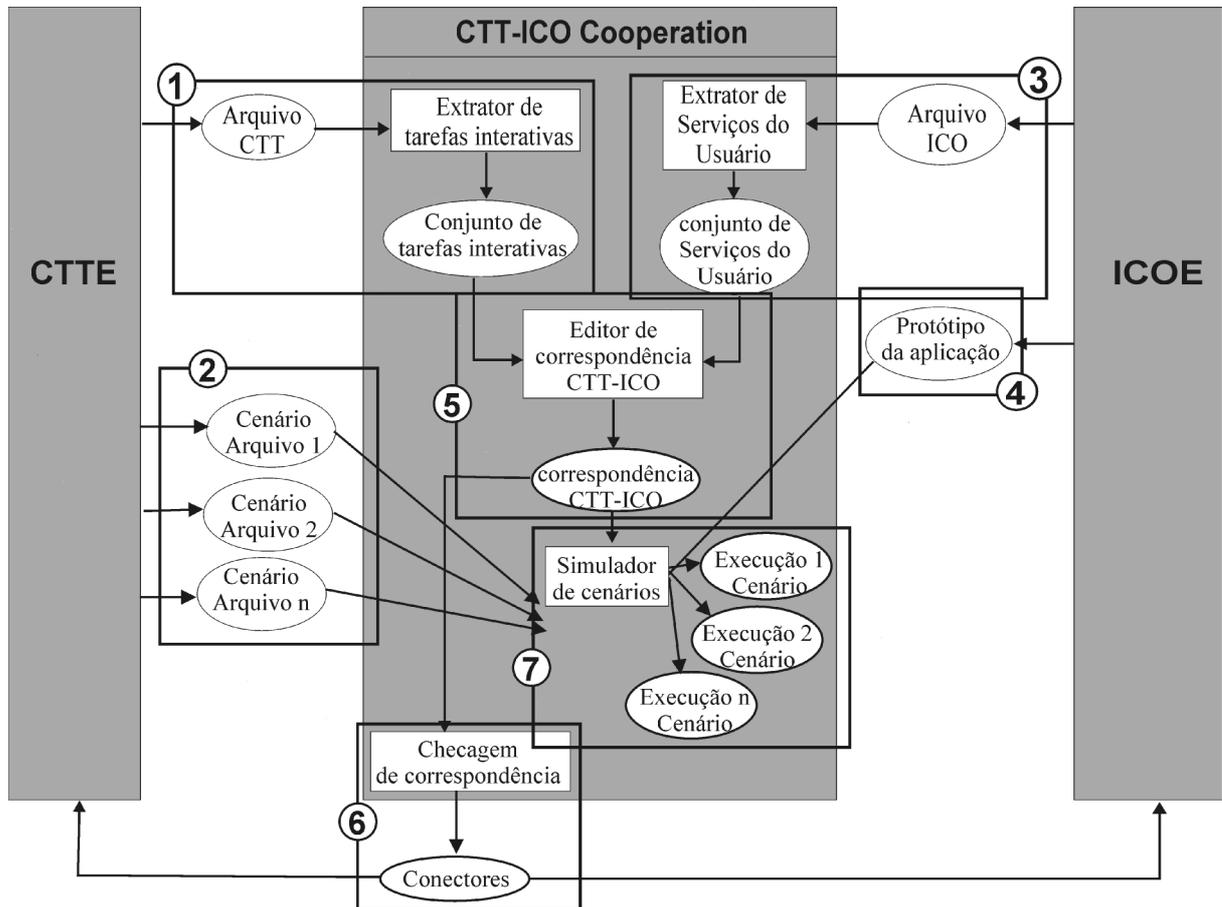


Figura 23 - Framework de integração CTTE e PetShop [Nava01]

CTTE

Ferramenta desenvolvida pelo CNUCE-CNR baseada no formalismo CTT que através de uma interface gráfica permite a modelagem da tarefa (Figura 24) [Pate97, Pate01, Mori02, Pate02].

Outras funções de destaque da ferramenta são os recursos para:

- Simulação do modelo gerado;
- Obtenção de dados estatísticos sobre o modelo;
- Elaboração de cenários de uso.

PetShop

Ambiente composto por ferramentas que possibilitam a modelagem de sistemas segundo modelos propostos pelo padrão CORBA [Sy00]. A Figura 25 representa a arquitetura geral do ambiente, na qual os retângulos representam os módulos funcionais enquanto que as elipses representam os modelos gerados e utilizados pelo sistema [Nava01].

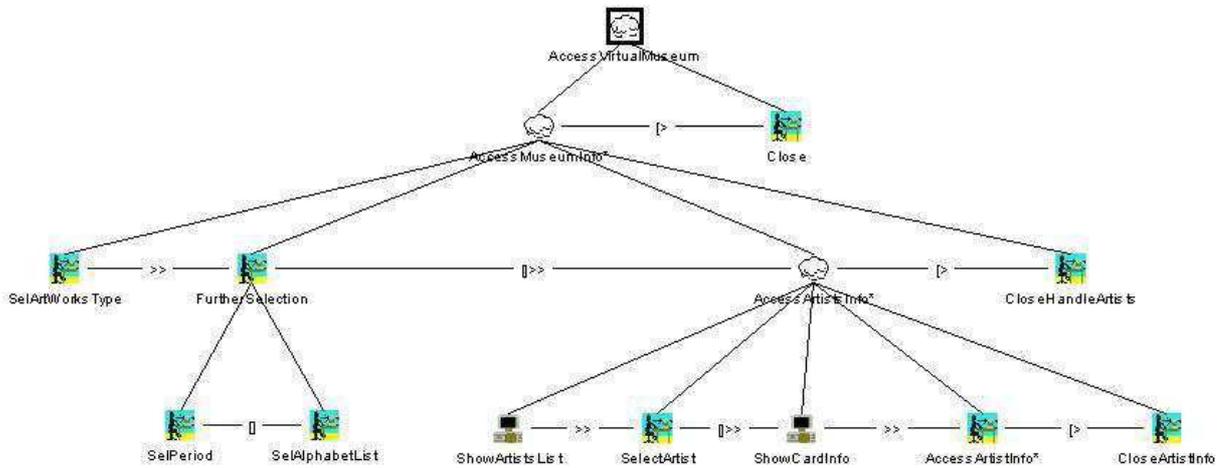


Figura 24 - Árvore de tarefas gerada no CTTE

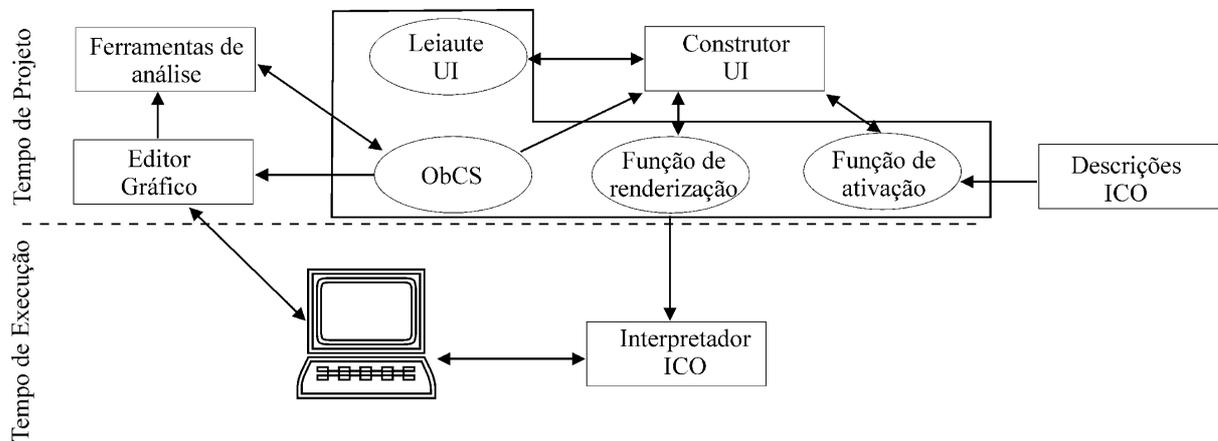


Figura 25 - Arquitetura do ambiente PetShop [Nava01]

Integração entre CTTE e Petshop

A Figura 23 permite observar os pontos de integração entre o CTTE e o Petshop, retângulos 1, 2 e 3, 4, respectivamente.

Como pode ser observado, aproveita-se não só os arquivos gerados por cada ferramenta (retângulos 1 e 3), como também os cenários gerados pelo CTTE (retângulo 2) e o protótipo de aplicação gerado pelo ICOE (retângulo 4).

As etapas marcadas pelos retângulos 5 e 6 são referentes ao **Editor de Correspondência** (Figura 26).

A primeira etapa de integração consiste em relacionar as *tarefas* interativas do modelo de tarefas com os *serviços do usuário* do modelo do sistema (retângulo 5 da Figura 23). Desta forma, há a possibilidade de refinar o modelo de tarefa de maneira a representar as interações em um nível mais baixo.

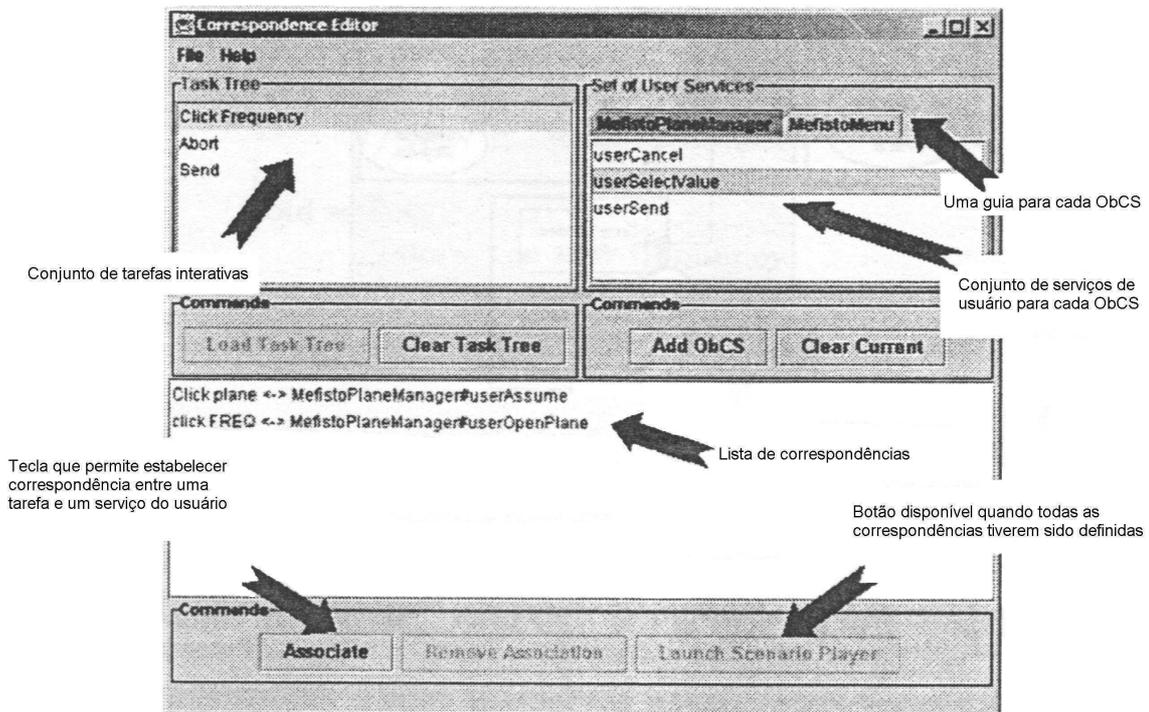


Figura 26 - Editor de Correspondência [Nava01]

Visando verificar se a correspondência realizada é válida, foi criada a *verificação de modelo* (retângulo 6 da Figura 23) que objetiva avaliar tanto a integração das ferramentas em termos de modelos quanto em termos de aplicação.

Navarre et al. apontam para o seguinte conjunto de questões, a ser formulado pela equipe de desenvolvimento, como forma de mensurar se as modelagens da tarefa e do sistema estão completas:

- “foram modeladas ao menos tantos *serviços de usuários* quanto *tarefas interativas*?”;
- “foram modelados todos os cenários possíveis?”.
- Já em termos de aplicação, cabem os seguintes questionamentos:
- “existem mais *serviços de usuários* na especificação ICO do que *tarefas interativas* no modelo CTT?”;
- “existem cenários no modelo de sistema que não estão disponíveis no modelo da tarefa?”.

3.3.5.2 Considerações

O produto resultante das somas de esforços é um ambiente que possibilita a geração de sistemas completos, funcionais e centrados no usuário. Outro ponto favorável diz respeito à linguagem utilizada no desenvolvimento: Java, o que possibilita o uso em multiplataformas.

Entretanto, apesar do ambiente dispor de obtenção de aplicações centradas no usuário, não há um método de concepção que o apóie, desta forma, cabe ao desenvolvedor a responsabilidade da escolha e aplicação de um método.

Outro ponto desfavorável diz respeito ao acesso às ferramentas citadas, pois apenas uma delas - CTTE, está disponível para livre uso.

Quadro 13 - Resumo sobre CTTE + Petshop

Nome da Ferramenta	CTTE + Petshop
Autor ou Grupo de Pesquisa Responsável	CNUCE-CNR LIHS, Universidade de Toulouse 1
Status / Versão	Disponibilidade parcial das ferramentas
Licença	-
Idioma	-
Plataforma	-
Linguagem em que foi desenvolvida	Java
Notação associada	CTT ICO
Forma de entrada de dados	-
Forma de saída de dados	-
Site associado	http://giove.cnuce.cnr.it/mefisto.html
Material bibliográfico associado	[Nava01]

3.3.6 TeleUSE

Consistindo em um *User Interface Management System* (UIMS), o TeleUSE foi desenvolvido pela empresa Aonix e oferece todas as ferramentas necessárias para projetar interativamente interfaces em C, C++, UIL ou Ada95 [Aoni99].

3.3.6.1 Processo

O ambiente, apresentado na Figura 27, está organizado de forma a trabalhar com três camadas distintas, **Apresentação**, **Diálogo** e **Aplicação**. Segundo a empresa desenvolvedora, as duas primeiras são apoiadas por ferramentas de apoio.

O VIP, ou *Visual Interactive Presentation*, (Figura 28) consiste em uma ferramenta projetada para a simplificação do projeto de itens da interface. Segundo informações do

fabricante, pode-se utilizar conjuntos padronizados ou criar novos objetos através de um conjunto de controles.

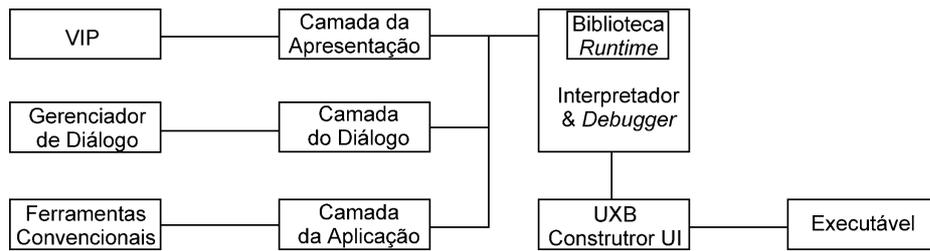


Figura 27 - Diagrama do TeleUSE

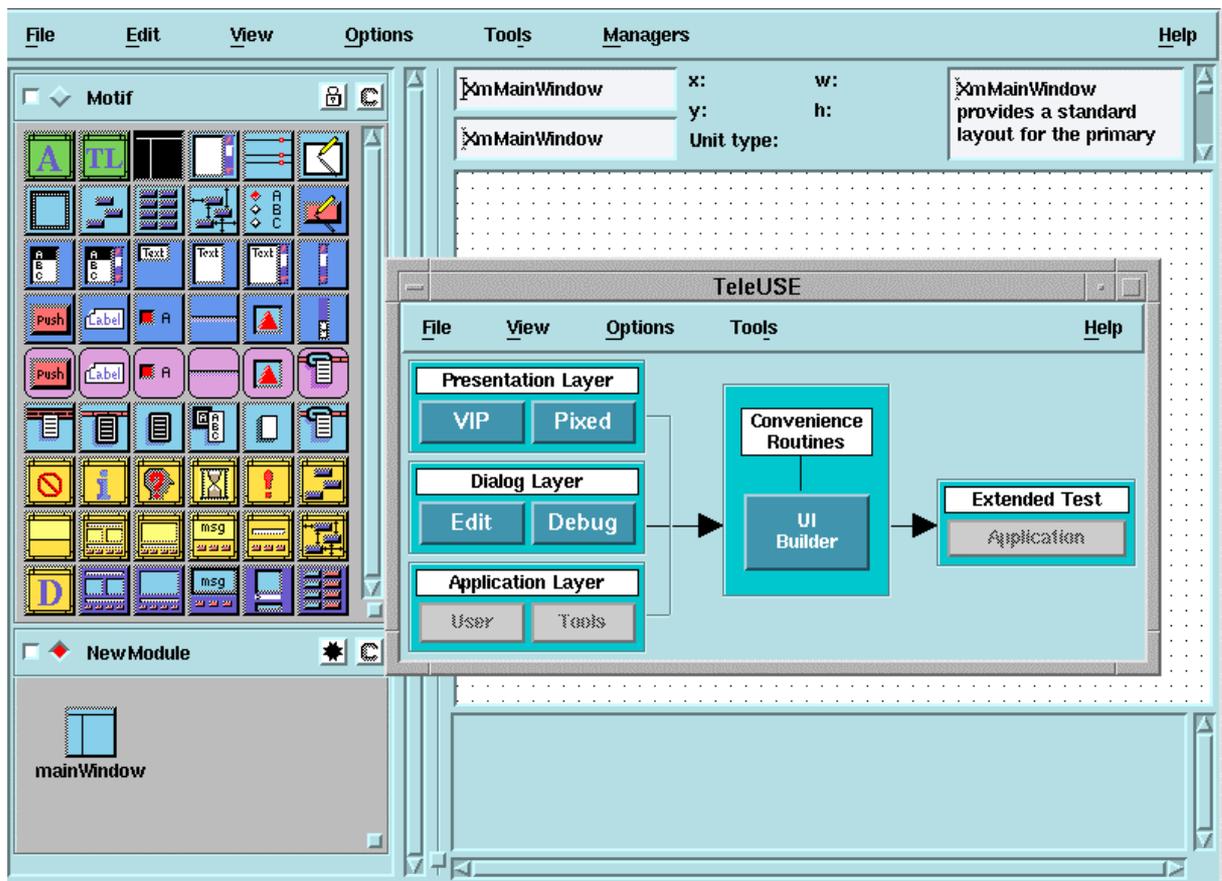


Figura 28 - VIP (Visual Interactive Presentation)

Já o **Gerenciador de Diálogo** permite a especificação, o gerenciamento e o controle da dinâmica ou diálogo entre a interface e o código da aplicação.

Uma vez fornecidas às ferramentas os dados necessários, o desenvolvedor pode testar seu produto através do interpretador, bem como realizar uma busca por erros. Um executável é gerado pelo uso de bibliotecas que permitem mapear os projetos desenvolvidos e transformá-los em classes.

3.3.6.2 Considerações

O ambiente parece bastante robusto e, segundo o fabricante, é utilizado por instituições de renome, tais como Citibank e Motorola. Entretanto, além de não haver uma explicitação acerca dos formalismos e métodos utilizados pela ferramenta, há também o fato da mesma ser comercializada.

Quadro 14 - Resumo sobre TeleUSE

Nome da Ferramenta	TeleUSE
Autor ou Grupo de Pesquisa Responsável	Aonix
Status / Versão	Disponível para uso
Licença	Comercial
Idioma	Inglês
Plataforma	Multiplataforma
Linguagem em que foi desenvolvida	-
Notação associada	Orientação Objetos
Forma de entrada de dados	-
Forma de saída de dados	-
Site associado	http://www.aonix.com
Material bibliográfico associado	[Aoni99]

3.4 RELAÇÃO ENTRE AMBIENTES E AS ETAPAS DO MCI

A exceção do TeleUSE, todos os demais ambientes pesquisados são de cunho acadêmico. O que pode ser observado é que apesar de desenvolvidos na academia, não há pronta disponibilidade para aquisição e uso. Ao contrário, a grande maioria encontra-se total ou parcialmente indisponível. Soma-se a isto, o fato de que a maioria não informa se há um processo de concepção associado que guie a equipe de desenvolvimento. Esta dificuldade é agravada pela disponibilidade de informações acerca de alguns dos ambientes, como pode ser observado pelo campo referente à bibliografia associada (presente nos quadros de resumo sobre cada ambiente).

Apesar dos ambientes pesquisados não explicitarem um método de concepção, propõe-se, a seguir, uma relação entre eles e as etapas constituintes do MCI, ilustrado no Quadro 15. Cabe observar que, em alguns casos, o apoio computacional é apenas **parcial**, o que significa que a ferramenta não realiza, de forma direta a referida etapa do MCI.

Do quadro, pode-se inferir quais etapas possuem maior grau de importância para os ambientes estudados. Observa-se que dos seis ambientes focalizados, cinco apresentam suporte (parcial ou completo) para a etapa de **Análise e Modelagem da Interação** e destes,

três apresentam suporte (parcial ou completo) para as etapas de **Análise e Modelagem da Tarefa e Avaliação do Modelo da Tarefa**.

Quadro 15 - Relação entre Ambientes e as Etapas do MCI

MCI		JVMCASE	DIGBE	INGRID	Teallach	CTTE+PetShop	TeleUSE
Análise de Requisitos	Análise e Modelagem do Perfil do Usuário	-	Sim	-	-	-	-
	Análise e Modelagem do Contexto	-	-	-	Parcial	-	-
	Análise e Modelagem da Tarefa	Parcial	Parcial	-	Parcial	Sim	-
	Avaliação do Modelo da Tarefa	-	Sim	-	Sim	Sim	-
Concepção do Modelo de Interação	Análise e Modelagem da Interação	-	Parcial	Parcial	Parcial	Sim	Sim
	Avaliação do Modelo de Interação	-	-	-	-	-	Sim
	Representação do Cenário	-	-	-	-	Sim	-
	Construção do Modelo de Navegação	-	-	-	-	-	-
	Avaliação do Modelo de Navegação	-	-	-	-	-	-
Construção e avaliação do protótipo	Geração do Protótipo	Sim	Sim	Sim	Sim	Sim	Sim
	Avaliação do Protótipo	-	-	-	-	-	-
Disponível para uso		Não	Não	Não	Não	Parcial ²³	Sim

3.5 FERRAMENTAS

Apesar de existirem ambientes bastante promissores, tais como Teallach e CTTE + Petshop, para a concepção de interfaces centradas no usuário, estes não podem ser completamente aplicados, seja por não estarem diretamente relacionados a um processo de concepção centrado no usuário, seja por não estarem plenamente disponíveis para livre uso ou por ambas as razões.

²³ A ferramenta CTTE encontra-se disponível para *download*, sendo abordada em tópico posterior.

Pelas razões expostas, ampliou-se o horizonte de busca de forma a incluir outros suportes que pudessem apoiar etapas isoladas do processo de concepção. Tais ferramentas, desenvolvidas interna e externamente ao GIHM, são descritas a seguir.

3.5.1 Euterpe

Desenvolvida por M. van Welie, da Faculdade de Matemática e Ciência da Computação da Vrije Universiteit de Amsterdam, o Euterpe (Figura 29) é uma ferramenta gráfica que permite a modelagem e análise de tarefas.

- Foi desenvolvida em Prolog (funcionalidade) e C++ (interface gráfica) e baseia-se em ontologia advinda do formalismo GTA.

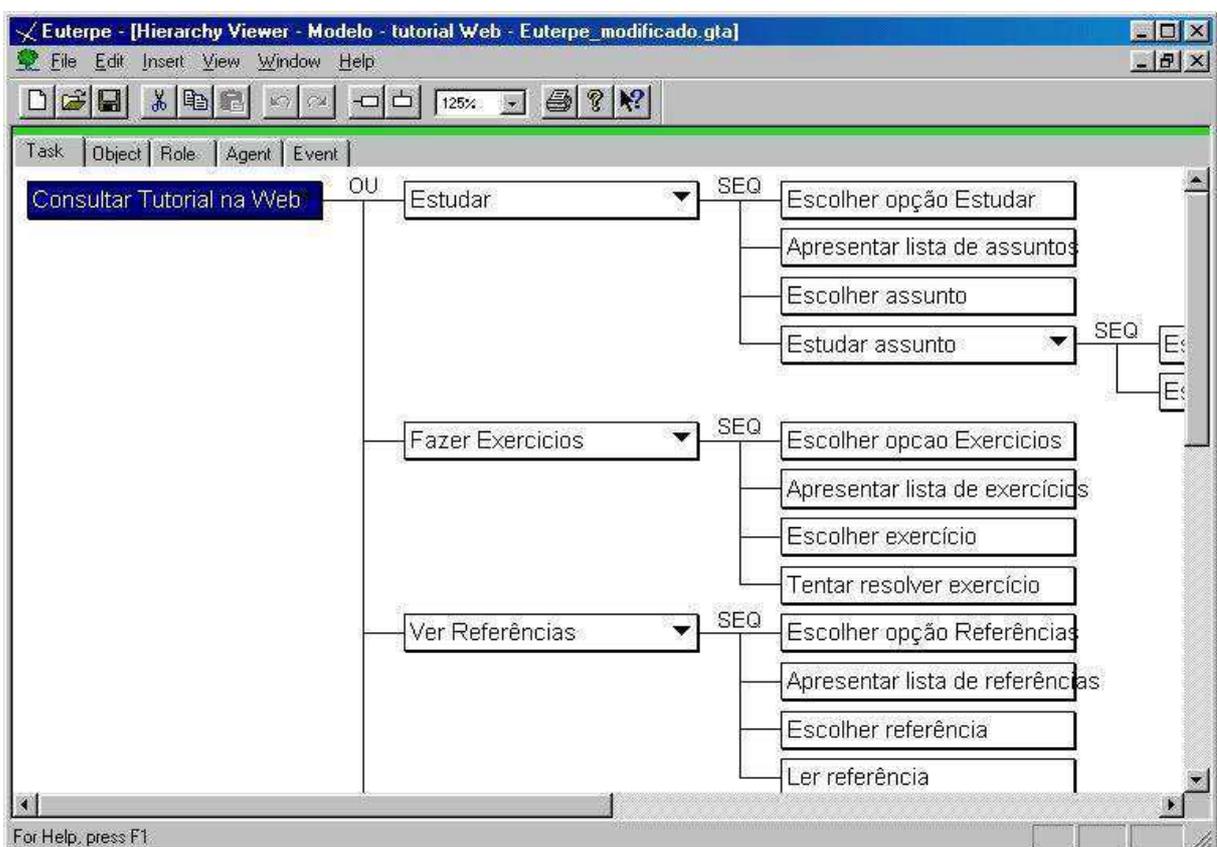


Figura 29 - Euterpe

3.5.1.1 Funcionalidades

Uma vez modelada a tarefa, seus objetos, papéis, agentes e eventos, pode-se:

- analisar o modelo (visualmente ou através de métricas). Em [Weli98], menciona-se o uso de cores objetivando facilitar a identificação visual de tarefas, entretanto esta facilidade ainda não está implementada. Em [Weli99] e na própria ferramenta

observa-se a existência de um *Analyser*. Entretanto não há informações de como utilizar a ferramenta e ao tentar testá-la, recebe-se uma mensagem de erro;

- criar um **Documento de Interação** que consiste em um conjunto de tabelas baseadas na notação UAN²⁴.

3.5.1.2 Considerações

A ferramenta fornece suporte tanto para a modelagem da tarefa quanto para a criação de um documento de interação. Entretanto oferece escassa documentação de ajuda. Além disto, tentando-se utilizá-la, segundo princípios de tentativa-e-erro, esbarra-se em problemas (*bugs*) para os quais não se encontra explicação ou solução.

Outro ponto desfavorável à ferramenta diz respeito a sua licença de uso que restringe a aplicação para atividades não comerciais, o que contraria a idéia de disponibilizar o ambiente em proposição para livre uso.

Quadro 16 - Resumo sobre Euterpe

Nome da Ferramenta	Euterpe
Autor ou Grupo de Pesquisa Responsável	M. van Welie Division Mathematics and Computer Science Vrije Universiteit Amsterdam
Status / Versão	1.00.91
Licença	Livre para uso não comercial
Idioma	Inglês
Plataforma	Windows
Linguagem em que foi desenvolvida	Prolog e C++
Notação associada	Modelagem da Tarefa: GTA Modelagem da Interação: UAN
Forma de entrada de dados	*.gta
Forma de saída de dados	*.gta Permite exportar para HTML para fins de documentação.
Site associado	http://www.cs.vu.nl/~martijn/gta
Material bibliográfico associado	[Weli98, Weli99, Veer00]

3.5.2 TAMOT

Ferramenta pertencente ao Commonwealth Scientific and Industrial Research Organisation (CSIRO) da Austrália, atualmente sendo desenvolvido pelo *Intelligent Interactive Technology Group* (IIT), como parte do projeto ISOLDE²⁵ (Figura 31).

²⁴ UAN: *User Action Notation*.

Consiste em um editor gráfico (Figura 30), desenvolvido em Java, baseado no formalismo Diane+ e tem por objetivo a modelagem de tarefas e domínios.

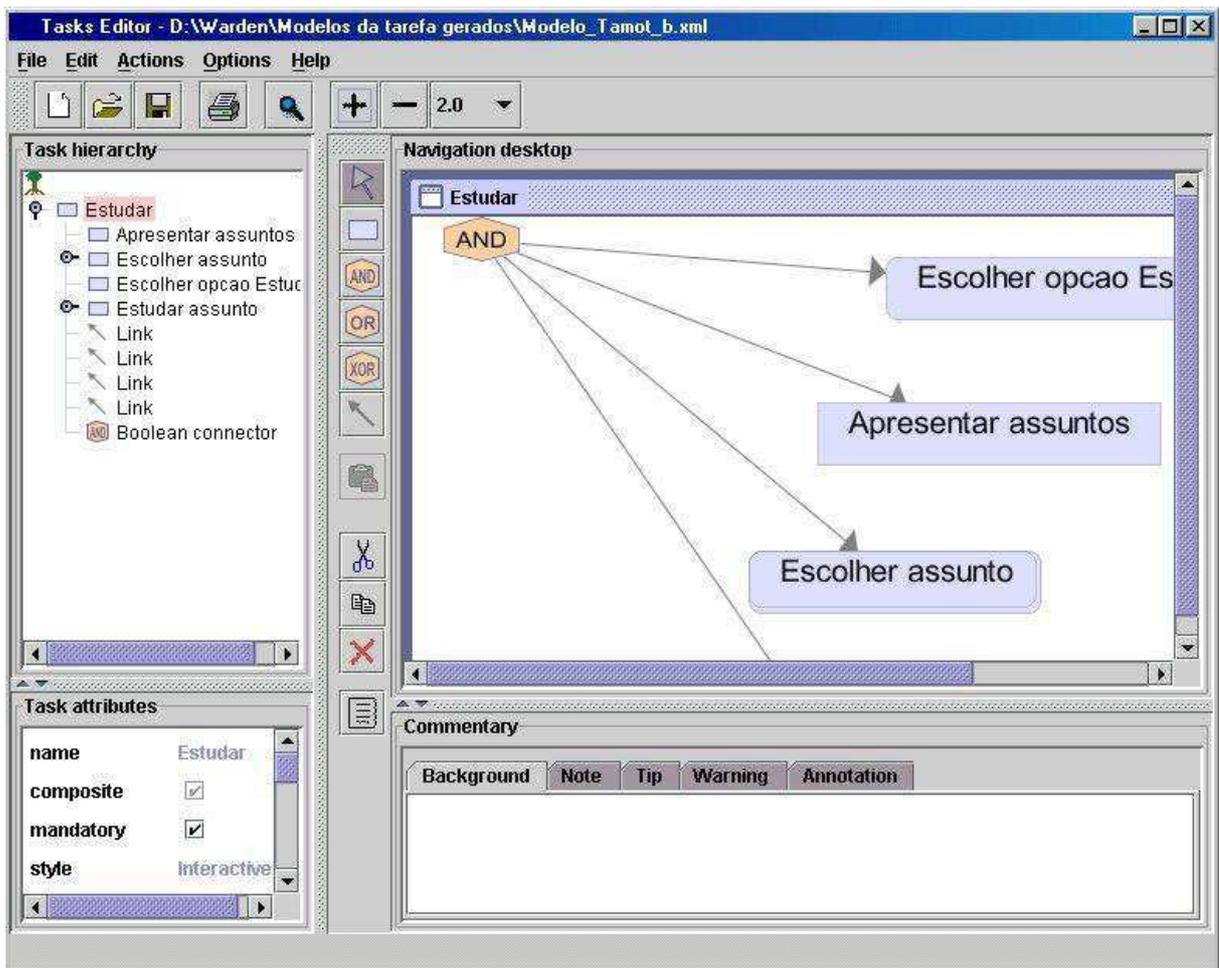


Figura 30 - TAMOT

O Diane+ foi criado por Jean-Claude Tarby e Marie-France Barthet e visa ser um método de desenvolvimento (*design method*) e projeto, a ser compreendido tanto por designers quanto por usuários. Para tanto, o formalismo embasa-se em abordagens orientadas a tarefas e orientadas a objetos.

Apesar do formalismo possuir um conjunto próprio de notações gráficas, no âmbito da ferramenta a modelagem das tarefas é realizada amparada por três elementos básicos:

- **Tasks** (tarefas), representando os objetivos;
- **Links** (conexões/ligações), permite conectar as tarefas;

²⁵ ISOLDE consiste em um projeto que objetiva a máxima automatização possível da geração de documentação [Pari02].

- **Boolean Connectors** (conectores lógicos), permite especificar relações entre tarefas.

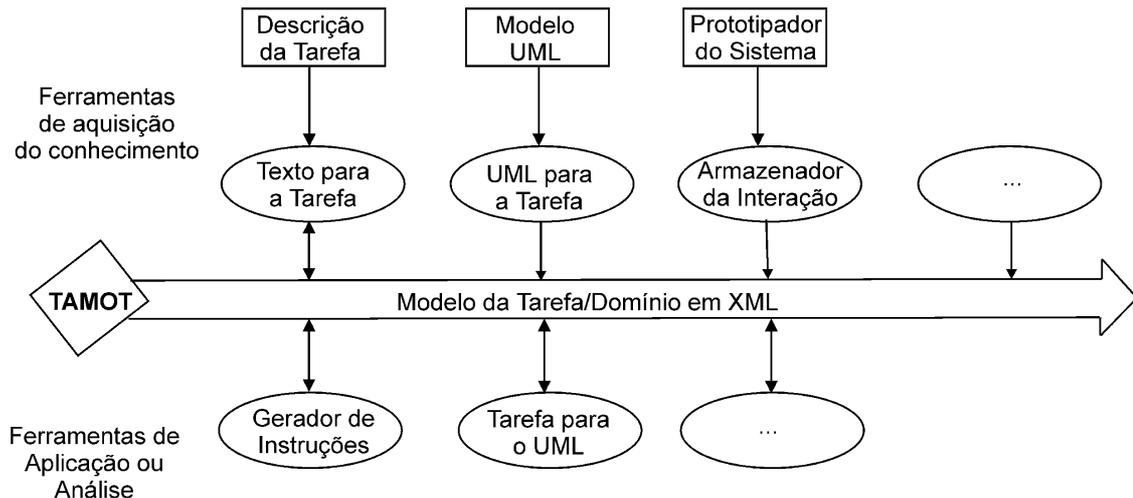


Figura 31 - Tamot e o ambiente ISOLDE

3.5.2.1 Funcionalidades

Uma vez modelada a tarefa, a ferramenta permite:

- gerar código HTML do modelo, visando documentação;
- armazenar o modelo criado em três formatos distintos: Tamot Java2, Tamot Imp e Tamot XML.

3.5.2.2 Considerações

A ferramenta possui interface bem projetada, porém, provavelmente em função da ferramenta ser parte de um projeto mais amplo - Projeto ISOLDE - que busca meios de automatizar a documentação, implica que o TAMOT não oferece funcionalidades além daquelas que são utilizadas para fins de documentação.

Outro ponto fraco do sistema é a quase inexistência de documentação sobre suas funcionalidades e possibilidades de uso, o que dificulta um aprofundamento no uso de suas funcionalidades.

Quadro 17 - Resumo sobre TAMOT

Nome da Ferramenta	TAMOT
Autor ou Grupo de Pesquisa Responsável	<i>Intelligent Interactive Technology Group Commonwealth Scientific and Industrial Research Organisation - Austrália</i>
Status / Versão	2.19
Licença	<i>Freeware</i>
Idioma	Inglês
Plataforma	Windows
Linguagem em que foi desenvolvida	Java
Notação associada	Diane+
Forma de entrada de dados	Tamot Java Tamot XML Tamot Imp
Forma de saída de dados	Tamot Java Tamot XML Tamot Imp HTML para fins de documentação
Site associado	http://www1.cmis.csiro.au/iit/Projects/Isolde/Tamot/index.asp
Material bibliográfico associado	[Pari02, Coli02, Pari03]

3.5.3 iTAOS

Ferramenta (Figura 32) desenvolvida em Java, no âmbito do GIHM por Cordeiro e Medeiros, e que objetiva apoiar a concepção e análise do Modelo da Tarefa, baseada no formalismo TAOS²⁶.

O ambiente é formado pela junção de dois módulos, desenvolvidos, respectivamente, por Cordeiro [Cord03] e Medeiros [Mede03]:

- TAME (*Task and Action Modeling Environment*), implementa as funcionalidades;
- TAOS-Graph, implementa a interface gráfica.

O TAOS consiste em um formalismo para aquisição e representação do conhecimento, baseado na modelagem do domínio (Anexo B) [Mede03, Cord03].

²⁶ TAOS: *Task and Action Oriented System*, formalismo desenvolvido por J. H. de Medeiros [Mede95].

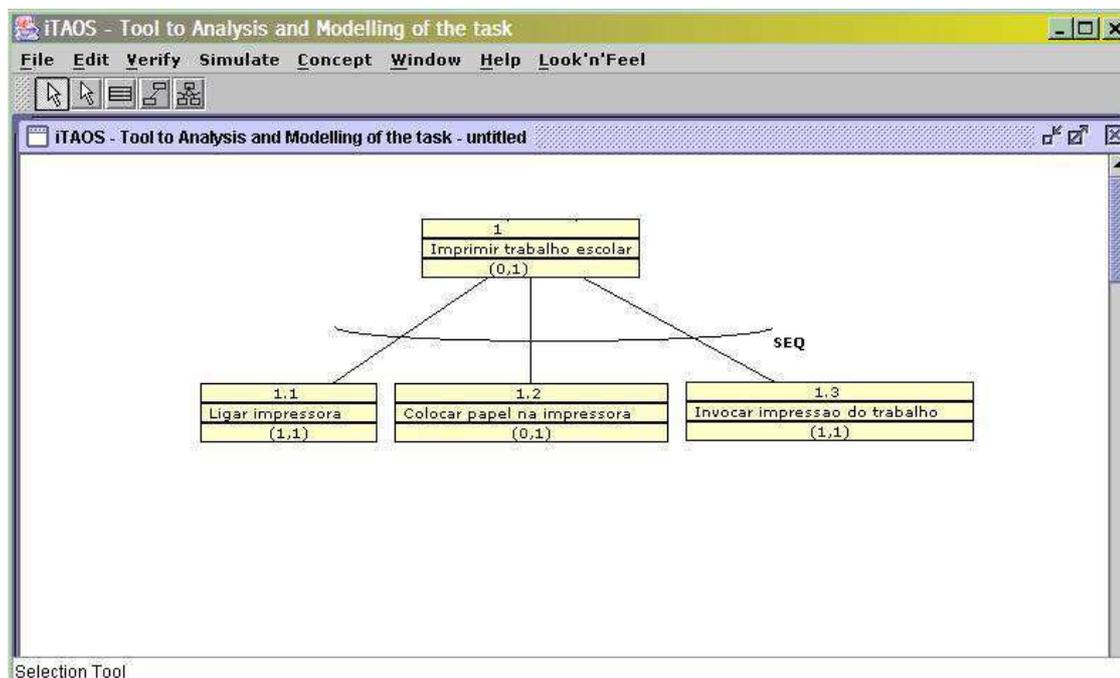


Figura 32 - iTAOS

3.5.3.1 Funcionalidades

Segundo Cordeiro [Cord03], a ferramenta possui as seguintes funcionalidades:

- permite a modelagem de tarefas concorrentes;
- possibilita a verificação de coerência e completude;
- possibilita a geração de documentação.

3.5.3.2 Considerações

Com uma interface bastante intuitiva, a ferramenta possibilita, de forma simples e prática, a concepção do modelo da tarefa. O fato de ter sido desenvolvida em Java, lhe confere a possibilidade de uso em multiplataformas. A ferramenta é de livre uso (*freeware*) e foi desenvolvida pela equipe do GIHM, o que permite acesso ao código fonte, tornando-a uma forte opção para uso no MCI. Outro ponto a ressaltar é o formato de persistência, que apesar de não estar explícito para o usuário, consiste de um formato proprietário e em XML, este último gerado automaticamente.

Entretanto, como seu desenvolvimento ainda é recente, algumas das funcionalidades apontadas em [Cord03] e [Mede03] ainda não foram implementadas, constando apenas como trabalhos futuros. Há também a necessidade por uma documentação de uso da ferramenta mais detalhada contendo exemplos.

Quadro 18 - Resumo sobre iTAOS

Nome da Ferramenta	iTAOS
Autor ou Grupo de Pesquisa Responsável	Francisco Petrônio Alencar de Medeiros Pedro Barbosa Cordeiro Grupo de Interfaces Homem-Máquina Universidade Federal de Campina Grande
Status / Versão	1.0.0
Licença	Freeware
Idioma	Inglês
Plataforma	Multiplataforma
Linguagem em que foi desenvolvida	Java
Notação associada	TAOS
Forma de entrada de dados	*.draw
Forma de saída de dados	*.draw *.xml (gravação automática sem interferência do usuário)
Site associado	http://www.dsc.ufcg.edu.br/~itaos
Material bibliográfico associado	[Mede95, Cord03, Mede03]

3.5.4 CTTE

Ferramenta desenvolvida pelo *Human-Computer Interaction Group* - CNUCE da Itália e baseada no formalismo CTT. O CTTE consiste em um editor gráfico que suporta a construção de modelos de tarefas cooperativas ou monousuárias (Figura 33).

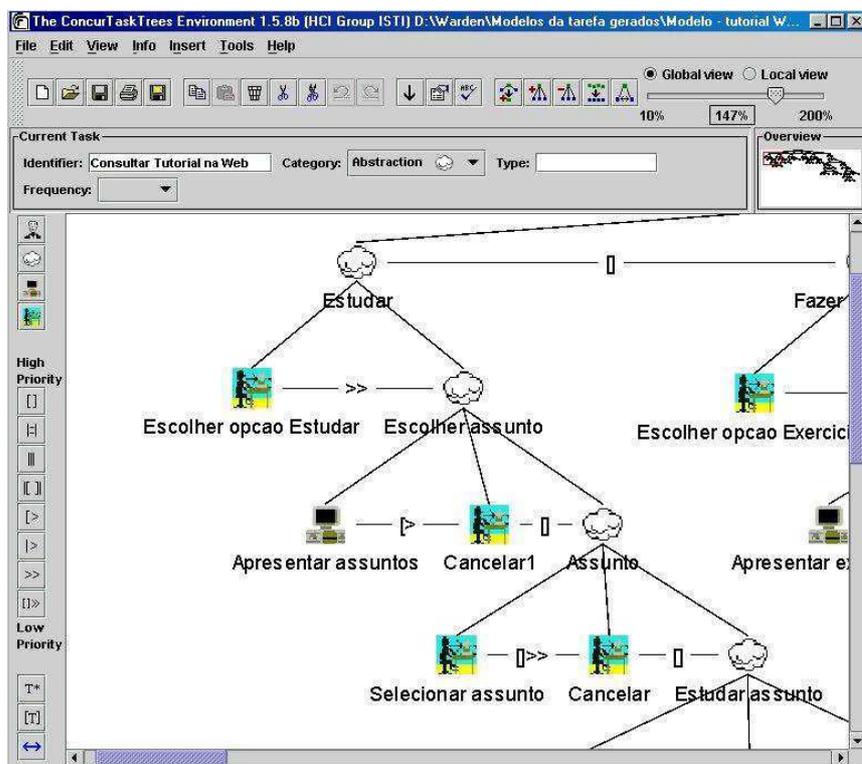


Figura 33 - CTTE

Segundo Paternò et al. [Pate97] a construção do modelo da tarefa é realizado em três fases:

- construção de uma estrutura em forma de árvore lógica e hierárquica das tarefas;
- identificação de relações temporais entre tarefas;
- identificação de objetos associados a cada tarefa e as ações existentes para a comunicação.

3.5.4.1 Funcionalidades

O projetista pode gerar o modelo de tarefas de duas formas:

- modelando a árvore através do editor gráfico; ou
- através de uma descrição informal das tarefas.

Uma vez criado o modelo, é possível:

- verificar a completude sintática e semântica do modelo, de acordo com os preceitos (ex.: se há tarefas sem operadores temporais, se há subtarefas de categorias incompatíveis com a tarefa pai) da ferramenta;
- comparar modelos, havendo uma visão comparativa quanto ao número de tarefas, “quais” e “quantos” operadores foram utilizados, entre outras informações;
- converter as especificações em CTT para LOTOS, possibilitando o uso de ferramentas de análise existentes para o formalismo LOTOS;
- simular o modelo criado, bem como utilizar cenários.

3.5.4.2 Considerações

Assim como a ferramenta iTAOS, também o CTTE possui uma interface bastante intuitiva, apesar de não ser tão prática para gerar o modelo da tarefa. Alguns de seus pontos positivos são: (I) ter sido desenvolvida em Java (multiplataforma), (II) ser de livre uso (*freeware*) e (III) oferecer funcionalidades para simulação e a possibilidade de aplicar análises mais aprofundadas (ex.: identificação de marcações mortas)²⁷.

²⁷ Havendo, neste caso, a necessidade de ferramentas extras que trabalhem sob o formalismo LOTOS.

Entretanto, a ferramenta não oferece documentação substancial acerca de seu uso e funcionamento, além de apresentar algumas dificuldades em termos da usabilidade de sua interface gráfica.

Quadro 19 - Resumo sobre CTTE

Nome da Ferramenta	CTTE
Autor ou Grupo de Pesquisa Responsável	<i>Human-Computer Interaction Group - CNUCE</i>
Status / Versão	1.5.8b
Licença	<i>Freeware</i>
Idioma	Inglês
Plataforma	Multiplataforma
Linguagem em que foi desenvolvida	Java
Notação associada	CTT (UAN e LOTOS constituem formalismos embaixadores)
Forma de entrada de dados	*.ctt / *.cctt *.xml
Forma de saída de dados	*.ctt / *.cctt *.xml *.jpg
Site associado	http://giove.cnuce.cnr.it/ctte.html
Material bibliográfico associado	[Pate97, Pate01, Mori02, Pate02]

3.5.5 DENIM

Ferramenta (Figura 34) desenvolvida pelo *Group for User Interface Research* da Universidade da Califórnia, em Berkeley, objetivando fornecer suporte a projetistas de *web sites* nos estágios iniciais do projeto.

O método utilizado pela ferramenta segue o princípio de *storyboards*, segundo o qual o projetista desenha as telas e a seqüência de execução destas, desta forma, isenta o projetista da obrigação de possuir conhecimentos acerca de programação.

O DENIM faz parte de uma pesquisa em interfaces informais com o usuário (*informal user interfaces*), sendo uma ferramenta similar ao SILK²⁸ [Land96, Land95].

Segundo Lin et al. [Lin02a], a pesquisa em interfaces informais busca o suporte computacional para as entradas de dados em linguagem natural, tais como escrita e fala, objetivando minimizar os problemas de reconhecimento e transformação desta entrada.

²⁸ SILK: Sketching Interfaces Like Krazy e suporta os estágios iniciais de projeto de interfaces com o usuário.

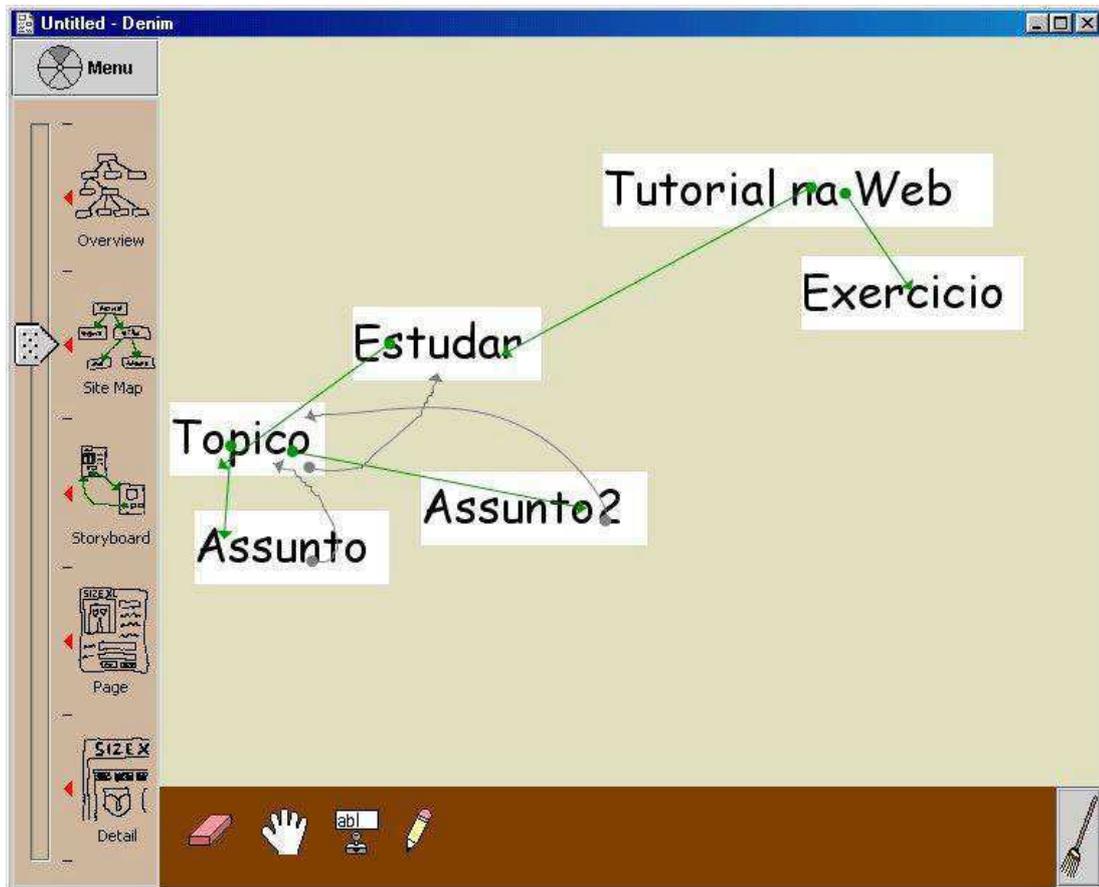


Figura 34 - DENIM

Seguindo este princípio, o grupo de pesquisa desenvolveu a ferramenta de forma que esta fosse utilizada em conjunto com uma caneta e mesa digitalizadora, apesar da ferramenta permitir o uso de mouse, ao invés da caneta e mesa digitalizadora. Porém, há conseqüente aumento de dificuldade no uso da ferramenta.

A ferramenta consiste em um editor gráfico constituído por:

- **zoom slider**, no qual o designer controla a amplitude de visão, indo do nível mais alto (*overview*) até o nível de dados internos a cada página (*detail*);
- **canvas**, área de trabalho do projetista;
- **toolbox**, consiste de um conjunto de ferramentas disponíveis ao designer.

3.5.5.1 Funcionalidades

Uma vez criado o modelo, a ferramenta permite:

- executar o *web site* criado, sem haver a necessidade de uso de *browser*;

- exportar os projetos criados para HTML. Neste caso, a ferramenta gera uma imagem mapeada de cada página esboçada e um arquivo HTML correspondente.

3.5.5.2 Considerações

Ainda que o usuário disponha do equipamento considerado mais adequado para a atividade de concepção, ie. mesa digitalizadora e caneta, será necessário ter habilidade no uso destes instrumentos caso deseje explorar plenamente as potencialidades da ferramenta e dos caracteres pré-estabelecidos. Além do conjunto de caracteres especiais que permitem o acesso direto aos comandos de controle da ferramenta, estão disponíveis menus e atalhos (*box* acessíveis com o botão direito do mouse), que entretanto, não se mostraram como opções práticas de uso.

O uso desta ferramenta como suporte ao método MCI oferece a possibilidade de gerar *storyboards* do protótipo em conjunto com o usuário promovendo e facilitando sua interação com a equipe de projeto. Entretanto, os modelos gerados pela ferramenta, apesar de serem exportados para o formato HTML, não consistem de linhas de código puras, mas de imagens mapeadas e conectadas por *links*.

Quadro 20 - Resumo sobre DENIM

Nome da Ferramenta	DENIM
Autor ou Grupo de Pesquisa Responsável	<i>Group for User Interface Research</i> Universidade da Califórnia em Berkeley
Status / Versão	1.1
Licença	<i>Freeware</i>
Idioma	Inglês
Plataforma	Multiplataforma
Linguagem em que foi desenvolvida	-
Notação associada	<i>Storyboard</i>
Forma de entrada de dados	*.dnm
Forma de saída de dados	*.dnm *.html
Site associado	http://guir.cs.berkeley.edu/projects/denim/
Material bibliográfico associado	[Land95, Land96, Lin01, Lin02a, Grou03]

3.5.6 Gartner Web Evaluation Tool

O Gartner Web Evaluation Tool (Figura 35) é uma ferramenta desenvolvida pelo Gartner Group com base em tabelas construídas com o aplicativo Microsoft Excel, com o propósito de apoiar a avaliação de *sites* web a partir de três critérios:

- *Design do Site;*

- Funcionalidade do *Site*;
- Valor atribuído pelo cliente (*Customer Value*).

Gartner Web Evaluation Tool		SITE 1	SITE 2
CRITERIA		Type site 1 Firm Name Here	Type site 2 Firm Name Here
1	Site Design		
1.1	<i>Navigation</i>		
1.1a	Site Map --structure clearly communicated		
1.1b	Ease of / straightforward navigation		
1.1c	Segment-specific navigation		
1.1d	Error statements/broken links encountered?		
1.1e	Site prevents errors / helps users recover		
1.1f	Did the site launch additional browsers?		
1.1g	Did the site use pop up windows/interstitials?		
	<i>Sub-Total</i>	0	0
	<i>Weighted Sub-Total</i>	0	0

Figura 35 - Gartner Web Evaluation Tool [Gart01]

Através do cálculo automático de médias dos valores atribuídos às questões relativas a cada um dos critérios, a ferramenta gera gráficos de resultados.

3.5.6.1 Funcionalidades

Permite a entrada de dados referentes a cada um dos critérios citados e de seus subcritérios para então gerar, de forma automática, gráficos e resultados para análise posterior.

3.5.6.2 Considerações

Apesar de ser uma ferramenta vinculada a um software comercial (Excel) e de ter seu uso restrito segundo as normas do Gartner Group, a maior contribuição desta ferramenta consiste na idéia de usar uma planilha de cálculos informatizada como forma de automatizar a coleta de dados de avaliação de um produto e totalizar os resultados. Inspirados nesta idéia é possível desenvolver um produto com características similares para facilitar a coleta e análise de dados referentes ao Perfil do Usuário e ao Contexto de uso.

Quadro 21 - Resumo sobre Gartner Web Evaluation Tool

Nome da Ferramenta	Gartner Web Evaluation Tool
Autor ou Grupo de Pesquisa Responsável	Gartner Group
Status / Versão	-
Licença	-
Idioma	Inglês
Plataforma	Windows
Linguagem em que foi desenvolvida	-
Notação associada	-
Forma de entrada de dados	-
Forma de saída de dados	-
Site associado	-
Material bibliográfico associado	[Gart01]

3.5.7 Design/CPN

Ferramenta desenvolvida pela Meta Software Corporation (EUA), em parceria com pesquisadores do grupo de Redes de Petri da Aarhus University (Dinamarca) [Desi02]. O Design/CPN (Figura 36) esteve no catálogo de produtos da Meta Software de 1989 a 1995, transformando-se em ferramenta *freeware* somente a partir de 1996, quando então o grupo de pesquisadores da Aarhus obtiveram plenos direitos sobre a ferramenta.

A ferramenta é voltada para a aplicação do formalismo de Redes de Petri Coloridas (*Coloured Petri Nets* - CPN), tendo origem em 1962, na Tese de Doutorado de Carl Adam Petri. CPN consiste em modelar graficamente, porém fortemente embasado em conceitos matemáticos, diversos tipos de sistemas [Jens92].

Esta ferramenta vem sendo usada na fase de construção de modelos de navegação no método MCI, quando é adotado o formalismo CPN.

3.5.7.1 Funcionalidades

Como primeira funcionalidade, destaca-se o **editor** que permite a construção dos modelos. Para completo aproveitamento das demais funcionalidades, além de representar graficamente o modelo gerado, a ferramenta também aceita linhas de código baseadas na linguagem ML.

Uma vez gerados os modelos, pode-se:

- simulá-los passo a passo ou automaticamente;
- gerar gráficos de seqüência de mensagens (MSC);

- analisar os modelos, tendo como base os gráficos MSC gerados ou utilizar-se das ferramentas de análise disponíveis na ferramenta.

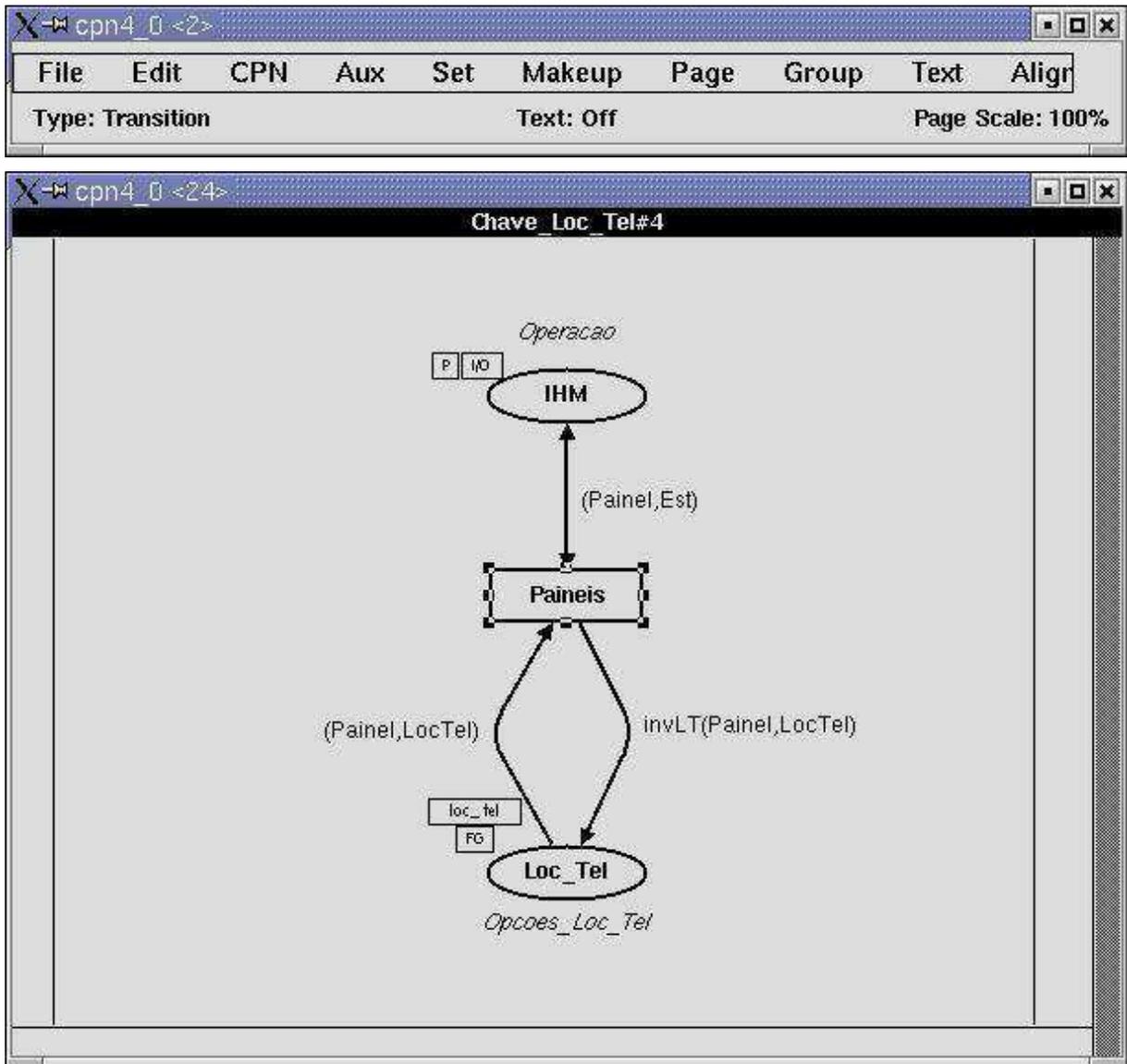


Figura 36 - Design/CPN

3.5.7.2 Considerações

O Design/CPN é uma ferramenta amplamente utilizada pelo Grupo de Interfaces Homem-Máquina da UFCG, entretanto, o uso desta ferramenta exige um mínimo de conhecimento acerca do formalismo CPN e das particularidades da linguagem ML.

Outro problema diz respeito à plataforma de uso da ferramenta que é restrita ao ambiente Linux. Porém, o Design/CPN permite armazenar e recuperar os modelos gerados em XML, o que se apresenta como uma possibilidade de integração com outras ferramentas no ambiente de suporte ao MCI.

Existem iniciativas, dentro e fora do grupo CPN da Aarhus, para implementação de uma versão desta ferramenta para o ambiente Windows, entretanto esta ainda não está disponível para livre uso.

Quadro 22 - Resumo sobre Design/CPN

Nome da Ferramenta	Design/CPN
Autor ou Grupo de Pesquisa Responsável	CPN Group Universidade de Aarhus
Status / Versão	3.0
Licença	Freeware
Idioma	Inglês
Plataforma	Linux
Linguagem em que foi desenvolvida	-
Notação associada	Coloured Petri Nets (CPN)
Forma de entrada de dados	*.xml
Forma de saída de dados	*.xml
Site associado	http://www.daimi.au.dk/designCPN/
Material bibliográfico associado	[Desi02, Scai01]

3.5.8 OpUS (Opinião do Usuário de Software)

Instrumento desenvolvido no GIHM (Figura 37) por Queiroz [Quei01] e aplicado na sondagem da opinião do usuário acerca de um produto, após a realização de um teste de usabilidade.

Este questionário integrado tem o propósito de coletar informações acerca de como você se sente a respeito do uso do PRODUTO sob condições de teste e de sua documentação *online* e *offline*. Por favor analise cada aspecto considerado, selecionando a opção que mais adequa à sua condição de usuário.

Aspecto		O Produto - Uso e Navegação				
		ESCALA				
		1	2	3	4	5
		Muito fácil	Fácil	Nem fácil nem difícil	Difícil	Muito difícil
1	Uso do produto na realização de tarefas de interesse	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Comunicação com o produto (terminologia, linguagem, retorno da informação e das ações em geral)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Localização dos itens de menu associados às tarefas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Acesso aos itens de menu associados às tarefas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura 37 - OpUS

Inicialmente, o OpUS foi concebido sobre o Microsoft Excel. Entretanto, existe uma versão impressa em Microsoft Word e uma versão para o formato HTML. A versão web da ferramenta foi motivada pela necessidade de flexibilizar o acesso aos respondentes.

O questionário OpUS é dividido em três seções, a citar:

- **o produto - uso e navegação**, consta de 20 itens voltados para aspectos da interface, referentes ao seu uso e navegação;
- **o produto - documentação *online* e *offline***, segundo a qual, através de 10 itens, objetiva-se sondar acerca da ajuda oferecida pelo produto;
- **você e o produto**, composta por 20 itens, visando sondar acerca das impressões do respondente quanto sua aceitação do produto.

3.5.8.1 Funcionalidades

Como exposto anteriormente, a ferramenta possui um conjunto de questionamentos visando a sondagem da opinião pós-teste de determinado produto.

Cabe destacar que todos os itens constituintes do suporte estão associados a escalas numérico-semântica com cinco pontos, sendo que os 30 primeiros possuem escala constituída pelos adjetivos:

- muito fácil (1);
- fácil (2);
- nem fácil nem difícil (3);
- difícil (4);
- muito difícil (5).

Já os 20 itens finais, devido a sua natureza e estilo, possuem escala constituída por:

- concordo totalmente (1);
- concordo parcialmente (2);
- nem concordo nem discordo(3);
- discordo parcialmente (4);
- discordo totalmente (5).

3.5.8.2 Considerações

Assim como no caso da ferramenta iTAOS, também o OpUS é de autoria do GIHM, o que facilita sua adaptação a diferentes contextos de aplicação.

Um dos pontos desfavoráveis incide, justamente, na questão da persistência dos dados, uma vez que nenhuma das versões disponíveis possui preocupação em realizar persistência em formato XML.

Quadro 23 - Resumo sobre OpUS

Nome da Ferramenta	OpUS
Autor ou Grupo de Pesquisa Responsável	José Eustáquio Rangel de Queiroz GIHM Universidade Federal de Campina Grande
Status / Versão	-
Licença	<i>Freeware</i>
Idioma	Português
Plataforma	Microsoft Windows (versões para Excel e Word) Multiplataforma (versão web)
Linguagem em que foi desenvolvida	-
Notação associada	Questionários
Forma de entrada de dados	-
Forma de saída de dados	-
Site associado	http://www.dee.ufcg.edu.br/~copele/gihm/opus.html
Material bibliográfico associado	[Quei01]

3.5.9 DePerUSI (Delineamento do Perfil do Usuário de Sistemas Interativos)

Assim como o OpUS, também esta ferramenta (Figura 38) foi desenvolvida no âmbito do GIHM, por Queiroz [Quei01], neste caso, visando ser um instrumento de pré-teste para a sondagem do perfil do usuário.

Constituído por 24 itens visando a sondagem de 19 características, apontadas no Quadro 24, o OpUS possibilita realizar mapeamento e categorização dos usuários em função de características **físicas, conhecimento e experiências** e sobre a **tarefa e seu trabalho**.

Outro ponto de semelhança com a ferramenta OpUS é quanto ao formato em que foi concebido. Inicialmente sobre o Microsoft Excel, porém devido a necessidade de tê-lo impresso, converteu-se para Microsoft Word, e posteriormente, para o formato HTML.

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

ASPECTO	OPÇÕES
1. Você é:	Nível acadêmico <input type="radio"/> Pós-graduado <input type="radio"/> Graduado <input type="radio"/> Estudante de Pós-graduação <input type="radio"/> Estudante de Graduação
2. Você é do sexo:	Sexo <input type="radio"/> Masculino <input type="radio"/> Feminino
3. Você é:	Destreza manual <input type="radio"/> Destro <input type="radio"/> Canhoto <input type="radio"/> Ambidestro
4. Você usa óculos ou lentes de contato?	Uso de corretivos visuais <input type="radio"/> Sim <input type="radio"/> Não
5. Você pertence à faixa etária de:	Faixa etária

Figura 38 - DePerUSI

Quadro 24 - Características relativas ao usuário [Quei01]

Natureza da característica	Especificação
Física	Sexo
	Destreza Manual
	Uso de Corretivos Visuais
	Faixa Etária
Relativa ao Conhecimento e à Experiência	Nível Acadêmico
	Experiência Computacional Prévia
	Tempo de Uso de Sistemas Computacionais
	Frequência de Uso de Sistemas Computacionais
	Experiência Prévia com o Produto
	Tempo de Uso do Produto
	Frequência de Uso do Produto
	Experiência Prévia com Produtos Similares
	Discriminação de Produtos Similares Utilizados
Tempo de Uso de Produtos Similares	
Relativa à Tarefa e ao Trabalho	Natureza das Atividades (utilizando o produto)
	Uso Contextual do Produto
	Plataforma Computacional
	Versão Utilizada do Produto
	Forma de Ajuda mais Utilizada

3.5.9.1 Funcionalidades

Maior funcionalidade do produto vem a ser a possibilidade de agilizar o levantamento e compilação dos dados sobre grupos de usuários.

3.5.9.2 Considerações

Novamente, assim como iTAOS e OpUS, também com o DePerUSI, por ser de autoria do GIHM, existe maior liberdade de acesso aos fontes.

Entretanto a ferramenta não realiza persistência em formato XML.

Quadro 25 - Resumo sobre DePerUSI

Nome da Ferramenta	DePerUSI
Autor ou Grupo de Pesquisa Responsável	José Eustáquio Rangel de Queiroz GIHM Universidade Federal de Campina Grande
Status / Versão	-
Licença	<i>Freeware</i>
Idioma	Português
Plataforma	Microsoft Windows (versões para Excel e Word) Multiplataforma (versão web)
Linguagem em que foi desenvolvida	-
Notação associada	Questionários
Forma de entrada de dados	-
Forma de saída de dados	-
Site associado	http://www.dee.ufcg.edu.br/~copele/gihm/deperusi.html
Material bibliográfico associado	[Quei01]

3.5.10 TERESA (Transformation Environment for interactive Systems representations)

Ferramenta (Figura 39) desenvolvida pelo grupo de HCI do ISTI-C.N.R como parte integrante do projeto CAMELEON visando fornecer apoio computacional à etapa posterior à modelagem da tarefa, de forma a gerar a interface do usuário.

Segundo apontado em [Mori03], a ferramenta provê “soluções gerais que podem ser utilizadas para casos específicos”. Ainda segundo os autores, a ferramenta suporta transformações *top-down*, podendo-se obter descrições concretas a partir de representações abstratas.

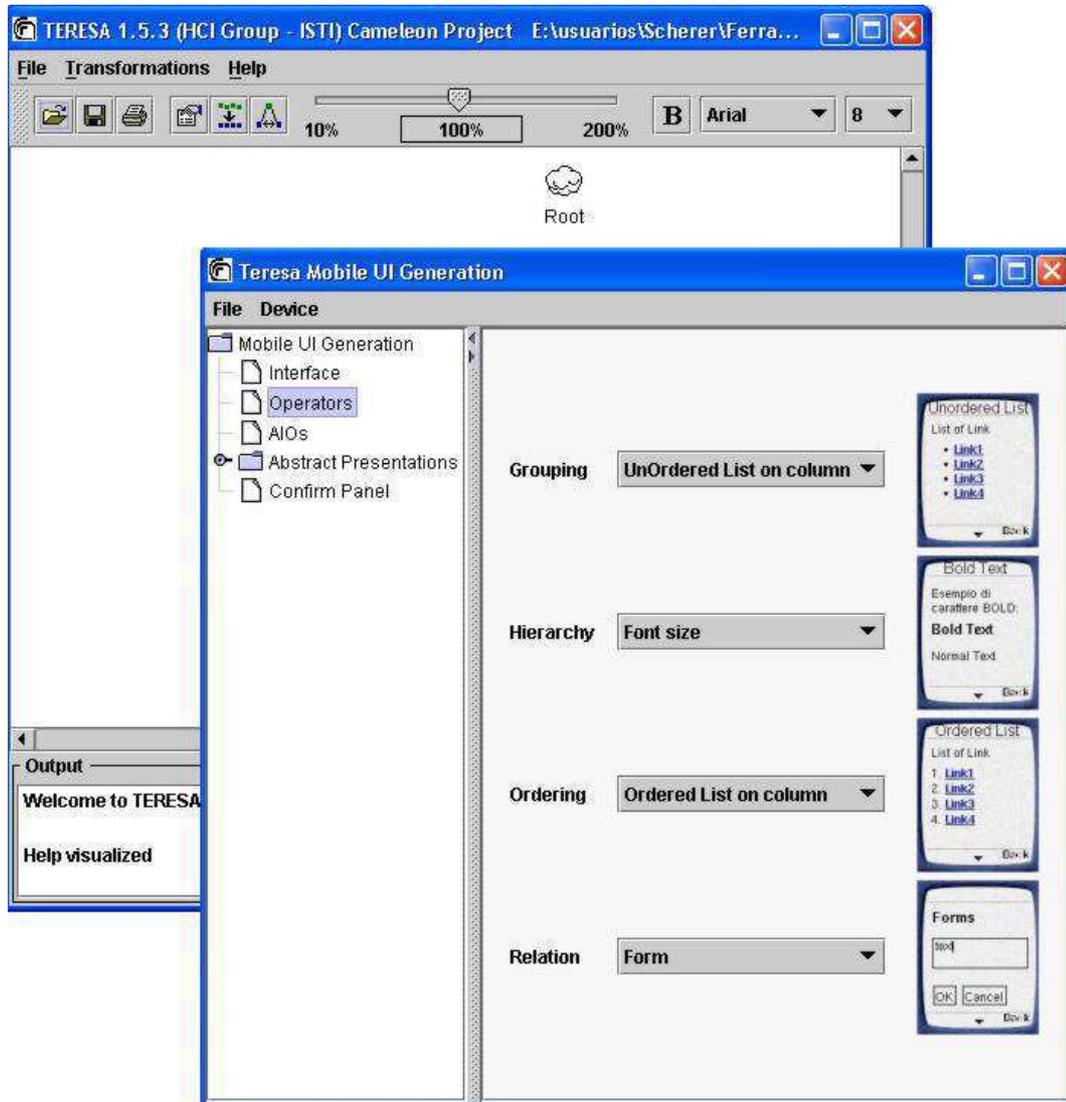


Figura 39 - TERESA

3.5.10.1 Funcionalidades

As funcionalidades apontadas pelos desenvolvedores da ferramenta são:

- **Gerador do Conjunto de Tarefas Habilitadas**, segundo o qual há um conjunto de tarefas disponíveis (ETS²⁹) em um mesmo período de tempo, em função de constantes indicadas no modelo da tarefa;
- **Gerador da interface abstrata a partir do modelo da tarefa**, após obtenção da ETS, gera-se a interface abstrata, cuja especificação é dividida em **estrutura estática** (“apresentação”) e o **comportamento dinâmico** (“diálogo”). Esta especificação é gravada para análises e transformações adicionais;

²⁹ ETS: Enable Task Sets.

- **Transformar a interface abstrata em interface concreta para sistemas *Desktops***, a partir da especificação abstrata da interface e aliado a um conjunto de parâmetros configuráveis, pode-se gerar a interface concreta para sistemas *Desktops*;
- **Transformar a interface abstrata em interface concreta para dispositivos móveis**, a partir da especificação abstrata da interface e aliado a um conjunto de parâmetros configuráveis, pode-se gerar a interface concreta para dispositivos móveis;
- **Geração automática da interface com o usuário**, possibilita ao projetista a concepção da interface de forma automática, através do modelo da tarefa e utilizando-se um conjunto de configurações padrão.

3.5.10.2 Considerações

Ferramenta desenvolvida pelo mesmo grupo que desenvolveu o CTTE e criada para ser facilmente integrada com este, TERESA promete ser uma ferramenta de grande utilidade, justamente por atacar uma etapa de difícil manejo e, talvez por isto, ainda carente de ferramentas: **modelagem e análise da interação**.

Entretanto, ao que parece, o sistema ainda está longe de cumprir tudo o que promete. TERESA é uma ferramenta que poderia ser classificada de “imatura”. As funcionalidades que são apontadas ainda não são executadas plenamente, deixando muito a desejar em diversos pontos do processo.

Quadro 26 - Resumo sobre TERESA

Nome da Ferramenta	TERESA
Autor ou Grupo de Pesquisa Responsável	<i>Human-Computer Interaction Group - CNUCE</i>
Status / Versão	2.0
Licença	<i>Freeware</i>
Idioma	Inglês
Plataforma	Multiplataforma
Linguagem em que foi desenvolvida	Java
Notação associada	-
Forma de entrada de dados	*.xml, *.ctt
Forma de saída de dados	*.html, *.wml, *.xml
Site associado	http://giove.cnuce.cnr.it/teresa.html
Material bibliográfico associado	[Mori03]

3.5.11 CADP (Caesar/Aldebaran Development Package)

Conjunto de ferramentas desenvolvido pelo grupo VASY do INRIA *Rhone-Alpes and the Verimag laboratory* objetivando: compilar eficientemente, simular, verificar formalmente e testar descrições escritas na linguagem LOTOS [Gara02].

Segundo informações do grupo responsável, o CADP aceita três formalismos diferentes como entrada, a citar:

- descrições em alto nível (LOTOS) que são compiladas para C por duas ferramentas (CAESAR e CAESAR.ADT) de forma a permitir a simulação, verificação e testes dos modelos;
- descrições de mais baixo nível como o LTS (*Labelled Transition Systems*);
- formatos de autômatos de comunicação de redes, tais como o EXP (autômatos combinados com a composição paralela de LOTOS e operadores ocultos) e o FC2 (autômatos combinados com um produto de sincronização).

3.5.11.1 Funcionalidades

O pacote oferece um conjunto de componentes interconectados:

- **CAESAR**, compilador do pacote que permite transformar a parte comportamental das especificações em LOTOS em código C ou LTS;
- **CAESAR.ADT**, compilador que permite transformar a parte de dados das especificações em LOTOS em bibliotecas de tipos e funções C;
- **CAESAR.INDENT**, consiste em um formatador de especificações LOTOS, verificando sua sintaxe e indentando o código para facilitar sua leitura;
- **OPEN/CAESAR**, *framework* com diversas ferramentas;
- **OCIS**, **XSIMULADOR** e **SIMULATOR**, simuladores interativos;
- **EXECUTOR**, ferramenta de execução randômica;
- **GENERATOR** e **REDUCTOR**, duas ferramentas de análise de alcançabilidade;
- **PREDICTOR**, ferramenta que verifica a possibilidade de se analisar a alcançabilidade;
- **EXHIBITOR**, ferramenta de *sequence-searching*;

- **PROJECTOR**, ferramenta que em um primeiro momento transforma uma semi-composição em LTS e em seguida gera o código LTS referente ao comportamento descrito na semi-composição;
- **EVALUATOR**, ferramenta para verificação de modelo;
- **TVG**, ferramenta para a geração de testes de conformidade baseados em uma tecnologia de verificação desenvolvida pelo VASY.
- **ALDEBARAN**, ferramenta que permite verificar sistemas de comunicação representados através de LTS;
- **BCG (*Binary-Coded Graphs*)**, é tanto um formato para a representação explícita de LTS quanto uma coleção de bibliotecas e programas de conduta deste formato;
- **XTL (*eXecutable Temporal Language*)**, é uma linguagem funcional de programação projetada como uma implementação compacta e simplificada de vários operadores lógicos temporais;
- **EUCALYPTUS**, interface gráfica escrita em Tcl/Tk que integra o CADP e disponibiliza um conjunto de ferramentas de forma a dispor uma interface *user-friendly* e unificada;
- **SVL (*Script Verification Language*)**, linguagem de scripts que objetiva simplificar e automatizar a verificação de programas LOTOS;

3.5.12 Considerações

O CADP possui um conjunto de ferramentas que lhe fornece um amplo rol de recursos para a análise e avaliação de modelos especificados em LOTOS.

Entretanto há dificuldades para a aquisição do pacote de ferramentas, que apesar de estar publicamente disponível, exige o preenchimento de cadastro antes de obter a disponibilização da ferramenta.

Quadro 27 - Resumo sobre CADP

Nome da Ferramenta	CADP
Autor ou Grupo de Pesquisa Responsável	VASY
Status / Versão	1.184
Licença	<i>Freeware</i>
Idioma	Inglês
Plataforma	UNIX / Linux / Solaris / Windows

Linguagem em que foi desenvolvida	
Notação associada	LOTOS LTS EXP FC2
Forma de entrada de dados	
Forma de saída de dados	
Site associado	http://www.inrialpes.fr/vasy/cadp
Material bibliográfico associado	[Gara02]

3.6 RELAÇÃO ENTRE ETAPAS DO MCI E FERRAMENTAS

O Quadro 28 apresenta a relação entre as etapas do MCI e as ferramentas discutidas neste tópico. Importante observar que existem etapas, como a **Análise e Modelagem da Tarefa**, que possuem um grupo mais extenso de ferramentas associadas enquanto que outras, como a **Análise e Modelagem do Contexto**, não possuem nenhuma ferramenta diretamente relacionada. Entretanto, como foi apontado nas considerações individuais das ferramentas, existem algumas que, mesmo não contemplando determinada etapa, podem servir de inspiração para o desenvolvimento do suporte adequado.

Dentre os pontos negativos levantados, merece destaque a dificuldade no aprendizado das ferramentas, principalmente devido a escassez de documentação e a qualidade destes documentos que deixa de apresentar exemplos que facilitem o aprendizado.

3.6.1 Considerações

A análise do conjunto de suportes computacionais levantados, tanto ambientes quanto de ferramentas, permitiu verificar que:

- boa parte dos aparatos computacionais de cunho acadêmico encontra-se indisponível para aquisição e uso, seja por ainda estar em desenvolvimento, seja por ser de uso exclusivo dos grupos que criaram os aparatos ou ainda por ter sido descontinuado;
- nenhum dos suportes apurados possui documentação adequada. Todos exigem que o projetista busque dados complementares em artigos ou mesmo em teses e dissertações, muitas vezes inacessíveis;
- alguns sistemas encontram-se disponíveis para aquisição e uso (vide Anexo A - Outros Suportes), entretanto, em virtude da escassez de documentação adequada, não houve meios de verificar seus funcionamentos ou até mesmo de instalá-los.

Quadro 28 - Relação entre Etapas do MCI e as Ferramentas

MCI		Euterpe	TAMOT	iTAOS	CTTE	DENIM	Gartner Web Evaluation Tool	Design/CPN	OpUS	DePerUSI	TERESA	CADP
Análise de Requisitos	Análise e Modelagem do Perfil do Usuário	-	-	-	-	-	-	-	-	X	-	-
	Análise e Modelagem do Contexto	-	-	-	-	-	-	-	-	-	-	-
	Análise e Modelagem da Tarefa	X	X	X	X	-	-	-	-	-	-	-
	Avaliação do Modelo da Tarefa	X	X	X	X	-	-	-	-	-	-	-
Concepção do Modelo de Interação	Análise e Modelagem da Interação	X	-	-	-	-	-	-	-	-	X	-
	Avaliação do Modelo de Interação	-	-	-	-	-	-	-	-	-	X	-
	Representação do Cenário	-	-	-	X	-	-	-	-	-	-	-
	Construção do Modelo de Navegação	-	-	-	X	X	-	X	-	-	-	-
	Avaliação do Modelo de Navegação	-	-	-	X	X	-	X	-	-	-	X
Construção e avaliação do protótipo	Geração do Protótipo	-	-	-	-	X	-	-	-	-	X	-
	Avaliação do Protótipo	-	-	-	-	-	X	-	X	-	-	-
Disponível		SIM	SIM	SIM	SIM	SIM	SIM	SIM	SIM	SIM	SIM	SIM*

* Apesar de publicamente disponível e de preencher o cadastro exigido, não foi possível sua obtenção.

Com referência aos recursos que contemplam as etapas do método, pode-se observar:

Etapa: Análise de Requisitos

Análise e Modelagem do Perfil do Usuário e Análise e Modelagem do Contexto

Especificamente acerca da elaboração dos perfis do usuário e seu contexto de uso, propõe-se o uso de questionários e entrevistas como estratégia mais indicada.

Porém, as dificuldades iniciam pela (I) própria concepção dos questionários e entrevistas, com a necessidade de determinar quais dados devem ser apurados, passando pela (II) aplicação destes, onde deve-se definir quais os meios mais indicados para a sua aplicação e culminando na (III) dificuldade de compilar e analisar os dados apurados, de forma a torná-los significativos.

Grande parte dos problemas apontados podem ser, ao menos, minimizados, quando do uso de meios computacionais, seja através do uso de questionários totalmente informatizados, seja a partir da automatização da captura dos dados e, uma vez tendo os dados armazenados em meio informático, sua compilação e análise podem ser delegadas a sistemas computacionais que forneçam os resultados desejados.

Entretanto, como pode ser observado no Quadro 28, foi encontrada apenas uma ferramenta que fornece apoio à Análise e Modelagem do Perfil do Usuário e não foi encontrada ferramenta que contemple a Análise e Modelagem do Contexto.

Portanto, em função de ambas as atividades citadas adotarem metodologia semelhante, a ferramenta apresentada para uma pode ser, após os devidos ajustes, ser utilizada pela outra atividade.

Análise e Modelagem da Tarefa e Avaliação do Modelo da Tarefa

Atividades que possuem quatro ferramentas, com destaque para iTAOS e CTTE.

Tanto o CTTE quanto o iTAOS são recomendadas como ferramentas para a modelagem da tarefa, entretanto a ferramenta iTAOS, por ter sido desenvolvida recentemente, ainda possui uma limitação nos recursos oferecidos.

Etapa: Concepção do Modelo de Interação

Etapa de ampla complexidade por possuir atividades que utilizam dados advindos da etapa anterior além de dados externos (a exemplo de regras ergonômicas).

Análise e Modelagem da Interação e Avaliação do Modelo de Interação

Para a análise, observa-se no Quadro 28 a existência de duas ferramentas: Euterpe e TERESA. A primeira indica ser possível modelar a interação através da notação UAN, porém não esclarece como fazê-lo. Já a ferramenta TERESA permite modelar a interação em conjunto com a ferramenta CTTE. Apesar de seguir um estilo de concepção diferente do utilizado pelo MCI (como poderá ser observado no próximo capítulo), modelar a interação através de TERESA é a melhor das opções disponíveis.

Quanto à avaliação do modelo gerado, dentre as ferramentas apuradas, somente TERESA possibilita a realização desta etapa.

Representação do Cenário

Das ferramentas apuradas, o CTTE possibilita gerar cenários. Entretanto, como poderá ser observado no próximo capítulo, esta concepção é feita em formato diferente daquele proposto no MCI.

Construção do Modelo de Navegação e Avaliação do Modelo de Navegação

Quanto a geração e avaliação do modelo de navegação, observa-se no Quadro 28 que as ferramentas Design/CPN e DENIM, podem ser usadas para a geração deste modelo. Para a avaliação, além das ferramentas citadas, inclui-se o CADP.

Entretanto, para um melhor aproveitamento do potencial fornecido pela ferramenta DENIM, é necessário que o projetista utilize dispositivos especiais (caneta óptica e mesa digitalizadora). Assim, para as equipes que não disponham deste equipamento, a ferramenta pode perder seu atrativo.

O Design/CPN já é utilizado pelo método MCI, porém exige que se tenha conhecimentos do formalismo CPN. Uma alternativa ao uso do Design/CPN é utilizar o recurso de simulação do CTTE que, aliado à ferramenta CADP, permite realizar amplas análises e avaliações do modelo gerado. Entretanto, o CADP, mesmo sendo gratuito, é de difícil aquisição e o CTTE, por si só, possui poucos recursos de análise do modelo.

Etapa: Construção e Avaliação do Protótipo

Geração do Protótipo

DENIM e TERESA são as duas únicas ferramentas, das contidas no Quadro 28, que geram protótipo. Entretanto, a primeira realiza a prototipação tendo como base o modelo da própria ferramenta e que não permite uso de estilos de interação diferentes de *links*.

Já TERESA gera o protótipo a partir do modelo da tarefa gerado pelo CTTE e dispõe de um conjunto maior de estilos de interação.

Avaliação do Protótipo

Já para a avaliação do protótipo gerado, tem-se o Gartner Web Evaluation Tool e o OpUS. A primeira consiste em um documento Excel e, portanto, exige o uso da ferramenta da Microsoft, além de ser voltada para interfaces web, apenas. Já a segunda é de autoria do GIHM e existe, atualmente, tanto em formato Excel quanto em formato HTML.

Capítulo 4 - Formas de Integração

4.1 INTRODUÇÃO

Troca de mensagens, compras através de comércio eletrônico ou pessoas “vivendo” e “trabalhando” dentro de mundos virtuais (a exemplo de StarWars Galaxies e Ultima Online) são apenas alguns exemplos do crescente aumento do uso da Internet.

Aliado a este crescimento, tem-se a diversificação nas formas de interagir com a informação através de uma grande variedade de dispositivos (Figura 40)³⁰.

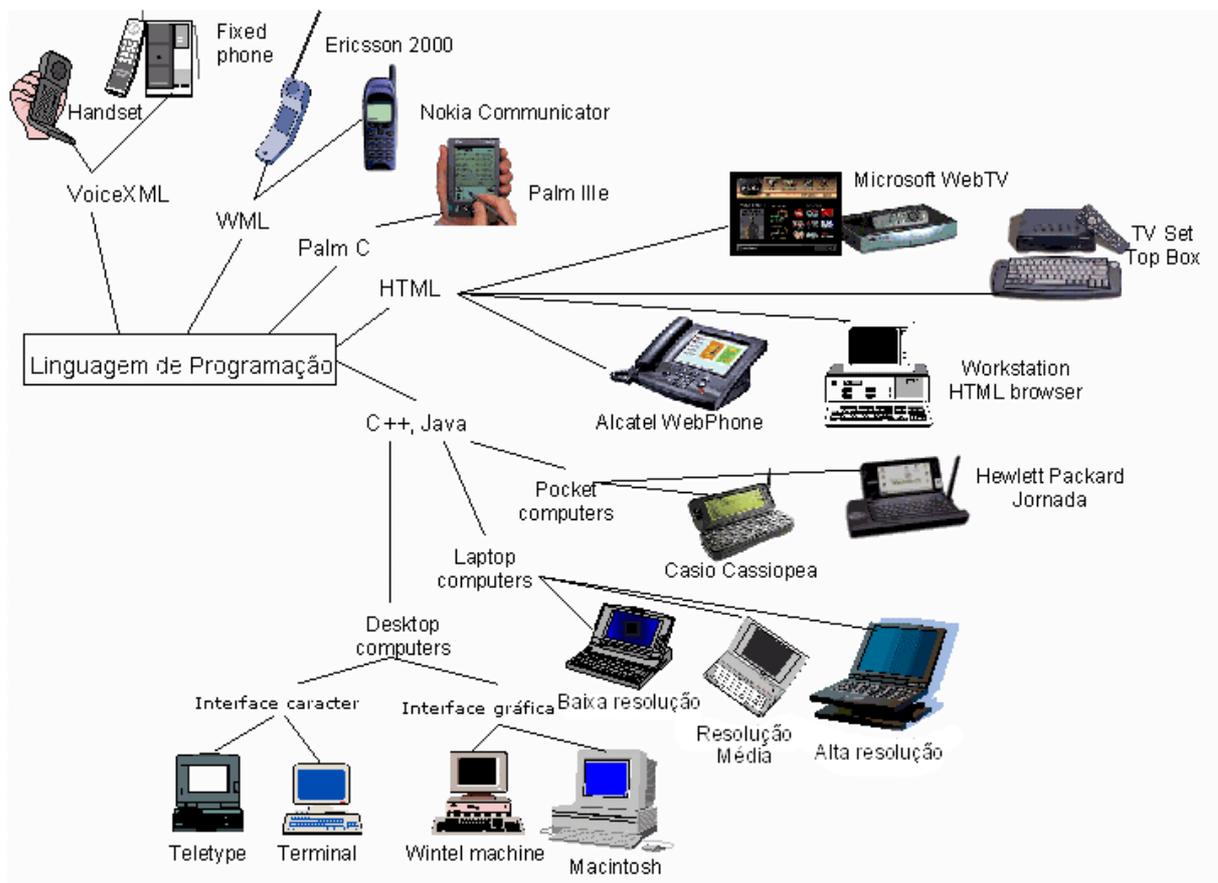


Figura 40 - Diversidade de dispositivos

³⁰ Cabe observar que, além das linguagens apresentadas na Figura 40, há a tecnologia Studio.Net, pertencente a Microsoft. Por seu caráter proprietário, esta tecnologia não foi incluída na referida figura.

Desta forma, atividades como **comércio eletrônico**, que se tornaram bastante populares devido a sua versatilidade e praticidade, exigem atenção redobrada da equipe de desenvolvimento, pois para atender um número maior de usuários, é necessário que o serviço esteja disponível para o maior número de dispositivos. Em virtude de diferenças entre características e recursos dos dispositivos, surge a necessidade de desenvolvimento de diferentes interfaces para cada equipamento. Assim, tem-se o desenvolvimento de tantas interfaces quanto forem os dispositivos que se deseja abranger.

Os atuais esforços no sentido de conceber um modelo que possibilite simplificar o desenvolvimento de diversas interfaces a partir de um meio comum buscam suprir as necessidades da Web, o que não significa que esteja restrito a ela.

O *Hyper Text Markup Language* (HTML), deve ter sido um dos primeiros padrões a serem considerados, em virtude de possuir características tais como:

- ser um padrão de ampla utilização na Web;
- prover um forma para criar e apresentar páginas que podem ser navegadas através de textos e figuras.

Entretanto, o mesmo não provê um mecanismo que descreva o conteúdo e que permita o gerenciamento remoto dos dados.

Outro padrão é *Standard Generalized Markup Language* (SGML)³¹, entretanto, o SGML consiste em um padrão demasiadamente complexo e que não facilita a abrangência desejada.

Desta forma, em 1996, formou-se o grupo *XML Working Group* que, sob o patrocínio do *World Wide Web Consortiun* (W3C), desenvolveu o padrão XML (*eXtensible Markup Language*), cujos objetivos são:

- deve ser facilmente utilizado sobre Internet;
- deve suportar uma ampla variedade de aplicações;
- deve ser compatível com SGML;
- deve ser fácil a escrita de programas que processem documentos XML;

³¹ Padrão descrito na norma ISO 8879 e consiste em um padrão para uso de marcações descritivas (que se preocupam com conteúdo [Godo00]) mescladas a documentos.

- a quantidade de atributos opcionais em XML deve ser a mínima possível, idealmente zero;
- documentos XML devem ser legíveis e razoavelmente claros para o homem;
- *design* de documentos XML deve ser rápido;
- *design* de documentos XML deve ser formal e conciso;
- documentos XML devem ser fáceis de ser criados;
- XML não deve “se preocupar” em sintetizar informações.

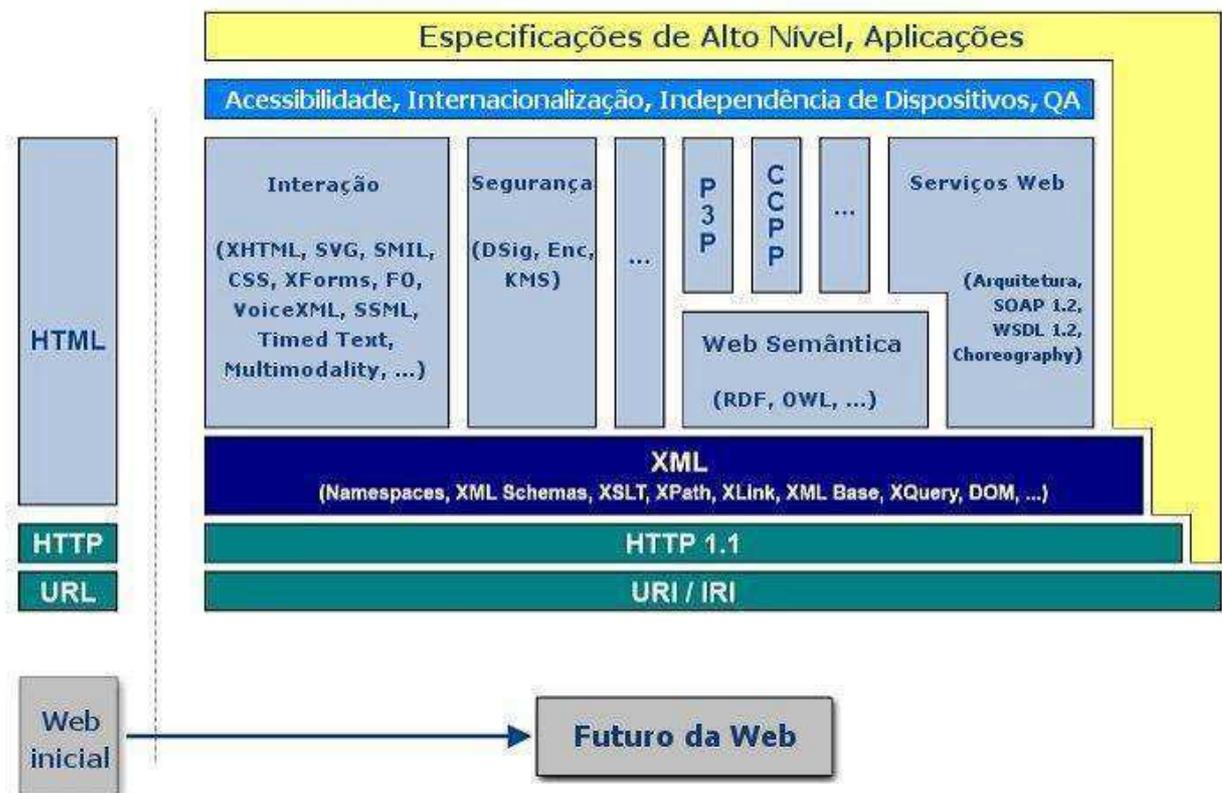


Figura 41 - Arquitetura WEB [W3C00]

A Figura 41 apresenta o projeto de arquitetura para a Web, segundo a W3C, no qual pode ser observada a posição fundamental do padrão XML para os demais formalismos Web.

Por ser um padrão que tem recebido apoio tanto do W3C quanto da indústria, por sua capacidade de descrever concisa e formalmente os dados, o XML consiste em forte opção de integração entre aplicações. Entretanto, por ser um padrão extremamente genérico, para um mesmo tipo de aplicação são utilizadas ontologias diferentes. Diante de tal possibilidade

diversos grupos de pesquisa, dentro e fora da W3C, foram formados objetivando a especificação de ontologias para diversas áreas, tais como:

- WML (*Wireless Markup Language*), visando a telefonia móvel;
- VoiceXML, visando telefonia fixa;
- UIML (*User Interface Markup Language*), visando integrar linguagens para o desenvolvimento de interfaces.

A seção seguinte versará acerca de alguns dos padrões XML relacionados à concepção de interfaces.

4.2 AAIML

AAIML (*Alternate Abstract Interface Markup Language*) é parte integrante do projeto de *Universal Remote Console* (URC) que objetiva auxiliar pessoas, com ou sem deficiências, a operar remotamente qualquer dispositivo eletrônico.

Sob a chancela do comitê técnico V2 do *InterNational Committee for Information Technology Standards* (INCITS) há o desenvolvimento do padrão de Acesso Alternativo a Interface do Usuário.

AAIML é em uma linguagem baseada em XML que será usada para comunicar uma definição abstrata da interface do usuário para um serviço ou dispositivo a ser controlado (ex.: máquinas copadoras ou elevadores). Em função destes dispositivos serem produzidos por empresas diferentes, como requisito fundamental tem-se a necessidade de que a linguagem seja suficientemente abstrata de forma a permanecer independente [Cove03].

4.3 AUIML

Desenvolvido pelo grupo *Ease of Use* da IBM, o AUIML (*Abstract User Interface Markup Language*) consiste em um vocabulário XML com o qual facilita-se a definição de uma ferramenta, independentemente da linguagem de descrição da interface com o usuário [Nune02]. Segundo os autores, o vocabulário tenciona ser independente de plataforma, linguagem ou tecnologia de implementação de interfaces.

Segundo Nunes e Cunha [Nune02], o padrão consiste de dois grandes conjuntos de elementos:

- *Data Model*, no qual especifica-se a interação com o usuário, porém de forma independente da modalidade da interação do usuário;
- *Presentation Model*, no qual especifica-se a apresentação (*look and feel*) da interface com o usuário no grau de especificidade que é esperado pelo renderizador³².

Um mesmo código AUIML pode ser convertido para diversos dispositivos, entretanto exige-se tradutores (*Rendering engine*) específicos para cada interface diferente a ser criada.

4.4 XUL

Criado para facilitar e acelerar o desenvolvimento do *browser* Mozilla, XUL (*XML User-interface Language*) consiste em uma linguagem XML que possui os seguintes atributos chaves e oferece os seguintes benefícios [Boja03]:

- *powerful widget-based markup language*, ou seja XUL é orientada a artefatos da aplicação, tais como: janelas, botões e *links* de hipertexto;
- baseada em padrões existentes, ou seja é baseada em padrões reconhecidos tais como: XML 1.0, HTML 4.0, CSS 1 e 2, entre outros;
- portabilidade de plataforma, a linguagem foi projetada para ser neutra em relação a plataformas, ficando assim mais fácil portar as aplicações para todos os sistemas operacionais onde o Mozilla é executado;
- separação de camadas, de forma a ter uma clara separação entre:
- definição da aplicação do cliente e a lógica do programa, ou o *content* (XUL, XBL³³ e JavaScript);
- a apresentação ou *skin* (CSS e imagens);
- especificação da linguagem ou *locale* (DTDs);

³² Ferramenta (software) capaz de converter o documento AUIML na interface concreta do usuário.

³³ XBL (*eXtensible Bindings Language*) consiste em uma linguagem que permite definir novos elementos especiais para os *widgets* XUL [Boja03].

- fácil personalização, localização ou *branding*, outra facilidade advinda da separação da apresentação da lógica da aplicação, pois desta forma o desenvolvedor pode manter um código base e customizá-lo para diversos clientes;

Entretanto a linguagem restringe seu campo de ação a ambientes nos quais seja executado o *browser* Mozilla, ficando portanto limitada a interfaces baseadas em janelas (*windows-based*).

4.5 UIML

Trabalho iniciado em 1997, idealizado por Marc Abrams do Virginia Tech e apresentado na tese de PhD de Phanouriou Constantinos, a UIML (*User Interface Markup Language*) foi desenvolvida para ser uma linguagem única que permita a descrição de interfaces com o usuário para qualquer dispositivo [Vögl03].

Em [Abra99] Abrams et al. é apresentado o seguinte conjunto de propósitos para a linguagem:

- prover separação natural entre o código da interface e o código da aplicação;
- facilitar o reuso para não programadores;
- reduzir o tempo de desenvolvimento das interfaces para múltiplos dispositivos;
- permitir prototipagem rápida da interface;
- facilitar a internacionalização e localização;
- possibilitar a correta apresentação das interfaces obtidas da rede de computadores através dos *browsers* web;
- incrementar a segurança;
- possibilitar a extensão do suporte para tecnologias futuras.

Segundo Richter, em [Rich03], a interface em UIML é descrita em um nível genérico através de quatro dimensões:

- **estrutura**, representando os elementos existentes e como estão estruturados;
- **estilo**, representando como os elementos são concebidos;
- **conteúdo**, representando o conteúdo textual;
- **comportamento**, representando o que acontece, os eventos.

Em termos de modelos, tem-se a representação de três modelos:

- diálogo;
- apresentação;
- domínio.

A descrição da interface resultante é traduzida (*rendering*) para outras linguagens, tais como WML e HTML ou pode ser diretamente interpretada pelo dispositivo do cliente.

Atualmente, a linguagem encontra-se na versão 3.0.

4.6 XIML

O XIML (*eXtensible Interface Markup Language*) é um subconjunto da meta-linguagem para modelagem da interface com o usuário MIMIC [Vand01] e objetiva representar os elementos da interface através de seus atributos e relacionamentos [Puer01].

Algumas características do produto, segundo Puerta et al.:

- representa aspectos concretos (apresentação e diálogo) e abstratos (contexto) da interface;
- provê uma ponte entre o projeto e o desenvolvimento da interface através de um *framework* com representação única para ambos os processos;
- provê um repositório de conhecimento para operações em tempo de execução tais como personalização, adaptação, contexto e interação baseada em agentes.

XIML, portanto, é uma linguagem de representação da interface para um suporte universal à funcionalidade inerente a todo o ciclo de vida de uma interface com o usuário: projeto, desenvolvimento, operação, gerenciamento, organização e avaliação.

A Figura 42 apresenta a estrutura básica da linguagem XIML [Puer03].

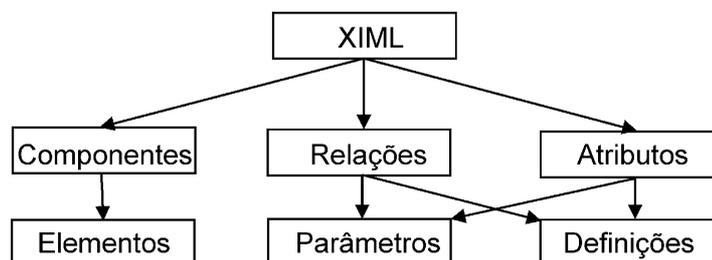


Figura 42 - Estrutura da Linguagem XIML [Puer03]

Em [Puer03], aponta-se que a linguagem é “uma coleção organizada de *elementos* de interface que são categorizados em um ou mais *componentes* da interface”.

Componentes

Em sua primeira versão (XIML 1.0) há a definição de cinco componentes básicos: tarefa, domínio, usuário, diálogo e apresentação; sendo os três primeiros caracterizados por serem contextuais e abstratos enquanto que os dois últimos são concretos e de implementação.

Relações

Consiste na *definição* ou em *parâmetros* que conectam dois ou mais elementos XIML internos a um componente ou entre componentes.

As relações de definição vêm a especificar uma forma canônica da relação enquanto que as relações de parâmetro especificam situações atuais das relações.

Atributos

São características ou propriedades dos elementos que podem ser relacionados a um valor, o qual por sua vez pode ser um conjunto básico de tipos de dados ou a instância de outro elemento.

4.7 CONSIDERAÇÕES

O Quadro 29 apresenta um comparativo entre as linguagens de marcação analisadas. Pode-se observar que, dentre os padrões analisados, merecem destaque: UIML e XIML. O primeiro apresenta um amplo escopo de aplicação, além de prover uma interface plenamente funcional, entretanto as ferramentas de suporte existentes só estão disponíveis mediante pagamento de licença de uso.

Já o padrão XIML possui um menor suporte de linguagens (HTML, WML e Java), porém possui modelos para todas as etapas do ciclo de vida da concepção de interfaces, é mais apropriado para a especificação de interfaces com o usuário para multiplataformas e múltiplos contextos de uso, além de permitir maior facilidade em exportar descrições da interface entre programas. Apesar do XIML possuir algumas restrições quanto ao acesso ao seu conjunto de *tags*, este padrão ainda se mostra como mais indicado para adoção, principalmente por não estar diretamente associado a uma organização privada e por apresentar maior afinidade com o método MCI (devido ao conjunto de modelos contemplados).

Quadro 29 - Quadro comparativo das linguagens de marcação

Linguagem	Modelos	Saída	Metodologia	Ferramenta para suporte	Escopo de aplicação	Suporte a Linguagens
AAIML	- Diálogo - Apresentação	-	Ainda não definida	-	Handheld, <i>smart TV</i>	Ainda não definidas
AUIML	- Diálogo - Apresentação	Interface plenamente funcional	Especificação de uma descrição genérica da interface	<i>Rendering engine</i>	Handheld, <i>Desktop</i>	HTML, DHTML, Java Swing, PalmOS, Wml
XUL	- Diálogo - Apresentação	Interface plenamente funcional	Especificação de múltiplas descrições da interface	<i>Rendering engine</i>	<i>Desktop</i> (aplicações Web que utilizam o <i>browser Mozilla</i>)	XUL
UIML	- Diálogo - Apresentação - Domínio	Interface plenamente funcional	Especificação de múltiplas descrições da interface	<i>Liquid UI*</i>	Handheld, <i>Desktop</i> , Smart, telefones móveis, interfaces vocais	Java, HTML, WML, VoiceXML, C++, PalmOS
XIML	- Tarefa - Domínio - Usuário - Diálogo - Apresentação - Plataforma - Projeto	Interface plenamente funcional	Especificação de múltiplas descrições da interface ou uma genérica	<i>Rendering engine</i> Gerador de código Editor	Handheld, <i>Desktop</i> , Telefones móveis, Terminais Java	HTML, WML, Java

* *rendering engine*, gerador de código e editor.

Capítulo 5 - Aplicação do MCI e de seu suporte computacional a um Estudo de Caso

O presente capítulo apresenta o suporte computacional proposto para o MCI com base nos estudos realizados dos formalismos, ferramentas e padrões de integração. O suporte computacional proposto é apresentado a partir de um estudo de caso, enfatizando para cada etapa do método o suporte computacional recomendado.

5.1 APRESENTAÇÃO DO SUPORTE COMPUTACIONAL PROPOSTO

A Figura 43 representa a relação entre o fluxo da informação no método MCI, em função das ferramentas propostas.

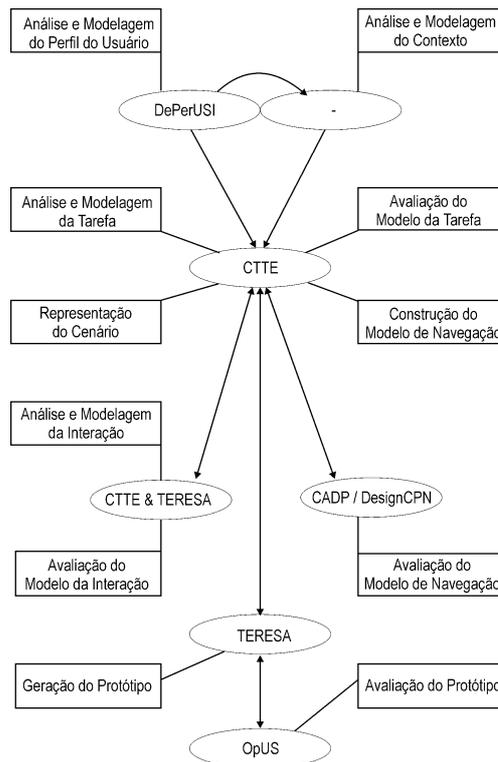


Figura 43 - Método MCI em função das ferramentas

A Figura 43 representa uma alternativa de aplicação do método com base em um número reduzido de ferramentas. Esta opção é recomendada quando o objetivo é um conjunto coeso definido de ferramentas ou quando a equipe de desenvolvimento já possui conhecimentos acerca dos apoios computacionais utilizados. Porém, pode enrijecer demasiadamente o método, de forma a restringir seu uso a uns poucos iniciados em seu ferramental. Por outro lado, caso se opte por aplicar o método de concepção sem depender do uso de ferramentas específicas, a alternativa consiste na definição de um formato padrão de integração, que sirva de meio de comunicação entre o método e as ferramentas.

Como apontado em tópico anterior, existem atualmente diversas alternativas de formatos, tais como AUIML, UIML ou XIIML, cada formato possuindo um escopo de aplicação com grau maior ou menor de abrangência.

Apesar de UIML ser um padrão que permite abranger uma grande gama de escopos de aplicação e estar mais avançado em termos de ferramentas desenvolvidas, o padrão XIIML é o recomendado para ser adotado no MCI. Isto se deve ao fato das especificações XIIML permitirem uma maior interoperabilidade no sentido de exportar, entre programas, as descrições de uma interface com o usuário, além de abranger todos os modelos do processo de concepção de interfaces. Desta forma, como pode ser observado na Figura 44, cada artefato utilizado pelo MCI pode ser especificado através de XIIML. Desta forma, ao invés de limitar o conjunto de ferramentas associadas, abre-se o leque de opções para incorporar uma variedade de ferramentas que façam uso do padrão XIIML ou que permitam converter seus artefatos para o padrão XIIML.

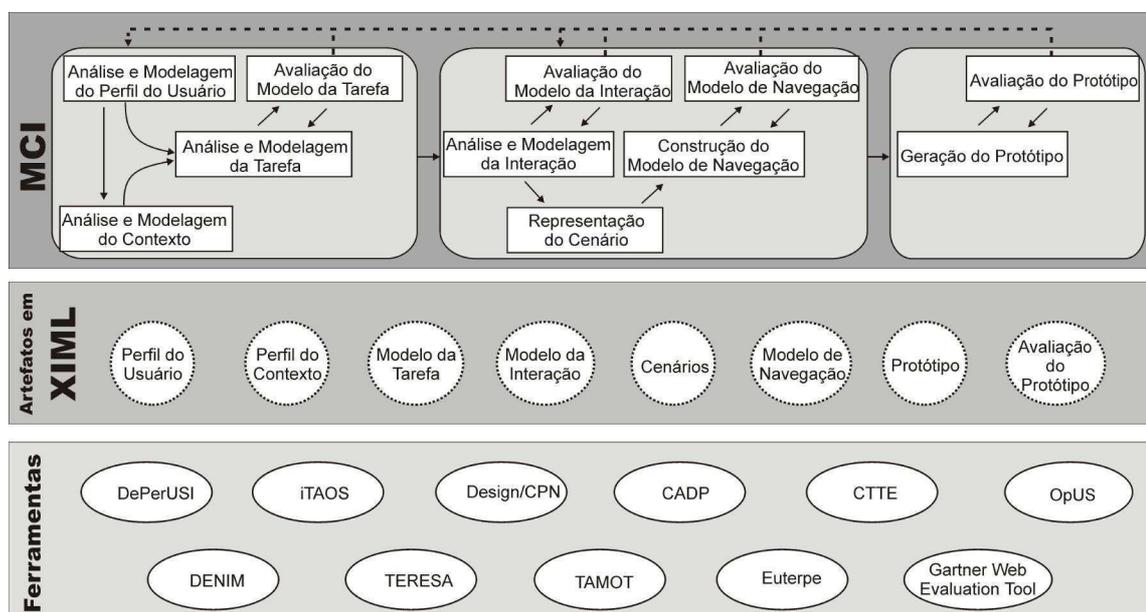


Figura 44 - Estratégia de Integração entre o Método MCI e Ferramentas de Suporte

5.2 ESTUDO DE CASO

Guerrero [Guer02] propõe uma metodologia para concepção de interfaces (MEDITE) e apresenta o exemplo “Um Tutorial na Web” para ilustrar as etapas da metodologia desenvolvida.

A idéia da concepção deste tutorial, segundo Guerrero, surgiu da necessidade de haver um recurso extraclasse na disciplina de “Teoria da Computação” que auxiliasse os alunos a absorver os conceitos teóricos-abstratos do tema: “Máquina de Turing”.

Neste exemplo, foram elencados os seguintes objetivos do usuário:

- adquirir conhecimento a respeito do tema;
- realizar exercícios referentes ao tema;
- buscar outras referências a respeito do tema ou relacionadas a ele;
- estudar tópicos específicos do tema;
- tirar suas dúvidas a respeito do tema;
- contatar pessoas que possam ajudá-lo.

Neste trabalho, será adotado o mesmo caso: “Um Tutorial na Web”, uma vez que se trata de um estudo bem documentado o que permite a comparação entre resultados além de ter seus resultados amplamente aplicados no âmbito do GIHM. Os resultados do estudo de caso original inspiraram a construção de tutoriais das disciplinas “Projeto de Interfaces Homem-Máquina”³⁴, “Arquitetura de Sistemas Digitais”³⁵ e “Informática Industrial”³⁶.

O objetivo principal deste estudo de caso foi avaliar a eficácia da aplicação do suporte computacional proposto, observando os efeitos do seu uso sobre o fluxo de execução do método MCI.

³⁴ Disciplina do programa de Pós-Graduação de Informática e Engenharia Elétrica da UFCG.

³⁵ Disciplina do programa de Graduação em Engenharia Elétrica da UFCG.

³⁶ Disciplina do programa de Graduação e Pós-Graduação em Engenharia Elétrica da UFCG.

5.3 APLICAÇÃO DO SUPORTE COMPUTACIONAL AO ESTUDO DE CASO

É importante ressaltar que este tópico está diretamente relacionado ao fluxo de informações apresentado na Figura 45, o qual, por sua vez, representa as ferramentas e a relação destas com as etapas e processos do MCI.

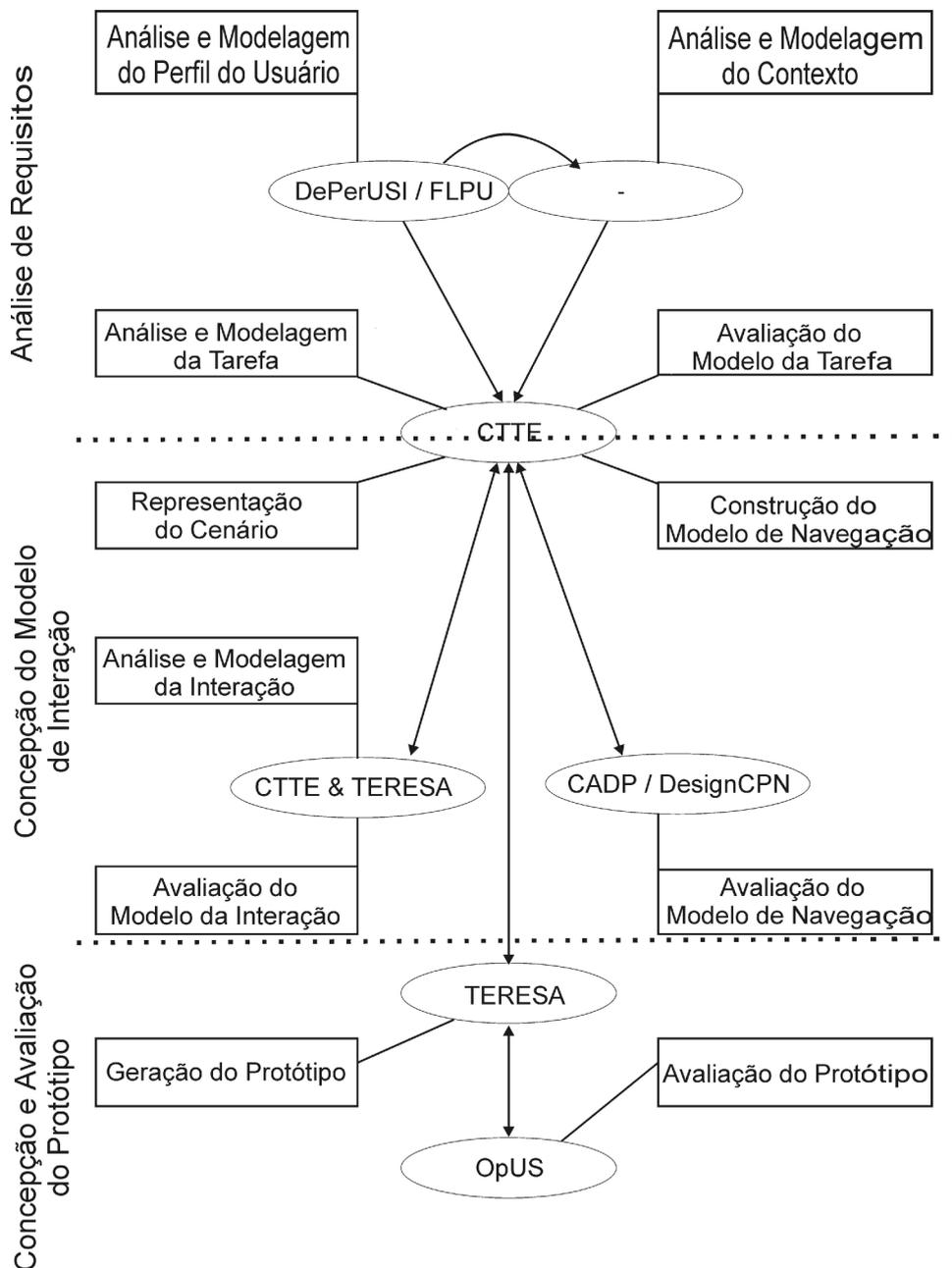


Figura 45 - Ferramentas, Etapas e Processos do MCI

5.3.1 Análise de Requisitos

5.3.1.1 Análise e Modelagem do Perfil do Usuário

Nesta etapa, será adotada a ferramenta DePerUSI para apurar o perfil do usuário, através do uso de questionários. O questionário deve ser aplicado a um número significativo de usuários, os quais devem representar o grupo de usuários pontencial.

Primeiramente, devem ser definidas as informações de interesse, bem como o meio a ser utilizado para realizar o levantamento (papel, meio eletrônico, outros) e o grupo de indivíduos que fará parte do levantamento.

A estrutura do DePerUSI é tal que as questões são agrupadas em:

- questões fixas, as perguntas e seu conjunto de respostas são mantidas (ex.: para a questão sobre o sexo do usuário, as respostas podem ser: masculino ou feminino);
- questões semi-configuráveis, as perguntas são mantidas, porém as possíveis respostas podem ser configuradas (ex.: faixa etária ou propósito de uso da ferramenta);
- questões configuráveis: perguntas que dizem respeito ao contexto apurado, sendo desta forma totalmente configuráveis.

Quanto ao meio a ser utilizado para a realização do levantamento, este dependerá do contexto. Por exemplo, caso se deseje levantar dados sobre usuários na escola pública, em virtude da escassa disponibilidade de meios eletrônicos para levantamento de dados (tais como computadores com acesso à internet) é mais indicado o uso de questionários em papel. Já para apurar dados sobre o grupo de pessoas que estudam na Universidade Federal de Campina Grande (UFCG) e que pertencem ao Departamento de Sistemas de Computação é possível a aplicação do questionamento através de formulários eletrônicos disponibilizados pela Internet.

A escolha do grupo de indivíduos parte da amostra consultada, depende do contexto, a exemplo do estudo de caso, o grupo de pessoas que utiliza o “Tutorial na Web” é compreendido por alunos da UFCG, pertencentes ao Centro de Ciência e Tecnologia, dos departamentos de Sistemas e Computação e Engenharia Elétrica que estejam cursando as disciplinas que possuem os tutoriais.

Após aplicados os questionários, cabe a apuração dos dados que pode ir da reunião das informações apuradas até a realização de análise estatística sobre os dados.

Atualmente, a ferramenta DePerUSI não oferece apoio nem para a elaboração dos questionários nem para as atividades de tratamento das informações coletadas. O Quadro 30 apresenta o conjunto de questões do DePerUSI, no formato elaborado para este estudo de caso.

Outra opção para descrever o perfil do usuário consiste em utilizar o Formulário para o Levantamento do Perfil do Usuário (FLPU), o qual deve ser preenchido pelo projetista a partir de suas crenças sobre o grupo de usuários, completado por dados obtidos a partir da observação ou até de entrevistas. O Quadro 31 apresenta o FLPU, preenchido com os dados relativos ao grupo de usuários do estudo de caso.

A decisão sobre qual opção utilizar para representar o perfil do usuário depende do conhecimento que o projetista detém sobre o grupo de usuários e do tempo disponível para realizar o levantamento.

Quadro 30 - Questões propostas para o Estudo de Caso em questão

DePerUSI - Delineamento do Perfil do Usuário de Sistemas Interativos	
1. Seu nível acadêmico é:	<input type="checkbox"/> Estudante de graduação <input type="checkbox"/> Estudante de pós-graduação
2. Você é do sexo:	<input type="checkbox"/> Masculino <input type="checkbox"/> Feminino
3. Você é:	<input type="checkbox"/> Destro <input type="checkbox"/> Canhoto <input type="checkbox"/> Ambidestro
4. Você usa óculos ou lentes de contato?	<input type="checkbox"/> Sim <input type="checkbox"/> Não
5. Você pertence à faixa etária de:	<input type="checkbox"/> 17 a 24 anos <input type="checkbox"/> 25 a 35 anos <input type="checkbox"/> 35 a 45 anos <input type="checkbox"/> Acima de 45 anos
6. No contexto de suas atividades você utiliza o PRODUTO em:	<input type="checkbox"/> Revisar o conteúdo apresentado em sala de aula <input type="checkbox"/> Reforçar o conteúdo apresentado em sala de aula <input type="checkbox"/> Resolver exercícios disponíveis <input type="checkbox"/> Baixar os conteúdos disponíveis <input type="checkbox"/> Acessar os sites disponíveis <input type="checkbox"/> Ver referências
7. Sua plataforma computacional é:	<input type="checkbox"/> Desktop PC <input type="checkbox"/> Notebook <input type="checkbox"/> MAC <input type="checkbox"/> PocketPC
8. Você tem experiência prévia com sistemas computacionais?	<input type="checkbox"/> Sim <input type="checkbox"/> Não

9. Há quanto tempo você usa sistemas computacionais?	<input type="checkbox"/>	Menos de 3 meses
	<input type="checkbox"/>	Entre 3 meses e 1 ano
	<input type="checkbox"/>	Mais de 1 ano
10. Com que frequência você usa sistemas computacionais?	<input type="checkbox"/>	Estou usando pela 1ª vez
	<input type="checkbox"/>	Diariamente
	<input type="checkbox"/>	Menos de 1 vez por semana
	<input type="checkbox"/>	1 vez por semana
	<input type="checkbox"/>	Mais de 1 vez por semana
	<input type="checkbox"/>	Menos de 1 vez por quinzena
	<input type="checkbox"/>	1 vez por quinzena
	<input type="checkbox"/>	1 vez por mês
11. Você tem experiência prévia no uso da Internet?	<input type="checkbox"/>	Sim
	<input type="checkbox"/>	Não
12. Há quanto tempo você acessa a Internet?	<input type="checkbox"/>	Menos de 3 meses
	<input type="checkbox"/>	Entre 3 meses e 1 ano
	<input type="checkbox"/>	Mais de 1 ano
13. Com que frequência você acessa a Internet?	<input type="checkbox"/>	Estou usando pela 1ª vez
	<input type="checkbox"/>	Diariamente
	<input type="checkbox"/>	Menos de 1 vez por semana
	<input type="checkbox"/>	1 vez por semana
	<input type="checkbox"/>	Mais de 1 vez por semana
	<input type="checkbox"/>	Menos de 1 vez por quinzena
	<input type="checkbox"/>	1 vez por quinzena
	<input type="checkbox"/>	1 vez por mês
14. Qual o local onde você costuma acessar a Internet?	<input type="checkbox"/>	Em casa
	<input type="checkbox"/>	No trabalho
	<input type="checkbox"/>	Na universidade
15. Se você respondeu Em casa no item anterior, especifique o tipo de acesso a Internet que possui:	<input type="checkbox"/>	Discado
	<input type="checkbox"/>	Via rádio
	<input type="checkbox"/>	Via cabo
	<input type="checkbox"/>	ADSL
16. Se você respondeu No trabalho no item 14, especifique o tipo de acesso a Internet que possui:	<input type="checkbox"/>	Discado
	<input type="checkbox"/>	Via rádio
	<input type="checkbox"/>	Via cabo
	<input type="checkbox"/>	ADSL
17. Você tem experiência prévia com o PRODUTO?	<input type="checkbox"/>	Sim
	<input type="checkbox"/>	Não
18. Há quanto tempo você usa o PRODUTO?	<input type="checkbox"/>	Menos de 3 meses
	<input type="checkbox"/>	Entre 3 meses e 1 ano
	<input type="checkbox"/>	Mais de 1 ano
19. Com que frequência você usa o PRODUTO?	<input type="checkbox"/>	Estou usando pela 1ª vez
	<input type="checkbox"/>	Diariamente
	<input type="checkbox"/>	Menos de 1 vez por semana
	<input type="checkbox"/>	1 vez por semana
	<input type="checkbox"/>	Mais de 1 vez por semana
	<input type="checkbox"/>	Menos de 1 vez por quinzena
	<input type="checkbox"/>	1 vez por quinzena
	<input type="checkbox"/>	1 vez por mês
20. Qual a forma de ajuda sobre o assunto que você costuma utilizar mais frequentemente?	<input type="checkbox"/>	Dúvidas
	<input type="checkbox"/>	Contatos
21. Qual a forma de ajuda sobre o site que você costuma utilizar mais frequentemente?	<input type="checkbox"/>	Mapa
	<input type="checkbox"/>	Esquema
22. Você tem experiência prévia com produtos similares?	<input type="checkbox"/>	Sim
	<input type="checkbox"/>	Não

Quadro 31 - Formulário FLPU

Formulário para o Levantamento do Perfil do Usuário	
© 2004, Laboratório de Interfaces Homem-Máquina, UFCC	
	
Características do usuário, escolhidas pelo projetista, de acordo com a relevância, para o projeto.	
Levantamento baseado em:	
Fatos <input checked="" type="checkbox"/>	Opinião do usuário <input type="checkbox"/>
Dados medidos ou observados <input type="checkbox"/>	
Parte I - Características Gerais	
Faixa etária: Entre 17 e 50 anos	
Sexo: Ambos	
Habilidades necessárias para executar a tarefa: Manejo de sistemas computacionais	
Níveis de percepção (ex. acuidade visual): Normais	
Habilidades motoras (ex. velocidade, precisão): Normais	
Grau de instrução (ex. técnico, superior): Cursando nível superior	
Função desempenhada na Organização (ex. gerência, atendimento ao público): Estudante	
Tarefas realizadas na Função (ex. cadastramento, supervisão): Estudo	
Objetivos (o que pretende com o sistema): Aprofundar e/ou reforçar aprendizado da sala de aula	
Motivações (por que usaria o sistema): Reforçar seu conhecimento acerca do assunto disponível	
Preferências (ex. uso do teclado, preenchimento do formulário): Uso de teclado e mouse	
Parte II - Conhecimento Conceitual necessário à execução das tarefas:	
<u>Conhecimento Semântico</u>	Nível de experiência ¹
Função: Estudante	Experiente
Método: Navegando na Web	Experiente
Tarefa: Consultar Tutorial na Web	Noviço
Computadores:	Experiente
Ferramentas utilizadas na execução das tarefas (ou similares): Teclado	Experiente
Mouse	Experiente
<u>Conhecimento Sintático</u>	Nível de experiência
Uso de teclado e mouse: sim	Experiente
Uso de dispositivos especiais de interação: não	-
Uso de terminologia específica: sim, referente ao conteúdo do tutorial	Experiente
Parte III - Estilo Cognitivo	
Aprendizado (ex. treinamento):	Por tentativa e erro
Capacidade de solucionar problemas (sozinho, com ajuda):	Sozinho ou com o auxílio da ajuda on-line
Capacidade de reter o aprendizado (alta, média, baixa):	Facilidade
Personalidade:	
Nível de curiosidade (baixo, médio, elevado):	Elevado
Nível de persistência (baixo, médio, elevado):	Elevado
Nível de inovação (baixo, médio, elevado):	Médio
Inovador <input checked="" type="checkbox"/>	Conservador <input type="checkbox"/>
Impulsivo <input type="checkbox"/>	Reflexivo <input type="checkbox"/>
¹ nível de experiência: noviço, experiente, especialista, ...	
Laboratório de Interfaces Homem-Máquina LIHM DEE-CCT/UFCC - Tel: 083 310 13 87 Prof. Fátima Vieira Turnell, e-mail fatima@dee.ufcg.edu.br	

5.3.1.2 Análise e Modelagem do Contexto

Apesar do DePerUSI ter sido desenvolvido para apurar o perfil do usuário, algumas das perguntas (tal como sobre a plataforma computacional utilizada), permitem inferir sobre aspectos do contexto de uso do produto.

Como pode ser observado pela Figura 45 não há nenhuma ferramenta associada a esta etapa. Segundo a norma ISO 9241, parte 11, o contexto de uso consiste na união das descrições do: **usuário**, **tarefa**, **equipamento** e **ambiente** [ISO98]. Ainda, segundo a norma ISO, em [ISO98], caberiam alguns novos questionamentos, tais como:

- número de horas de trabalho;
- monitoramento de performance;
- condições térmicas e auditivas;
- risco de periculosidade, entre outras.

Portanto, entende-se que não há a necessidade de uma nova ferramenta para esta etapa, mas sim da adição de novos questionamentos ao DePerUSI, de forma a dotá-lo de condições de apurar sobre o contexto.

5.3.1.3 Análise, Modelagem e Avaliação do Modelo da Tarefa

Como parte do levantamento de requisitos, além do levantamento do perfil do usuário e do contexto de uso, é necessária a modelagem da tarefa.

O modelo criado através da ferramenta CTTE possui um conjunto mais amplo de informações do que aqueles exigidos pelo MAD, justamente por isto, algumas informações provenientes do perfil do usuário e do contexto já são aproveitadas neste momento do processo (a exemplo da plataforma em que o software será utilizado, tempos e frequência de execução da tarefa).

Como foi discutido no Capítulo 2, o formalismo CTT possui um conjunto amplo de operadores, além de permitir a distinção entre os agentes de cada tarefa, os quais podem ser: abstratos, usuário, sistema, ou a interação entre usuário e sistema. Estas peculiaridades fazem diferença no momento da modelagem da tarefa, uma vez que o modelo tende a explicitar não só **o que** deve ser feito, como também já introduz a informação de **como** será feito.

A análise e modelagem da tarefa concentra informações sobre as tarefas e a ordem em que são executadas (ver Quadro 32).

Quadro 32 - Informações iniciais para a modelagem da tarefa

<p>Análise e Modelagem da Tarefa</p>	<p>Etapa inicial</p>	<ul style="list-style-type: none"> - Tarefas - Identificador de cada tarefa - Operador temporal associado
---	----------------------	--

A Figura 46 representa a sub-árvore da tarefa “**Estudar**” modelada na ferramenta CTTE. Nela pode ser observado que inicialmente não há a necessidade de serem apresentados todos os detalhes sobre cada tarefa.

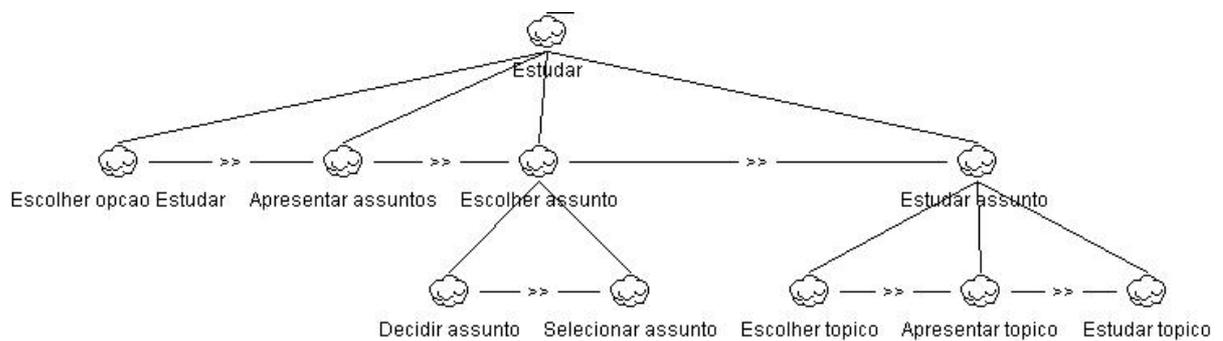


Figura 46 - Sub-árvore Estudar (versão 1)

Já na Figura 47, pode ser observada a definição das categorias para cada tarefa (a exemplo da tarefa “**Escolher opção Estudar**” que na Figura 46 é apresentada como sendo *Abstrata* e na Figura 47 já é apresentada como sendo *Interação*).

Na Figura 49, pode ser observada a evolução na modelagem da sub-árvore, a começar pela definição das categorias (interativa, abstrata, aplicação ou usuário), passando-se pela inclusão de novas tarefas e detalhamento das já existentes além da redefinição dos operadores entre as tarefas.

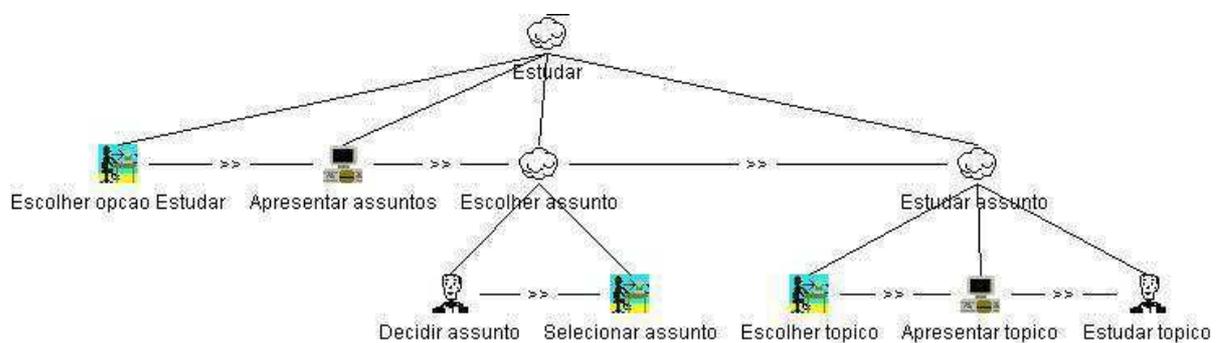


Figura 47 - Sub-árvore Estudar (versão 2)

Após utilizado o recurso de simulação, observou-se que haveria a necessidade de definir novas tarefas, portanto na Figura 48 vê-se:

- a inclusão de novas tarefas, tais como:
- a opção de “**Cancelar**”, uma vez apresentados os assuntos;
- a opção de “**Decidir tópico**”, antes de selecioná-lo.
- a alteração de alguns identificadores:
- a tarefa “**Escolher tópico**” (Figura 47) foi renomeada para “**Selecionar tópico**” (ver Figura 48);
- a modificação e/ou inclusão de alguns operadores temporais:
- ao invés do operador *enabling*, o uso do operador *enabling with information passing* entre as tarefas “**Selecionar tópico**” e “**Apresentar conteúdo**” (Figura 48).

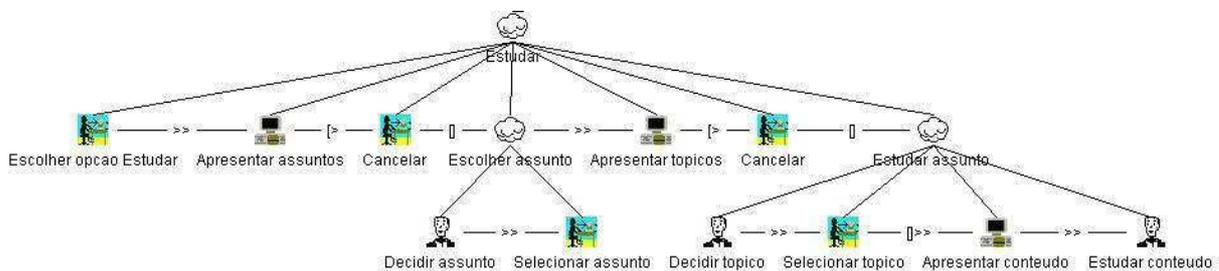


Figura 48 - Sub-árvore Estudar (versão 3)

A versão 4 da sub-árvore pode ser vista na Figura 49 onde pode ser observado que a tarefa “**Estudar assunto**”, que na versão 3 estava no mesmo nível que as demais tarefas, na versão 4 foi subordinada à tarefa “**Escolher assunto**”. Desta forma, não só o fluxo de execução está melhor representado como a própria hierarquia da tarefa.

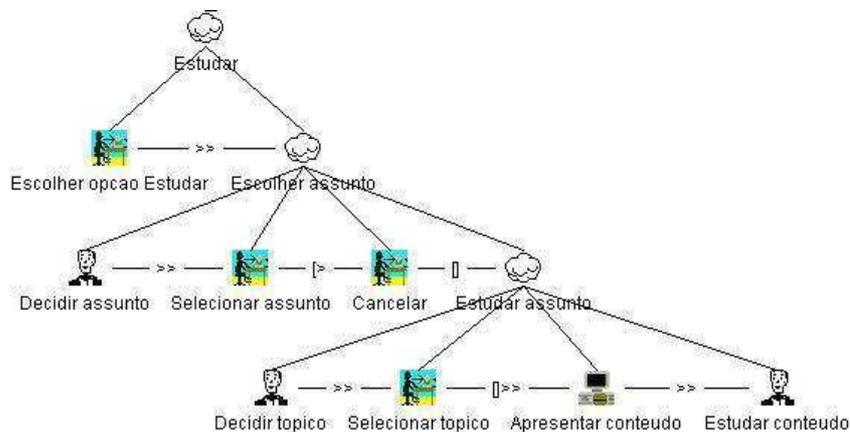


Figura 49 - Sub-árvore Estudar (versão 4)

A Figura 50 representa a versão final para a tarefa “**Estudar**”. Durante todo o processo de modelagem pode ser utilizada a ferramenta de simulação (desde que observadas

as exigências mínimas para execução da ferramenta, ou seja, que as tarefas possuam operadores temporais interligando-as).

Através do recurso *Check Model Structure*, é possível verificar a completude do modelo, exigida para a perfeita execução do simulador. Cabe observar que este recurso apenas verifica o modelo, indicando se há erros e quais são:

- tarefas que não possuem operador associado;
- tarefas que possuem subtarefas incompatíveis com sua categoria (ex.: tarefas interativas podem possuir apenas subtarefas interativas);
- tarefas abstratas que possuem todas as suas subtarefas da mesma categoria (ex.: todas as subtarefas são da categoria aplicação);

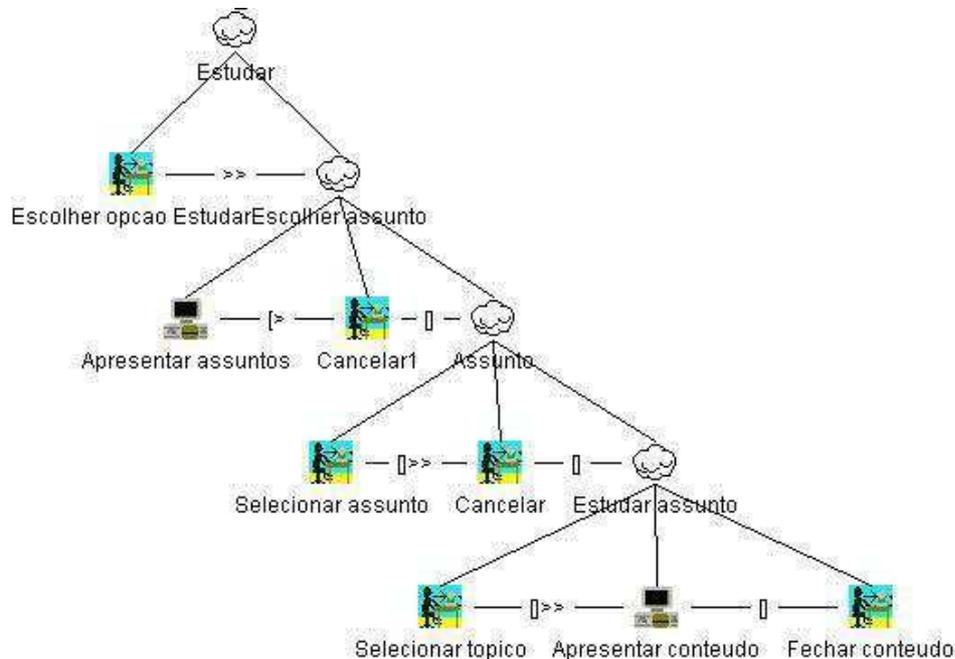


Figura 50 - Sub-árvore Estudar (versão final)

Outra etapa importante na criação do modelo de tarefas vem a ser o preenchimento dos descritores das tarefas. No caso da ferramenta CTTE, o projetista pode fazê-lo no momento que achar mais apropriado (seja desde o primeiro momento da modelagem, seja durante a modelagem, seja após finalizado o modelo de tarefas).

A definição dos descritores, como pode ser observado no Quadro 33, já faz uso de informações advindas da Análise e Modelagem do Perfil do Usuário (tais como a frequência e os tempos de execução de cada tarefa) e da Análise e Modelagem do Contexto (tal como a definição da plataforma).

Quadro 33 - Informações finais da modelagem da tarefa

Análise e Modelagem da Tarefa	Etapa inicial	- Tarefas - Identificador de cada tarefa - Operador temporal associado
	Etapa seguinte	- Definição das Categorias - Refinamento do modelo
	Descritores	- Definição do Tipo - Definição da Frequência - Definição da Plataforma - Definição dos tempos mínimo e máximo para execução da tarefa - Definição dos objetos usados na tarefa

Informações como Tipo e Objetos, bem como a própria Categoria da tarefa estão mais relacionadas à Análise e Modelagem da Interação e como poderá ser observado mais à frente são diretamente utilizadas pela ferramenta TERESA.

Outros recursos disponíveis no CTTE para a verificação do modelo gerado são:

- *Reachability Analysis*, que permite ao projetista verificar os possíveis caminhos existentes entre duas tarefas, tendo ou não uma tarefa intermediária. Para tarefa inicial: **Escolher opcao Exercicios** e tarefa final: **Fechar ver duvidas**, são apresentados duas opções diferentes de caminhos (Quadro 34). Já para tarefa inicial: **Apresentar exercicio**, tarefa intermediária: **Ver duvidas** e tarefa final: **Fechar exercicio**, é apresentada apenas uma opção de caminho (Quadro 35);

Quadro 34 - Reachability Analysis com definição de tarefa Inicial e Final

Tarefas inicial e final	Caminhos disponíveis	
	Opção 1	Opção 2
Tarefa Inicial: Escolher opcao Exercicios Tarefa Final: Fechar ver duvidas	- Escolher opcao Exercicios - Cancelar2 - Escolher opcao Ajuda - Escolher Sobre assunto - Ver duvidas - Apresentar duvidas - Fechar ver duvidas	- Escolher opcao Exercicios - Selecionar exercicio - Apresentar exercicio - Escolher opcao Ajuda - Escolher Sobre assunto - Ver duvidas - Apresentar duvidas - Fechar ver duvidas

Quadro 35 - *Reachability Analysis* com definição de tarefa Inicial, Final e Intermediária

Tarefas inicial, final e intermediária	Caminho disponível
Tarefa Inicial: Apresentar exercicio	- Apresentar exercicio
Tarefa Final: Fechar exercicio	- Escolher opcao Ajuda
Tarefa Intermediária: Ver duvidas	- Escolher Sobre assunto
	- Ver duvidas
	- Apresentar duvidas
	- Fechar ver duvidas
	- Voltar nivel
	- Escolher opcao Exercicios
	- Seleccionar exercicio
	- Fechar exercicio

- *Task Model Statistics*, apresenta um relatório (Figura 51) sobre o modelo, com informações sobre: o número total de tarefas, quantidade de níveis, entre outras;

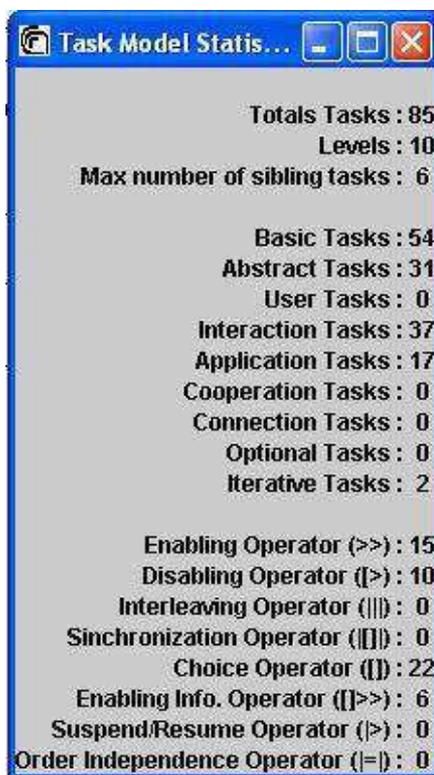


Figura 51 - Relatório *Task Model Statistics*

Compare, recurso similar ao *Task Model Statistics*, porém com o diferencial de apresentar um relatório comparando dois modelos (Figura 52). Consiste de um relatório gerado pelo CTTE, comparando o modelo final do tutorial³⁷ gerado com o ferramental e o

³⁷ Gerado utilizando os recursos e possibilidades do CTTE.

método MCI (primeira coluna de valores) e o modelo original³⁸, resultante da aplicação do MEDITE (segunda coluna de valores).

- há 56 tarefas diferentes entre os dois modelos;
- um modelo possui 10 níveis, enquanto que o outro possui 4, o que pode significar uma diferença no grau de complexidade entre os modelos, pois o primeiro possui maior profundidade de árvore que o segundo;
- enquanto o modelo final possui 54 tarefas básicas³⁹, o modelo original possui 17. Podendo significar que o primeiro está mais detalhado na construção das tarefas, chegando ao ponto das ações ou que está demasiadamente prolixo, havendo portanto a necessidade de rever o modelo gerado;
- o modelo final possui 17 tarefas de aplicação, enquanto que o modelo original não possui nenhuma. Esta informação *de per si* pode não possuir valor algum porém quando observados que, para o modelo original existem 29 tarefas e todas são abstratas pode-se inferir que não há, ao menos em termos de modelo, a existência de tarefas a serem realizadas pelo usuário, pela aplicação ou pela interação entre ambos.

Desta forma, uma mesma aplicação, quando modelada por diferentes projetistas pode ser comparada e, após apuradas diferenças, tais como a quantidade e a diversidade de operadores utilizados, ou ainda a quantidade de tarefas de cada categoria, pode-se decidir por quais aprimoramentos devam ser feitos em uma versão do modelo da aplicação.

Há ainda outros recursos existentes no CTTE que merecem destaque:

- a especificação da plataforma, de forma que para uma mesma tarefa, podem ser definidas diferentes plataformas. Assim, uma tarefa T1 pode ser executada tanto em telefones celulares quanto em *desktops*, enquanto que uma tarefa T2 só será executada em *desktops*. No caso do modelo gerado, todas as tarefas foram definidas para serem executadas em *desktops*;
- *Model Filter*, recurso que permite filtrar o modelo em função da plataforma. Assim, a ferramenta gera um novo modelo tendo como base apenas a plataforma

³⁸ Proposto por Guerrero porém modelado utilizando o CTTE.

³⁹ Ações elementares ou tarefas folhas.

escolhida. Este recurso não foi utilizado no estudo de caso em questão em virtude de haver apenas uma plataforma definida para todo o modelo;

- *Priority Tree*, através do qual a ferramenta “varre” o modelo redesenhando-o em função da prioridade de execução dos operadores. Recurso utilizado em modelos de teste porém não aplicado no estudo de caso modelado.

			Difference	Details
Totals Tasks :	85	29	56	→
Levels :	10	4	6	
Max number of sibling tasks :	6	6	0	
Basic Tasks :	54	17	37	→
Abstract Tasks :	31	29	2	→
User Tasks :	0	0	0	→
Interaction Tasks :	37	0	37	→
Application Tasks :	17	0	17	→
Cooperation Tasks :	0	0	0	→
Connection Tasks :	0	0	0	→
Optional Tasks :	0	0	0	→
Iterative Tasks :	2	0	2	→
Enabling Operator (>>):	15	7	8	
Disabling Operator (<<):	10	0	10	
Interleaving Operator ():	0	0	0	
Synchronization Operator ():	0	0	0	
Choice Operator ():	22	9	13	
Enabling Info. Operator (>>):	6	0	6	
Suspend/Resume Operator (>):	0	0	0	
Order Independence Operator (=):	0	0	0	

Figura 52 - *Task Model Compare* entre os modelos

5.3.2 Concepção do Modelo de Interação

5.3.2.1 Representação dos Cenários

Como apontado em capítulo anterior, o uso de cenário objetiva representar situações reais defrontadas pelo usuário no seu dia-a-dia de forma a: (I) permitir uma melhor captura do uso do sistema a ser desenvolvido e (II) ampliar o canal de comunicação projetista-usuário de forma a garantir a construção da interface desejada.

Portanto modelar um cenário permite uma melhor comunicação entre membros da equipe de projetistas e entre a equipe e o cliente, facilitando a discussão de possíveis soluções

propostas para a interação. Objetiva facilitar a decisão sobre o que prototipar (casos críticos e/ou problemáticos).

O cenário apresentado na Figura 53 consiste em um caso de uso (*Use Case*) modelado segundo UML, representando a tarefa “**Estudar**”. Observa-se que é possível identificar os fluxos que são iniciados pelo Aluno (ex.: **Acessar o Tutorial** ou **Escolher estudar**) e fluxos que são iniciados pelo sistema (ex.: **Apresentar opções** ou **Apresentar assuntos**).

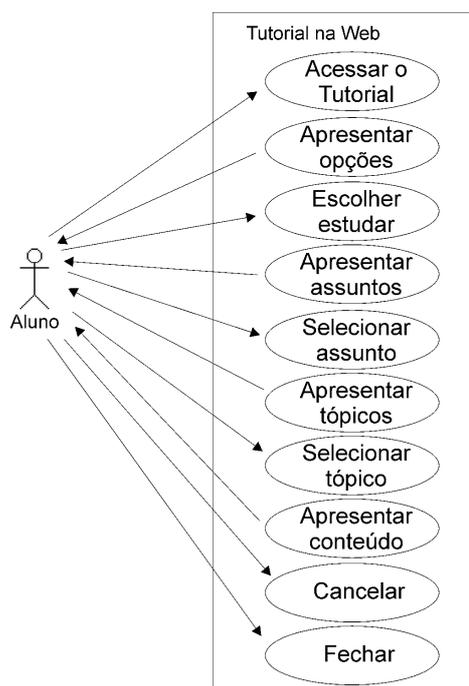


Figura 53 - Cenário da Tarefa Estudar através de Use Case

O cenário permite, portanto, restringir conjuntos a serem prototipados e, desta forma, verificar se o conjunto de estilos de interação propostos condiz com as expectativas do projeto. De outra parte, pode-se gerar vários protótipos a partir de um mesmo cenário para que, desta forma, possam ser testados diferentes alternativas de interação para então decidir qual corresponde à melhor opção.

Apesar da ferramenta CTTE possuir um recurso que permite diagramar casos de uso, esta função não está plenamente funcional, além de não possuir qualquer significado para o CTTE. Por outro lado, há um recurso que permite criar cenários durante a simulação do modelo da tarefa gerado. Para tanto, uma vez iniciada a simulação do modelo percorre-se, manualmente, o caminho referente ao cenário desejado. Assim, caso se deseje um cenário da atividade **estudar um assunto específico**, segue-se o seguinte conjunto de passos:

- Escolher opção Estudar;

- Selecionar assunto;
- Selecionar topico;
- Fechar conteúdo.

Este conjunto pode ser arquivado sob o formato SCN e re-utilizado para a verificação da execução do cenário. Entretanto, esta concepção de cenário está muito distante do que este recurso significa para o MCI. Um cenário, no CTTE, é gerado dentro do simulador e representa um conjunto de tarefas que podem ser usadas para verificar um modelo. Porém, pelo fato deste cenário estar vinculado a tarefas, a mera mudança no nome identificador de uma das tarefas acarreta em tornar obsoleto o cenário gerado. Assim, caso no modelo do estudo de caso ser alterado o identificador da tarefa **“Escolher opcao Estudar”** para **“Selecionar opcao Estudar”**, ao ser re-aplicado o cenário, a ferramenta indicará que a tarefa **“Escolher opcao Estudar”** não existe e que portanto o cenário é incompatível com o modelo.

5.3.2.2 Construção do Modelo de Navegação

O modelo de navegação deve permitir “sintetizar as situações de transição mais comuns em uma interface com o usuário (independente dos dispositivos de interação)” [Scai01, Turn01]. O primeiro modelo gerado foi desenvolvido por Sousa, em [Sous99] e buscava modelar o contexto da navegação em um *browser*. Este modelo foi aprimorado por Scaico, em [Scai01] (ver Figura 54), buscando incorporar novos mecanismos de navegação não contemplados no modelo original. Apesar de aplicado ao *browser*, o modelo original tem caráter genérico e poderia ser aplicado a este estudo de caso. No entanto, considerando que o tutorial não tem restrições de tempo associadas, esta etapa do MCI não foi realizada.

Embora o modelo em CPN possibilite realizar análises mais aprofundadas, a alternativa adotada no Estudo de Caso, consistiu no uso do recurso de simulação da ferramenta CTTE, através do qual pode-se observar as seqüências de transição disponíveis em um dado momento. O Quadro 36 apresenta uma síntese comparativa entre a modelagem utilizando Design/CPN e CTTE.

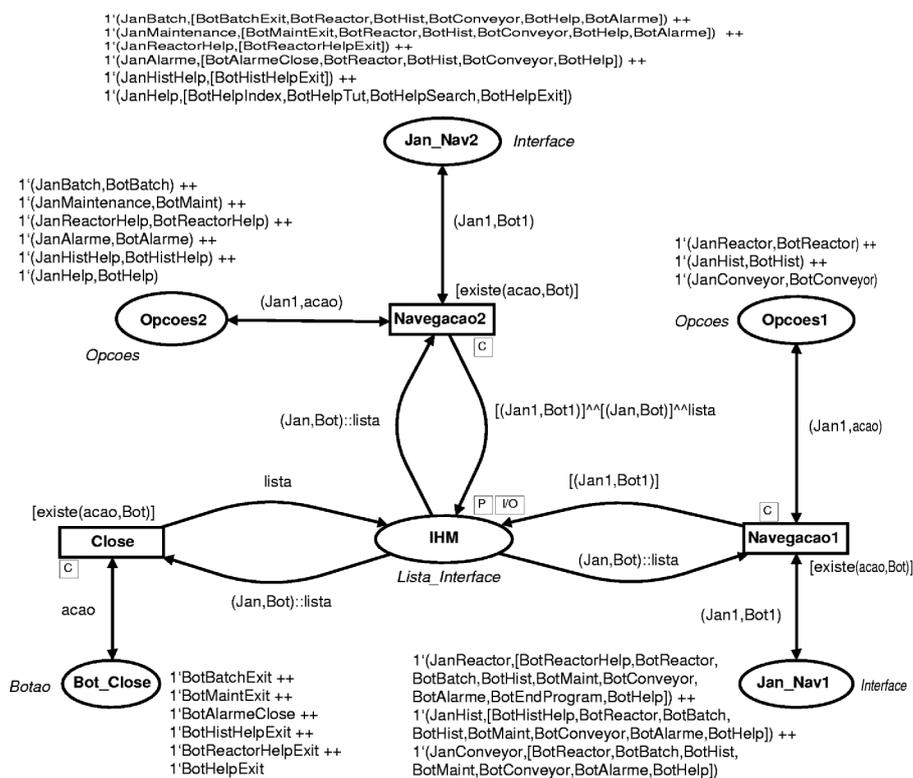


Figura 54 - Modelo de Navegação em CPN

Quadro 36 - Comparativo entre as opções de modelagem da navegação

Design/CPN	CTTE
Utiliza a ferramenta Design/CPN.	Utiliza a ferramenta CTTE.
Exige que o projetista conheça o formalismo CPN.	Não exige conhecimento do formalismo, uma vez que o modelo da tarefa gerado é utilizado na simulação.
Exige que o projetista saiba utilizar a ferramenta Design/CPN.	Exige que o projetista saiba utilizar o recurso de simulação fornecido pelo CTTE.
Oferece amplo conjunto de análises.	Oferece conjunto restrito de análises.
Permite simular, de forma automática ou manual.	Permite simular de forma manual
Permite acompanhar visualmente o transcorrer da simulação.	Permite acompanhar visualmente o transcorrer da simulação.
Uso da ferramenta está restrito ao ambiente Linux	Multiplataforma.

5.3.2.3 Avaliação do Modelo de Navegação

Independente de estilos e dispositivos de interação, o modelo é concebido de forma a sintetizar as situações de transição mais comuns em uma determinada interface.

A avaliação deste modelo visa o levantamento de informações, tais como:

- existência de caminhos alternativos para execução de uma mesma tarefa, tanto extensos (usuário leigo) quanto otimizados (usuário *expert*);
- existência de bloqueios na navegação, ou seja, se existem pontos nos quais o usuário não possui opções de navegação;

- existência de reversibilidade quando assim for demandado, ou seja, permite-se ao usuário retornar ações (*undo*);
- fácil acesso à ajuda;
- fácil acesso à saída do sistema;

Atualmente, o MCI faz uso do modelo criado utilizando-se o Design/CPN. Entretanto, como apresentado no Quadro 36, existem diversos fatores que dificultam o seu uso pelos desenvolvedores. Uma alternativa seria utilizar a ferramenta CTTE aliada à ferramenta CADP. Como apresentado no Capítulo 3, o CADP consiste em suporte computacional objetivando: compilar eficientemente; simular; verificar formalmente; e testar descrições escritas na linguagem LOTOS. Uma vez que a ferramenta CTTE possui recurso que transforma o modelo da tarefa em linguagem LOTOS, pode-se analisar o modelo gerado com abrangência equivalente àquela fornecida pelo Design/CPN. Um ponto bastante desfavorável ao uso do CADP consiste na dificuldade em se obter a ferramenta (mesmo sendo de uso gratuito).

5.3.2.4 Análise, Modelagem e Avaliação do Modelo da Interação

Segundo o MCI, modelar a interação consiste em, primeiramente, apurar os objetos e ações envolvidas na realização da tarefa e, após, associá-los a objetos e ações da interface.

Assim, para o cenário proposto anteriormente (Cenário da Tarefa Estudar), tem-se o modelo de interação apresentado no Quadro 37.

Quadro 37 - Modelo de Interação da Tarefa Estudar

Identificação da Tarefa	Modelo da Tarefa			Modelo da Interação		
	Ação	Objeto	Complexidade	Estilo de Interação	Objeto da Interface	Agrupamento Visual
Consultar Tutorial na Web	Ativar	Tutorial	Média	Tela	Janela	Janela Principal
Estudar	Ativar	Opção Estudar	Média	Tela	Menu principal	Janela Principal
Escolher Opcao Estudar	Selecionar	Opção	Baixa	Menu	Menu principal opção “Estudar”	Janela Principal
Escolher assunto	Ativar	Apresentação assunto	Média	Tela	Janela	Janela Estudar_assunto
Apresentar assuntos	Apresentar	Assunto	Média	Agrupar	Lista de assuntos disponíveis	Janela Estudar_assunto
Cancelar1	Ativar	Cancelar	Baixa	Controle	Botão de Cancelar	Janela Estudar_assunto
Assunto	Ativar	Opções assunto	Média	Tela	Janela	Janela Estudar_assunto
Selecionar assunto	Selecionar	Opção	Baixa	Menu	Caixa de seleção simples	Janela Estudar_assunto
Cancelar	Ativar	Cancelar	Baixa	Controle	Botão de Cancelar	Janela Estudar_tópico
Estudar assunto	Ativar	Opções tópicos	Média	Tela	Janela	Janela Estudar_tópico
Selecionar topico	Selecionar	Opção	Baixa	Menu	Caixa de seleção simples	Janela Estudar_tópico

Apresentar conteúdo	Apresentar	Conteúdo	Média	Visualizar	Apresentar o conteúdo solicitado	Janela Estudar_conteúdo
Fechar conteúdo	Ativar	Fechar	Baixa	Controle	Botão de Fechar	Janela Estudar_conteúdo

Preenchido o quadro, o projetista pode inferir sobre os dados e, se necessário, prototipar casos de uso de forma a verificar quão adequadas foram as decisões tomadas.

O uso das ferramentas CTTE e TERESA permite modificar esta sistemática. Grande parte das informações apresentadas no Quadro 37 fazem parte do conjunto de dados presentes nos descritores de tarefas do CTTE. Assim, o quadro poderia ser reescrito na forma representada pelo Quadro 38.

Quadro 38 - Modelo da interação da Tarefa Estudar através do CTTE

CTTE												
Identificação da Tarefa	Modelo da Tarefa				Modelo da Interação							
	Frequência	Plataforma	Iterativa	Opcional	Categoria	Tipo	Objetos					
							Nome	Classe	Tipo	Accessibilidade	Cardinalidade	Plataforma
Consultar Tutorial na Web	Alta	Desktop	Sim	Não	Abstrata	-	-	-	-	-	-	-
Estudar	Alta	Desktop	Não	Não	Abstrata	-	-	-	-	-	-	-
Escolher Opcao Estudar	Alta	Desktop	Não	Não	Interação	Controle	Estudar	Texto	Visível	Acesso	-	Desktop
Escolher assunto	Alta	Desktop	Não	Não	Abstrata	-	-	-	-	-	-	-
Apresentar assuntos	Alta	Desktop	Não	Não	Aplicação	Agrupamento	-	-	-	-	-	-
Cancelar1	Alta	Desktop	Não	Não	Interação	Controle	-	-	-	-	-	-
Assunto	Alta	Desktop	Não	Não	Abstrata	-	-	-	-	-	-	-
Selecionar assunto	Alta	Desktop	Não	Não	Interação	Seleção Escolha_Simples	Assuntos	Texto	Visível	-	Alta	Desktop
Cancelar	Alta	Desktop	Não	Não	Interação	Controle	-	-	-	-	-	-
Estudar assunto	Alta	Desktop	Não	Não	Abstrata	-	-	-	-	-	-	-
Selecionar topico	Alta	Desktop	Não	Não	Interação	Seleção Escolha_Simples	Topicos	Texto	Visível	-	Alta	Desktop
Apresentar conteúdo	Alta	Desktop	Não	Não	Aplicação	Visualizar	Conteudo	Texto	Visível	Acesso	-	Desktop
Fechar conteúdo	Alta	Desktop	Não	Não	Interação	Controle	-	-	-	-	-	-

A ferramenta TERESA, a partir destas informações e seguindo uma série de passos descritos a seguir, possibilita gerar um protótipo. Iniciando pelo modelo gerado no CTTE e utilizando TERESA, o projetista deve gerar o *Presentation Tasks Sets* (PTS) que, através da análise das relações entre tarefas, resulta em um conjunto de tarefas por apresentação. Para tanto a ferramenta disponibiliza uma tela (ver Figura 55) na qual:

- apresenta (item 1) o conjunto de PTS inicialmente gerados pela ferramenta (Quadro 39);
- disponibiliza critérios (item 2) que permitem refinar o conjunto PTS inicial:
 - *Joining when Enabling*, segundo o qual, caso dois PTS estiverem no mesmo nível, conectados através do operador *enabling* e diferirem em apenas um elemento, então ocorrerão conjuntamente em uma mesma PTS;
 - *Single Element Sets*, quando uma PTS tiver apenas um elemento, esta será adicionada a outra PTS;
 - *Sharing most elements Sets*, caso as PTSs compartilhem muitos elementos, elas serão unificadas;
 - *Exchanging information*, caso haja um operador de troca de informação entre duas tarefas, elas serão apresentadas na mesma apresentação visando enfatizar tal transferência de informação.
- e os novos PTS gerados são apresentados no quadro (item 3) para que o projetista possa verificar as alterações geradas e avaliar se deseja aproveitar o novo conjunto PTS criado (Quadro 40).

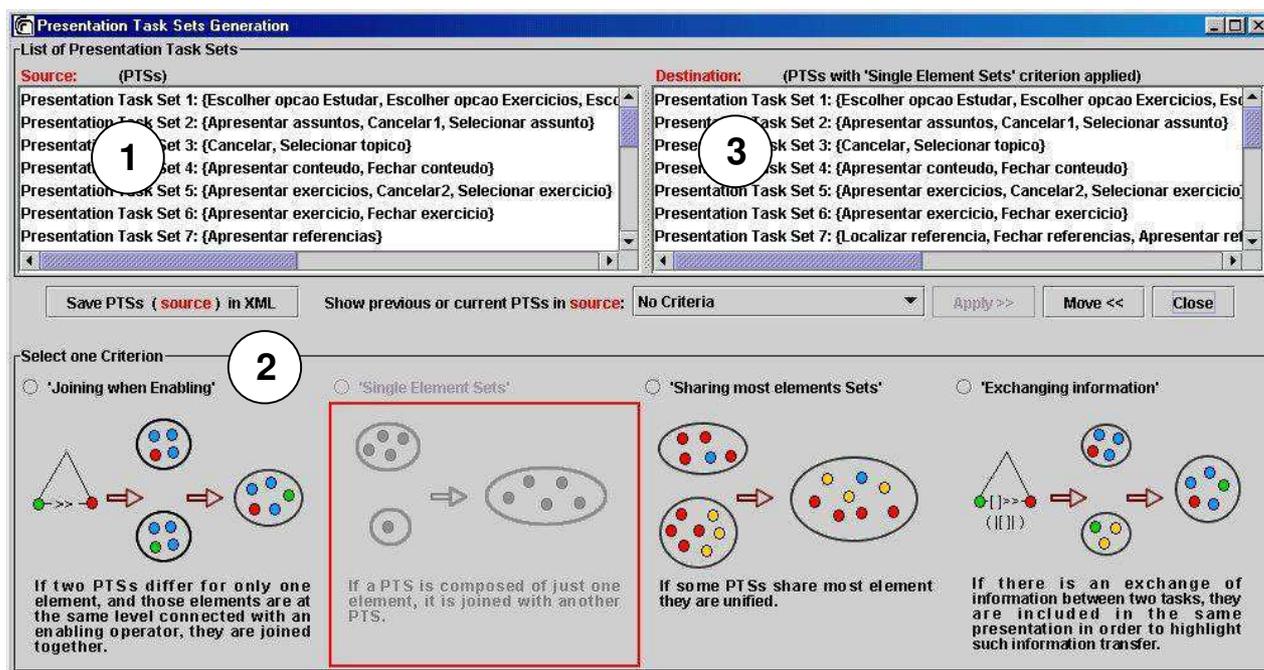


Figura 55 - Tela do *Presentation Task Sets Generation*

Quadro 39 - Conjunto inicial de PTS

Conjunto inicialmente apresentado de PTS
apresentação 1:{Escolher opcao Estudar, Escolher opcao Exercicios, Escolher opcao Referencias, Escolher opcao Sites, Escolher opcao Download, Escolher opcao Ajuda}
apresentação 2:{Apresentar assuntos, Cancelar1, Selecionar assunto}
apresentação 3:{Cancelar, Selecionar topico}
apresentação 4:{Apresentar conteudo, Fechar conteudo}
apresentação n: ...

Quadro 40 - Conjunto PTS após aplicação de critério

PTS apresentado após aplicado o critério <i>Exchanging information</i>
apresentação 1:{Escolher opcao Estudar, Escolher opcao Exercicios, Escolher opcao Referencias, Escolher opcao Sites, Escolher opcao Download, Escolher opcao Ajuda}
apresentação 2:{Apresentar assuntos, Cancelar1, Selecionar assunto, Cancelar, Selecionar topico, Apresentar conteudo, Fechar conteudo}
apresentação n: ...

A partir do PTS gerado a ferramenta permite criar a interface abstrata (AUI). Segundo Paternò et al. [Mori03], a AUI consiste em uma descrição abstrata da interface do usuário. Baseada no PTS, a AUI é especificada através de objetos abstratos de interação. Esta etapa é totalmente realizada pela ferramenta, cabendo ao projetista apenas iniciar o processo. Como resultado final o projetista tem a geração de um arquivo em XML da AUI.

Por fim, tem-se a etapa de geração da interface do usuário (Figura 56). Etapa totalmente dependente da plataforma de execução, pois considera propriedades específicas dos dispositivos [Mori03].

Apesar de Mori et al. descreverem em [Mori03] que a ferramenta utiliza técnicas tais como *presentation patterns* para a elaboração das interfaces concretas (CUI) a partir das AUI (geradas anteriormente), não há nenhuma outra informação que remeta a existência e uso de critérios ergonômicos.

Nesta etapa do processo, a ferramenta disponibiliza ao projetista alguns níveis de contextualização, tais como:

- quanto ao projeto visual da interface, de forma a definir para todas as interfaces geradas propriedades tais como: alinhamento do texto, tipo da fonte, tamanho da fonte, entre outras;
- quanto ao uso de *operators*, esta opção permite definir se elementos, tais como itens de uma lista, serão agrupados por uma caixa, apresentados em colunas ou linhas, entre outras;

- quanto a *interactors*, de forma a definir para todas as interfaces geradas níveis de acesso tais como: o tipo de controle de interação, se os campos de preenchimento serão protegidos (para senhas, por exemplo) ou visíveis, entre outras;
- quanto a cada apresentação, uma vez definidas as informações *default*, pode-se analisar cada apresentação gerada e personalizá-la (indicando o rótulo a ser apresentado, os campos existentes, as cores de fundo e de fonte, entre outras).

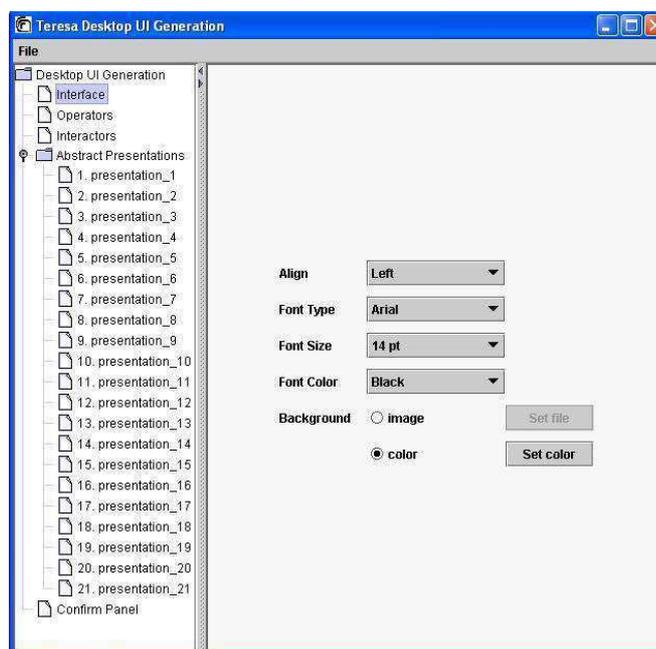


Figura 56 - Geração do Protótipo

Outro recurso disponibilizado nesta etapa é o *Confirm Panel* (Figura 57). Esta tela permite ao projetista observar quais objetos de interação estão associados a cada item de cada apresentação, bem como a forma de agrupamento definida. Pode-se verificar, por exemplo, no conjunto “**1.presentation_1**” (marcado na Figura 57 por uma elipse) que as opções de escolha (Escolher_opcao_Referenci..., Escolher_opcao_Ajuda1, Escolher_opcao_Sites1) serão disponibilizadas através de um *Link* para cada uma e que serão apresentadas como uma lista ordenada na forma de coluna.

Entretanto, apesar da ferramenta fazer uso de objetos de interação, tais como: caixas *drop down*, *list box* ou *radio button* estas opções são pré-definidas pela ferramenta e não é permitido ao projetista alterá-las ou mesmo selecioná-las para outro elemento da interface.

Caso a ferramenta permitisse ao projetista inferir e alterar as informações apresentadas no *Confirm Panel*, seria possível fazer uso de regras ergonômicas através do conhecimento e discernimento do projetista. Outra opção desejável é que a ferramenta permita a inclusão de

novas regras ergonômicas que possam ser aplicadas sobre os modelos, a exemplo da aplicação dos **critérios de refino dos PTS** (citado anteriormente).

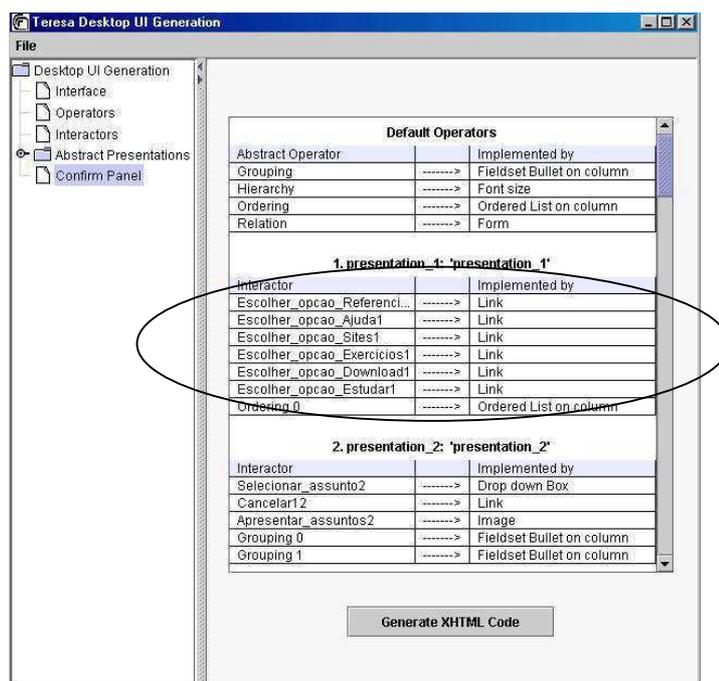


Figura 57 - Confirm Panel

5.3.3 Concepção e Avaliação do Protótipo

5.3.3.1 Geração do Protótipo

Após definidos os níveis de contextualização específicos para cada apresentação disponível, a ferramenta permite gerar o código fonte em formato XHTML⁴⁰, que por sua vez, pode ser facilmente visualizado em um *browser*.

5.3.3.2 Avaliação do Protótipo

Como apresentado no Capítulo 2, no MCI, a avaliação do protótipo baseia-se na tríade:

- inspeção de conformidade a padrões - através de *cheklists* do ISO 9241;
- mensuração do desempenho do usuário - através de **Ensaio de Usabilidade**;
- sondagem da satisfação do usuário - através da ferramenta **OpUS**.

⁴⁰ Utilizado para as plataformas: *Desktop* e telefone celular. Para interfaces vocais a ferramenta gera em formato VXML.

A avaliação do protótipo deve gerar um relatório contendo informações que possibilitem identificar e isolar problemas além de servir como referência para a resolução destes problemas [Turn00]. Portanto, é imprescindível que os suportes computacionais utilizados apoiem tanto a captura dos dados quanto sua compilação. Das opções, atualmente utilizadas, apenas o OpUS já possui algum grau de informatização. As demais avaliações demandam tempo considerável do especialista na compilação dos dados apurados até transformá-los em resultados. O Quadro 41 apresenta uma versão alterada do OpUS para aplicação no estudo de caso.

Quadro 41 - Questionário OpUS

O Produto - Uso e Navegação					
Aspecto	Escala				
	1	2	3	4	5
	Muito fácil	Fácil	Nem fácil Nem difícil	Difícil	Muito difícil
1. Uso do produto na realização de tarefas de interesse					
2. Comunicação com o produto (terminologia, linguagem, retorno da informação e das ações em geral)					
3. Localização dos itens de menu associados às tarefas					
4. Acesso aos itens de menu associados às tarefas					
5. Acesso às instruções e advertências do produto					
6. Compreensão das instruções e advertências do produto					
7. Acesso às janelas de diálogo do produto					
8. Compreensão dos termos e solicitações apresentadas nas janelas de diálogo do produto					
9. Recuperação de situações de erro					
10. Recuperação de situações de travamento					
11. Compreensão das mensagens de erro apresentadas					
12. Navegação através das diferentes opções do menu, janelas de diálogo e barras de ícones do produto					
13. Navegação através de diferentes partes das tarefas executadas pelo produto					
14. Memorização das seqüências de ações associadas a cada tarefa executada com o auxílio do produto					
15. Aprendizado de novas seqüências de ações					
16. Uso das funcionalidades mais comuns do produto					
17. Exploração de novas funcionalidades do produto					
18. Visualização à primeira vista das seqüências de ações necessárias para completar as tarefas de interesse com o auxílio do produto					
19. Visualização à primeira vista da estruturação dos menus, barras de ícones ou listas de informações disponibilizadas pelo produto					
20. Processo de entrada e saída de dados durante o uso do produto					
O Produto - Documentação Online					
Aspecto	Escala				
	1	2	3	4	5
	Muito fácil	Fácil	Nem fácil Nem difícil	Difícil	Muito difícil
21. Localização e acesso aos mecanismos de ajuda online do produto					
22. Uso dos mecanismos de ajuda online do produto					

23. Compreensão das informações de interesse existentes na ajuda online					
24. Linguagem utilizada na ajuda online					
25. Localização das informações de interesse na ajuda online					
26. Uso da ajuda online					
27. Superação de dúvidas e problemas encontrados durante as sessões de uso do produto a partir dos mecanismos oferecidos pela ajuda online					
O Produto - Você e o Produto					
Aspecto	Escala				
	1	2	3	4	5
	Muito fácil	Fácil	Nem fácil Nem difícil	Difícil	Muito difícil
28. Acho que a apresentação do produto é bastante atraente, o que estimula seu uso por mim.					
29. Acho que a apresentação do produto facilita o aprendizado rápido de seus comandos.					
30. Acho o modo de apresentação das mensagens do produto bastante claro e compreensível .					
31. A resposta do produto às minhas entradas é muito lenta.					
32. Sempre me sinto no controle das ações quando uso o produto.					
33. Consigo executar as tarefas de modo direto ao usar o produto.					
34. Acho que o produto atende plenamente às minhas necessidades.					
35. Perco muito tempo tentando aprender os comandos do produto.					
36. Consigo fazer exatamente o que desejo com os recursos oferecidos pelo produto.					
37. As funcionalidades oferecidas pelo produto são facilmente configuráveis de modo a atenderem às minhas necessidades.					
38. O produto apresenta tantas funcionalidades que às vezes acho que nunca aprenderei ou terei necessidade de usá-las todas.					
39. Ao usar o produto, consulto freqüentemente os mecanismos de ajuda oferecidos.					
40. Não acho que as informações da ajuda deste produto são suficientemente eficazes para tirar minhas dúvidas.					
41. Ao consultar a ajuda do produto, sempre perco muito tempo tentando encontrar as informações de interesse.					
42. De um modo geral, a quantidade de informações oferecidas pela ajuda do produto é insuficiente para solucionar meus problemas e dúvidas.					
43. De um modo geral, a qualidade das informações oferecidas pela ajuda do produto não contribui para a solução dos meus problemas e dúvidas.					
44. Sinto-me receoso em alguns momentos das sessões de uso do produto.					
45. Em alguns momentos, sinto-me frustrado com o modo como o produto executa tarefas de meu interesse.					
46. De um modo geral, sinto-me satisfeito ao usar o produto.					
47. Recomendaria sem hesitação o uso do produto aos meus colegas.					

Para a avaliação por inspeção de conformidade, faz-se necessária a informatização dos *checklists* existentes no padrão, utilizando algum recurso semelhante ao OpUS. Já o Ensaio de Usabilidade, como discutido no Capítulo 2 (seção 2.4.3.2), possui várias etapas a maioria delas sem suporte computacional (Quadro 42).

A etapa de **Preparação**, como pode ser observada, utiliza material textual e, portanto, pode se beneficiar de editores de texto.

Para a etapa de **Teste**, utilizam-se meios como a captura de áudio e vídeo dos ensaios e o preenchimento de planilhas com as impressões e observações do(s) especialista(s) envolvido(s).

Quadro 42 - Relação entre Etapas do Ensaio e Materiais utilizados

Etapa	Materiais utilizados
Preparação	- formulários de cunho legal para utilização dos dados apurados junto ao usuário; - instruções das seqüências de tarefas a serem executadas pelos usuários; - planilhas de acompanhamento do ensaio.
Experimento	- captura da interação entre usuário e computador através das ferramentas FAIdos e FAIwin (ver Capítulo 2, seção 2.4.3.2). - captura da interação entre usuário e especialista. - captura das impressões e observações do especialista
Questionamento/ Análise de Resultados	- associação e arquivamento dos dados apurados; - relatório do ensaio; - relatório final

A etapa de **Questionamento/Análise de Resultados** tende a ser extremamente exaustiva para o ESPECIALISTA e sua equipe devido à necessidade de se concentrar os dados apurados e compilá-los, demandando apoio computacional.

5.4 INTEGRAÇÃO

5.4.1 Integração através de um conjunto fixo de ferramentas

As fases iniciais do método (**Análise e Modelagem do Perfil do Usuário** e **Análise e Modelagem do Contexto**), são caracterizadas pela apuração de um volume considerável de informações sobre o usuário. Em termos de integração com as fases seguintes, observa-se a necessidade da automatização da coleta e compilação dos dados levantados. Já as etapas seguintes estão, quase em sua totalidade, supridas pelas ferramentas CTTE e TERESA. A Figura 45 permite observar a abrangência de fases que são apoiadas por estes suportes computacionais. Deve ser ressaltado que entre estas ferramentas não há problemas de integração, uma vez que ambas foram desenvolvidas pelo mesmo grupo de pesquisa com o objetivo de serem utilizadas em conjunto.

A fase de **Avaliação do Modelo de Navegação** possui duas alternativas de suportes computacionais: CADP ou Design/CPN. O CTTE gera um artefato que pode ser diretamente utilizado pela ferramenta CADP, ficando portanto, uma integração direta. O Desing/CPN utiliza uma notação diferente do CADP, assim o artefato gerado pelo CTTE não pode ser

diretamente utilizado. Uma forma de realizar a integração seria trabalhar com os arquivos XML (que tanto o CTTE quanto o Design/CPN aceitam). Entretanto, os formatos dos arquivos são diferentes, exigindo um conversor entre as ferramentas.

Outro ponto que permanece desfavorável ao uso do Design/CPN é que mesmo havendo um conversor, o projetista ainda teria de compreender o funcionamento da ferramenta para efetuar alterações no modelo obtido, para só então realizar as atividades de análise.

Uma vez gerado o protótipo cabe avaliá-lo e a ferramenta disponível é o OpUS. Por ser uma etapa de questionamentos, novamente pode haver um grande volume de dados que precisam ser compilados de forma a permitir ao projetista avaliar as decisões tomadas. A integração que deve haver destas informações com as etapas anteriores consiste na re-alimentação dos modelos com os resultados obtidos pela avaliação. Neste caso, o meio de integração é o próprio projetista, que após inferir sobre os resultados compilados decide por quais alterações devem ser efetivamente realizadas. Cabe ressaltar que a ferramenta OpUS aborda apenas a avaliação pela sondagem da satisfação do usuário quanto ao produto. As demais formas de avaliação são discutidas em tópico posterior.

5.4.2 Integração através de um canal de comunicação XIML

Outra forma de compor um suporte computacional ao MCI parte do princípio de uma “padronização” do meio de integração. Ou seja, ao invés de serem definidas ferramentas específicas para cada etapa do processo, define-se um canal único de comunicação entre ferramentas e o método. Há grandes dificuldades nesta opção, principalmente quanto a escolha do canal de comunicação que será adotado. Entretanto há um grande benefício que é justamente tornar o MCI independente de ferramentas específicas.

Como já foi discutido, o padrão XIML, é o meio de integração recomendado por permitir maior interoperabilidade entre programas, além de abranger todos os modelos do processo de concepção de interfaces. Infelizmente o XIML exige um cadastro no grupo de pesquisa XIML.org, o que respalda a manutenção do padrão, porém dificulta a obtenção de informações essenciais ao seu uso a exemplo dos *tags* utilizados pela linguagem.

Atualmente, a maioria das ferramentas adota algum padrão de persistência que é regido pelo formato XML. Entretanto, em virtude da busca por um padrão genérico, XML permite que o grupo de desenvolvimento crie seu próprio conjunto de *tags*. Os grupos, tal

como o XIML.org, objetivam a padronização de ontologias, de forma a facilitar a comunicação entre ferramentas.

Uma vez que as ferramentas ainda não possuem suporte nativo XIML, faz-se necessária a utilização de um conversor que permita à ferramenta interagir com o MCI (Figura 58).

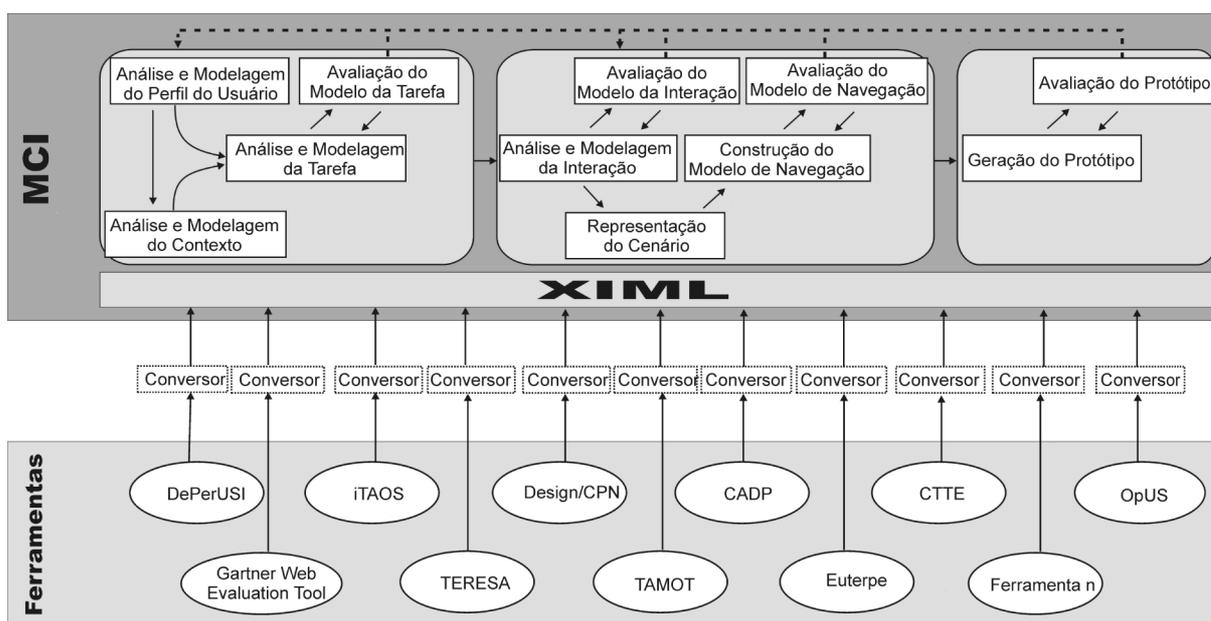


Figura 58 - MCI com XIML

O Quadro 43 apresenta um extrato do código do modelo da tarefa do Estudo de Caso gerado através do iTAOS e do CTTE. Observa-se nos trechos de códigos marcados com elipses o uso de ontologias bastante diferentes para representar um mesmo modelo. Para identificar a tarefa raiz “**Consultar Tutorial na Web**”, o iTAOS utiliza as *tags* “task number” e “name”, enquanto que o CTTE utiliza “Task Identifier”. Já para a tarefa folha “**Escolher opcao Estudar**”, no CTTE é representada pela abertura de uma *tag* “SubTask” seguida da *tag* “Task Identifier”, enquanto que no iTAOS é representada pelas *tags* “action number” e “name”.

Portanto, o uso de um **conversor XML-XIML** permitiria que ambas as ferramentas pudessem exportar seus artefatos de forma que o MCI pudesse utilizá-los. Observa-se, portanto, que para o MCI o que importa é a geração dos artefatos e a melhor forma de fazer isto é através de um canal de comunicação que permita o uso de um número ilimitado de ferramentas.

Quadro 43 - Código XML iTAOS x CTTE

iTAOS	CTTE
<pre> <model name="" description=""> <taskTree> <task number="1" name="Consultar Tutorial na Web" description="" duration="0" frequency="" importance="" interruptability="" modality="" occurrence="(0,1)" type="" priority="0"> <preSituation/> <postSituation/> <howToPerform> <reference name="Method task Tutorial" /> <howToPerform/> <task number="1.1" name="Estudar" description="" duration="0" frequency="" importance="" interruptability="" modality="" occurrence="(0,1)" type="" priority="0"> <preSituation/> <postSituation/> <howToPerform> <reference name="Method task Estudar" /> <howToPerform/> <action number="1.1.1" name="Escolher opcao Estudar" description="" duration="0" frequency="High" importance="High" interruptability="No interruptible" modality="Manual" occurrence="(0,1)" type="Mental" priority="0" status=""> <preSituation/> <postSituation/> <actionAgents/> <actionTools/> </action> ... </methods> ... <method name="Method task Estudar" description="" instant="0" body="SEQ(Escolher opcao Estudar,Escolher assunto)" /> ... <method name="Method task Tutorial" description="" instant="0" body="XOR(Estudar,Fazer Exercicios,Ver Referencias,Visitar Sites,Fazer Downloads,Solicitar Ajuda)" /> </methods> </model> </pre>	<pre> <TaskModel NameTaskModelID="D:\Warden\Modelos da tarefa gerados_Modelo - tutorial Web - CTTE - 8d.xml"> <Task Identifier="Consultar Tutorial na Web" Category="abstraction" Iterative="true" Optional="false" PartOfCooperation="false" Frequency="null"> <Name>null</Name> <Type /> <Description>null</Description> <Precondition>null</Precondition> <TimePerformance> <Max>null</Max> <Min>null</Min> <Average>null</Average> </TimePerformance> <Object name="null" class="null" type="null" access_mode="null" cardinality="null"> <Platform>null</Platform> <InputAction Description="null" From="null" /> <OutputAction Description="null" To="null" /> </Object> <SubTask> <Task Identifier="Estudar" Category="abstraction" Iterative="false" Optional="false" PartOfCooperation="false" Frequency="null"> <Name>null</Name> <Type /> <Description>null</Description> <Precondition>null</Precondition> <TemporalOperator name="Choice" /> <TimePerformance> <Max>null</Max> <Min>null</Min> <Average>null</Average> </TimePerformance> <Parent name="Consultar Tutorial na Web" /> <SiblingRight name="Fazer Exercicios" /> <Object name="null" class="null" type="null" access_mode="null" cardinality="null"> <Platform>null</Platform> <InputAction Description="null" From="null" /> <OutputAction Description="null" To="null" /> </Object> </SubTask> <SubTask> <Task Identifier="Escolher opcao Estudar" Category="interaction" </pre>

```

Iterative="false" Optional="false" PartOfCooperation="false"
Frequency="High">
  <Name />
  <Type>Control</Type>
  <Description />
  <Platform>Desktop</Platform>
  <Precondition />
  <TemporalOperator name="SequentialEnabling" />
  <TimePerformance>
    <Max />
    <Min />
    <Average />
  </TimePerformance>
  <Parent name="Estudar" />
  <SiblingRight name="Escolher assunto" />
  <Object name="Estudar" class="Text" type="Perceivable"
  access_mode="Access" cardinality="null">
    <Platform>Desktop</Platform>
    <Platform>null</Platform>
  </Object>
</Task>

```

...

Capítulo 6 - Conclusões

São poucas as equipes de desenvolvimento que possuem profissionais capacitados na concepção de interfaces considerando o aspecto usabilidade. É indiscutível a necessidade de se adotar um método que guie/auxilie a equipe no desenvolvimento das interfaces levando em conta este critério. Ao longo deste trabalho foi constatado na literatura que a maioria dos métodos existentes exige o conhecimento de formalismos específicos além do conhecimento sobre o fluxo do método.

Uma solução para este problema, proposto neste trabalho, é a disponibilização de ferramentas ou suportes computacionais de modo a auxiliar e sistematizar o desenvolvimento das interfaces, atribuindo-lhes como consequência, um maior grau de usabilidade. Entretanto, é necessário esclarecer que a vinculação de suportes computacionais às etapas de um método não elimina a necessidade de um profissional qualificado em interface na equipe de desenvolvimento, porém facilita e agiliza o seu trabalho e o da equipe de desenvolvimento.

Neste trabalho, adota-se o princípio de que o uso de um método, amparado por suporte computacional, reduz o tempo de aprendizagem, bem como estabelece práticas mais eficazes no processo de desenvolvimento. Assim, ao reduzir o tempo de desenvolvimento da interface, surge no cronograma de desenvolvimento o tempo para outros aspectos do projeto.

Neste capítulo discutiremos os resultados obtidos da aplicação da abordagem de desenvolvimento de interfaces, proposta neste trabalho (a adoção de suportes computacionais no processo de desenvolvimento de interfaces, a partir de ambiente de integração), suas contribuições e as propostas de continuidade.

6.1 DISCUSSÃO DOS RESULTADOS

No processo de apuração dos suportes computacionais constatou-se a importância dos formalismos para a modelagem da tarefa e a grande diversidade de ferramentas e de formalismos disponíveis para realizar esta etapa de concepção de interfaces. Neste estudo, foi constatado que apesar da diversidade de nomenclaturas adotadas para a definição de tarefas e

subtarefas e que apesar de existir uma grande variação no número de operadores adotados nos vários formalismos estudados, há uma grande semelhança entre os operadores temporais.

Embora inicialmente este estudo propusesse associar suportes temporais computacionais aos formalismos já adotados no método MCI, verificou-se que dada a indisponibilidade de suportes para os formalismos adotados em algumas etapas do MCI, seria necessário propor uma mudança para formalismos compatíveis. Verificou-se, por exemplo no caso da etapa de construção do modelo da tarefa, que a mudança não acarretou prejuízo para aplicação do método, uma vez que, o formalismo originalmente adotado – MAD, era simples, e toda a semântica adotada já está contida no novo formalismo – CTT.

Neste capítulo, o tópico 6.1.1 apresenta considerações gerais acerca do resultado da pesquisa por suportes computacionais. No tópico seguinte, 6.1.2, são tecidas considerações sobre as ferramentas escolhidas para compor o suporte computacional proposto. No tópico 6.1.3 são apresentadas as limitações dos resultados alcançados nesta pesquisa. Por fim são apresentadas as considerações finais e as propostas de continuidade para este trabalho.

6.1.1 Suportes Computacionais

Ao longo deste trabalho foram estudados dezessete suportes computacionais, dentre os quais: seis ambientes e onze ferramentas. O Quadro 44 apresenta um resumo do estudo realizado. Pode-se observar que dos ambientes acadêmicos, apenas o CTTE+PetShop está parcialmente disponível e que quanto às notações associadas, apenas Teallach e CTTE+PetShop apresentam notações mais específicas.

Quadro 44 - Ambientes

	JMVCase	DIGBE	INGRID	Teallach	CTTE+PetShop	TeleUSE
Ambiente acadêmico	X	X	X	X	X	-
Ambiente comercial	-	-	-	-	-	X
Notação associada	X ₁	?	?	X ₂	X ₃	X ₁
Disponível para uso	-	-	-	-	X ₄	X

- : não.
- ?: informação indisponível.
- x: plenamente aplicado.
- x₁: orientação a objetos.
- x₂: ODMG e Modelos hierárquicos para tarefa.
- x₃: CTT e ICO.
- x₄: parcialmente disponível.

Por fim, através do Quadro 15 (ver página 51) pode-se observar a relação entre as etapas do MCI e os suportes oferecidos pelos ambientes, concluindo-se que a grande maioria dos suportes computacionais apóia apenas parcialmente as etapas do método.

Nenhum dos ambientes estudados contempla todas as etapas do método, exigindo, portanto, a utilização de ferramentas complementares. Desta forma, torna-se inviável a adoção de apenas um ambiente como suporte ao método MCI.

O Quadro 45 apresenta um resumo das principais informações sobre as onze ferramentas estudadas. Pode-se observar que a grande maioria está pública e livremente disponível, a exceção das ferramentas: Euterpe e *Gartner Web Evaluation Tool*. Três ferramentas são de autoria do GIHM e dentre onze ferramentas analisadas, sete podem ser utilizadas em multiplataformas.

Quadro 45 - Ferramentas

MCI	Euterpe	TAMOT	iTAOS	CTTE	DENIM	Gartner Web Evaluation Tool	Design/CPN	OpUS	DePerUSI	TERESA	CADP
Autoria do GIHM	-	-	X	-	-	-	-	X	X	-	-
Multiplataforma	-	-	X	X	X	-	-	X ₁	X ₁	X	X
XML como entrada	-	X	-	X	-	-	X	-	-	X	?
XML como saída	-	X	X	X	-	-	X	-	-	X	?
Disponível para ser utilizada	X ₂	X	X	X	X	-	X	X	X	X	X

- : não aplicável.
- ?: informação indisponível.
- x: plenamente aplicado.
- x₁: somente a versão para Web.
- x₂: disponível apenas para fins não comerciais.

O Quadro 28 (apresentada na página 76) relaciona as etapas do MCI com as ferramentas analisadas e diferentemente do que ocorre com os ambientes, o suporte computacional ao MCI é possível, quando realizado através de ferramentas.

6.1.2 Suporte Computacional Proposto

Houve a apresentação de duas abordagens de suporte computacional, (I) uma com o MCI em função de um conjunto de ferramentas e (II) uma segunda onde é indicada a inclusão de um padrão de integração ao MCI.

O Estudo de Caso apresentado neste trabalho foi aplicado sobre a primeira proposta de suporte (ver Figura 45 na página 91), sendo possível avaliar a eficácia do suporte proposto. Apesar dos avanços obtidos, há alguns pontos que devem ser observados, tais como etapas sem ferramental e sugestões para o ferramental indicado na proposta.

6.1.2.1 Etapas não contempladas por ferramental

As etapas iniciais de **Análise e Modelagem do Perfil do Usuário** e **Análise e Modelagem do Contexto** são baseadas na aplicação de questionários (informatizados ou não), sendo que a segunda não possui ferramental diretamente relacionado. Observou-se que apesar desta limitação, as informações desta etapa podem ser inferidas através do perfil do usuário, bastando para isto, a adição de novas questões à ferramenta existente.

A etapa de **Avaliação do Protótipo**, como apresentada no Capítulo 2 pode ser feita através de três formas distintas:

- inspeção de conformidade a padrões;
- mensuração do desempenho do usuário durante o uso do produto;
- sondagem da satisfação do usuário quanto ao produto.

Desta forma, na proposta de suporte apresentada, apenas a **sondagem da satisfação do usuário quanto ao produto** é contemplada por ferramental. Para as duas outras formas, sugere-se:

- inspeção de conformidade a padrões:
 - informatização dos *checklists*, a exemplo do que já ocorre com o ErgoList⁴¹.
- mensuração do desempenho do usuário durante o uso do produto:
 - a atualização das ferramentas FAIwin e FAIdos (pertencentes ao GIHM porém atualmente em desuso).

6.1.2.2 Ferramental indicado na proposta

Apesar de haverem avanços, principalmente na integração e transição entre etapas do método de concepção, há pontos que poderiam ser aprimorados nas ferramentas indicadas no suporte proposto.

A começar pelas ferramentas de autoria do GIHM: DePerUSI e OpUS, para as etapas de **Análise e Modelagem do Perfil do Usuário** e **Avaliação do Protótipo**, respectivamente,

⁴¹ Projeto resultante da colaboração entre SoftPólis, núcleo Softtex-2000 de Florianópolis e LabIUtil (<http://www.labiutil.inf.ufsc.br/ergolist/>).

sugere-se que as ferramentas apoiem a elaboração dos questionários utilizados, bem como que permitam o tratamento das informações coletadas.

Para a etapa de Representação do Cenário, como apontado no Capítulo 2, o MCI utiliza *Use Case* e *MSC*. Das ferramentas estudadas neste trabalho, apenas o CTTE disponibiliza meios de representação de cenários, apesar de não ser a mesma forma de representação utilizada pelo MCI.

Entretanto, cabe ressaltar que, apesar de não operacional, o CTTE possui um editor de *Use Case*. Portanto, sugere-se que este editor seja posto em funcionamento e que, além da atual forma de representação de cenários, a ferramenta também permita o uso de *Use Case*.

Quanto às etapas de **Análise e Modelagem da Interação** e **Avaliação do Modelo da Interação**, indica-se o uso da ferramenta TERESA que através de um conjunto fixo de objetos de interação os quais estão diretamente vinculados a alguns campos dos descritores da tarefa do CTTE possibilita automatizar a transição entre o modelo da tarefa e da interação além da geração automática do protótipo.

A definição destes objetos de interação é baseada em um conjunto de regras definidas pelos autores da ferramenta como *presentation patterns*. Apesar do CTTE permitir a inclusão de novos tipos, estes não estão vinculados a nenhuma destas regras da ferramenta TERESA, o que gera erro quando o modelo gerado no CTTE é executado no TERESA.

TERESA não permite acesso a este conjunto de regras, assim como não permite sua alteração ou mesmo a inclusão de novas regras ergonômicas.

Sugere-se que a ferramenta permita, ao projetista, acessar o conjunto de regras existente, de forma a possibilitar avaliar a validade destas frente ao domínio da aplicação em desenvolvimento. Além disto, sugere-se que haja maior liberdade para que se modifique e/ou altere o conjunto de regras utilizadas pela ferramenta TERESA.

Desta forma, a ferramenta fornecerá maior grau de mobilidade à equipe e conseqüentemente a obtenção de uma interface mais próxima da desejada.

6.1.3 Escopo da Solução Proposta

A padronização de ontologias XML é uma área de pesquisa relativamente recente e ainda em curso. Dado o escopo deste trabalho, ao considerar o prazo disponível, o estudo se limitou a propor um padrão de integração para as ferramentas escolhidas como suporte não tendo sido possível realizar a integração das ferramentas em um ambiente de concepção.

Neste trabalho foram abordados os padrões de integração mais comumente utilizados e diretamente relacionados com a área de concepção de interfaces. Deste conjunto, constatou-se que o padrão XIML é o mais indicado para ser adotado pelo MCI. Apesar de não possuir ferramentas que permitam gerar protótipos a partir das descrições obtidas e de não disponibilizar facilmente o conjunto de *tags* associadas, o XIML consiste na opção mais indicada, uma vez que é aberto e por abranger um conjunto amplo de modelos dos métodos de concepção de interfaces (ex.: modelos da tarefa e domínio e usuário).

Como apresentada anteriormente, a proposta de suporte computacional através de um padrão de integração exige o desenvolvimento de conversores de forma a permitir a troca de informações entre ferramentas e o método MCI. Para tanto, é necessária a análise das *tags* do padrão XIML visando comparar, e posteriormente relacionar, os elementos presentes no padrão com os elementos dos artefatos do MCI, definindo assim o conjunto de termos XIML-MCI. Em seguida, será necessário o estudo das ontologias utilizadas em cada ferramenta objetivando relacionar os termos de cada ontologia com os termos XIML-MCI. A partir deste conjunto de termos relacionados, pode-se criar os conversores necessários ao suporte computacional através do padrão de integração.

6.2 CONSIDERAÇÕES FINAIS

Apesar das dificuldades encontradas, principalmente quanto a disponibilidade das ferramentas e dos problemas com o acesso e a utilização da documentação de ajuda, foi proposta a composição de um conjunto de ferramentas associadas ao MCI e avaliou-se sua eficácia frente a um Estudo de Caso.

Os resultados obtidos, a partir do Estudo de Caso, são bastante animadores, principalmente quanto a redução da carga de trabalho manual necessária na confecção de modelos, tais como da tarefa e da interação.

Outro resultado deste trabalho foi a proposição de um padrão de integração – XIML, que permitirá concentrar esforços na “padronização do MCI”.

Por fim, com a proposição de um suporte computacional gratuito ao MCI, disponibiliza-se, não só para a comunidade acadêmica, mas também para usos comerciais, um método de concepção de interfaces centrado no usuário que pode vir a ser amplamente utilizado, agora de forma mais clara e simplificada, para o desenvolvimento de interfaces com maior grau de usabilidade.

6.3 PROPOSTAS DE CONTINUIDADE

Diante dos limites impostos pela implementação da solução proposta, a seguir são apresentadas propostas de continuidade:

- A partir do conjunto de *tags* do padrão, **construir conversores XIML para a integração** das ferramentas e aprofundar os estudos de forma a viabilizar o canal de integração, ampliando o conjunto de ferramentas que podem ser associadas ao MCI;
- **aplicar o suporte computacional proposto em outros estudos de caso;**
- **estudar a viabilidade de utilizar o modelo da tarefa gerado no CTTE como entrada para a etapa de modelagem da navegação realizada na ferramenta Design/CPN.** Uma vez que a ferramenta Design/CPN já vem sendo adotada na etapa de modelagem da navegação, sugere-se que seja realizado um estudo que viabilize a criação do modelo de navegação utilizado pelo GIHM em CPN a partir do modelo da tarefa gerado pelo CTTE;
- **ampliar o estudo das ferramentas e ambientes apresentados no Anexo A - Outros Suportes,** de forma a ampliar o conjunto de ferramentas que pode ser associado ao método MCI.

Referências Bibliográficas

- [Abra99] Abrams, M., Phanouriou, C.. *UIML: An XML Language for Building Device-Independent User Interfaces*. In: XML'99. Philadelphia. USA. Dezembro de 1999. Disponível em outubro de 2003 em www.harmonia.com/resources/xml99Final.pdf.
- [Aoni99] Aonix Corporation. *Product Overview - TeleUSE for Windows*. Disponível em dezembro de 2003 em <http://www.aonix.com/pdf/2141-AON.pdf>.
- [Balb99] Balbo, S., Paris, C., Ozkan, N.. *Characterizing Task Formalisms: Towards a Taxonomy*. In: 13th European Conference on Object-Oriented Programming - Workshop on Integrating Human Factors into Use Cases and OO Methods. Lisboa, Portugal. Junho de 1999. Disponível em novembro de 2003 em hci.cs.concordia.ca/www/ecoop99/balbo.doc.
- [Barc99] Barclay, P. J., Griffiths, T., McKirdy, J., Paton, N. W., Cooper, R., Kennedy, J.. *The Teallach Tool: Using Models for Flexible User Interface Design*. In proceedings of 3rd International Conference on Computer-Aided Design of User Interfaces (CADUI'99), Louvain-la-Neuve (Belgium), 21-23 October 1999.
- [Baro02] Baron, M. e Girard, P.. *SUIDT: A task model based GUI-Builder*. In Proceedings of TAMODIA: Task MOdels and DIagrams for user interface design, pp. 64-71. Bucharest, Romania, 18-19 July 2002.
- [Barr99] Barrére, T. da S., do Prado, A. F., Bonafe, V. C.. *CASE com Múltiplas Visões de Requisitos e Implementação Automática*. Anais do XIII Simpósio Brasileiro de Engenharia de Software. Florianópolis, SC. Outubro de 1999.
- [Beau00] Beaudouin-Lafon, M., Mackay, W. E., Andersen, P., Janecek, P., Jensen, M., Lassen, M., Lund, K., Mortensen, K., Munck, S., Ratzner, A., Ravn, K., Christensen, S., Jensen, K.. *CPN/Tools: A Post-WIMP Interface for Editing and Simulating Coloured Petri Nets*. In Petri Nets 2000, 21st International Conference on Application and Theory of Petri Nets, pp. 19-28. Aarhus, Denmark. Junho de 2000. Disponível em junho de 2003 em <http://www.daimi.au.dk/~mackay/pdf/files/PetriNets2000.CPNTools.pdf>.

- [Bent01] Bentley, T., Johnston, L.. *Analysis of Task Model Requirements*. In: OzCHI 2001. Fremantle, Australia. Novembro de 2001.
- [Boda94] Bodart, F., Hennebert, A.-M., Leheureux, J.-M., Vanderdonckt, J.. *Towards a Dynamic Strategy for Computer-Aided Visual Placement*. in Proc. of 2nd ACM Workshop on Advanced Visual Interfaces AVI'94 (Bari, 1-4 juin 1994), T. Catarci, M.F. Costabile, S. Levialdi & G. Santucci (éds.), ACM Press, pp. 78-87. New York, 1994.
- [Boda95] Bodart, F., Hennebert, A.-M., Leheureux, J.-M., Provot, I., Sacre, B., Vanderdonckt, J.. *Towards a Systematic Building of Software Architectures: the Trident methodological guide*. in Proc. of 2nd Eurographics Workshop on Design, Specification, Verification of Interactive Systems DSV-IS'95 (Toulouse, 7-9 juin 1995), Ph. Palanque & R. Bastide (éds.), Springer-Verlag, pp. 262-278. Vienne, 1995.
- [Boja03] Bojanic, P.. *The Joy of XUL*. The Mozilla Organization. Disponível em novembro de 2003 em www.mozilla.org/projects/xul/joy-of-xul.html.
- [Bolo87] Bolognesi, T. and Brinksma, Ed.. *Introduction to the ISO Specification Language LOTOS*. In: Computer Networks and ISDN Systems. Janeiro de 1987. Disponível em novembro de 2003 em www2.cs.uregina.ca/~sadaouis/CS872/bolognesi_brinksma.pdf
- [Carr90] Carrico, L., Guimarães, N., Antunes, P.. *INGRID: A Graphical Tool for User Interface Construction*. Proceedings of the EUUG'90 Conference. Munich, Germany. 1990. Disponível em junho de 2003 em www.di.fc.ul.pt/~paa/papers/euug-90.pdf.
- [Coli02] Colineau, N., Paris, C., Linden, K. V.. *An Evaluation of Procedural Instructional Text*. In the Proceedings of the International Natural Language Generation Conference (INLG) 2002, pp. 128-135. New York, USA. Julho de 2002.
- [Cord03] Cordeiro, P. B.. *Projeto e Implementação do Módulo TAME da Ferramenta iTAOS para Análise e Modelagem da Tarefa*. Dissertação de Mestrado em Informática. UFCG. Campina Grande, Brasil. Fevereiro de 2003.
- [Cout87] Coutaz, J.. *The Construction of User Interfaces and the Object Paradigm*. The European Conference on Object Oriented Programming, ECOOP'87. Paris, França. 1987.
- [Cout90] Coutaz, J.. *Interfaces Homme-Ordinateur - Conception et Réalisation*. Bordas, Paris. 1990.

- [Cove03] Cover Pages. *XML Markup Languages for User Interface Definition*. In: Technology Reports. Disponível em agosto de 2003 em <http://xml.coverpages.org/userInterfaceXML.html>.
- [Cybi96] Cybis, W. A.. *Ergonomia e Usabilidade de Software - Abordagem Ergonômica para IHC*. Florianópolis, 1996.
- [Cybi98] Cybis, W. de A., Pimenta, M. S., Silveira, M. C., Gamez, L.. *Uma Abordagem Ergonômica para o Desenvolvimento de Sistemas Interativos*. In: Anais I Simpósio Brasileiro de Interação entre Seres Humanos e Sistemas Computacionais (IHC98). Maringá - Paraná - Brasil. Outubro de 1998.
- [Desi02] Design/CPN. *Design/CPN Online Manuals*. University of Aarhus, Denmark. Agosto de 2002. Disponível em junho de 2003 em <http://www.daimi.au.dk/designCPN/>.
- [Elna99] Elnaffar, S. S. and Graham, T. C. N.. *Semi-Automated Linking of User Interface Design Artifacts*. In Proceedings of Computer-Aided Design of User Interfaces (CADUI'99), Kluwer Academic Press, 1999. Disponível em junho de 2003 em <http://www.cs.queensu.ca/home/elnaffar/Publications/adligo.pdf>.
- [Gara02] Garavel, H., Lang, F., Mateescu, R.. *An Overview of CADP2001*. In European Association for Software Science and Technology (EASST). Newsletter volume 4, pp 13-24. Agosto de 2002. Disponível em dezembro de 2003 em <http://www.inrialpes.fr/vasy/Publications/Garavel-Lang-Mateescu-01.html>.
- [Gart01] Gartner Group. *About the Gartner Web Evaluation Tool*. Fevereiro de 2001.
- [Godoy00] Godoy, J.. *Introdução à linguagem SGML*. LIE-BR. Outubro de 2000. Disponível em junho de 2003 em <http://lie-br.conectiva.com.br/godoy/sgml.html#toc1>.
- [Grif01] Griffiths, T., Barclay, P. J., Paton, N. W., McKirdy, J., Kennedy, J., Gray, P. D., Cooper, R., Goble, C. A., da Silva, P. P.. *Teallach: a model-based user interface development environment for object databases*. In: Proc. User Interfaces to Data Intensive Systems (UIDIS), IEEE pres. pp. 86-96. 1999. Disponível em dezembro de 2002 em <http://citeseer.nj.nec.com/article/griffiths99teallach.html>.
- [Grif99] Griffiths, T., Barclay, P. J., McKirdy, J., Paton, N. W., Gray, P. D., Kennedy, J., Cooper, R., Goble, C. A., West, A., Smyth, M.. *Teallach: A Model-Based User Interface Development Environment for Object Databases*. UIDIS'99: pp. 86-96, 1999.

- [Grou03] Group for User Interface Research. *DENIM Documentation*. Universidade da Califórnia. Junho de 2003. Disponível em junho de 2003 em http://guir.berkeley.edu/projects/denim/docs/denim_help.pdf.
- [Guer02] Guerrero, C. V. S.. *MEDITE - Uma Metodologia Orientada a Modelos para Concepção de Interfaces Ergonômicas*. Dissertação de Mestrado em Informática. UFCG. Campina Grande, PB. Fevereiro de 2002.
- [Guim91] Guimarães, N., Carrico, L., Antunes, P.. *Ingrid: A Object Oriented Interface Builder*. T. Korson, V. Vaishnavi, B. Meyer, Eds. Fifth International Conference on the Technology of Object-Oriented Language and Systems, Proceedings of the Tools'91 USA Conference. Pag. 291-300. Sta Barbara, USA. Julho de 1990. Disponível em junho de 2003 em www.di.fc.ul.pt/~paa/papers/tools-91.pdf.
- [Hamm95] Hammouche, H.. *De la modélisation des tâches utilisateurs au prototype de l'interface homme-machine*. Tese de Doutorado. Université Paris VI. Dezembro de 1995.
- [ISO98] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, *ISO 9241 Ergonomic requirements for office work with visual display terminals*. Geneva, 1998.
- [Jens92] Jensen, K. *Coulored Petri Nets - Basic Concepts, Analysis Methods and Pratical Use - Vol. I*. E. Spring-Verlag. USA. 1992.
- [King89] King, R. and Novak, M.. *FaceKit: A Database Interface Design Toolkit*. Proceedings of the Fifteenth International Conference on Very Large Data Bases, pp. 115-123. Amsterdam, The Netherlands. Agosto de 1989. Disponível e março de 2003 em <http://www.vldb.org/dblp/db/conf/vldb/KingN89.html>.
- [Land95] Landay, J. A., and Myers, B.. *Interactive Sketching for the Early Stages of User Interface Design*. In Proceedings of CHI '95: Human Factors in Computing Systems, Denver, CO, May 1995, pp. 43-50. Disponível em junho de 2003 em <http://www.cs.berkeley.edu>.
- [Land96] Landay, J. A.. *Interactive Sketching for the Early Stages of User Interface Design*. Tese de Doutorado. Carnegie Mellon University. Pittsburgh. USA. Dezembro de 1996. Disponível em junho de 2003 em <http://reports-archive.adm.cs.cmu.edu/hcii1996.html>.

- [Laue03] Lauesen, S.. *Task Descriptions as Functional Requirements*. IEEE Software 2003. pp. 58-65. Março/Abril de 2003. Disponível em www.itu.dk/people/slauesen/Papers/IEEEtasks.pdf.
- [Lin01] Lin, L., Newman, M. W., Hong, J. I., Landay, J. A.. *DENIM: An Informal Tool for Early Stage Web Site Design*. Video poster in Extended Abstracts of Human Factors in Computing Systems: CHI 2001, Seattle, WA, March 31-April 5, 2001, pp. 205-206. Disponível em junho de 2003 em <http://guir.berkeley.edu/projects/denim/pubs/denim-chi-2001-video.pdf>.
- [Lin02a] Lin, J., Thomsen, M., Landay, J. A.. *A Visual Language for Sketching Large and Complex Interactive Designs*. In CHI 2002. Minneapolis, Minnesota, USA. Abril de 2002.
- [Lin02b] Lin, J.. *Damask: A Tool for Early-Stage Design and Prototyping of Multi-Device User Interface*. Proposta de Dissertação. Universidade da Califórnia em Berkeley. 2002. Disponível em dezembro de 2002 em <http://guir.cs.berkeley.edu/projects/damask/proposal.pdf>.
- [Lula92] Lula, B. Jr.. *Elaboration d'un Environnement de Génération Interactived' 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.
- [Mark92] Markopoulos, P., Pycock, J., Wilson, S., Johnson, P.. *ADEPT - A task based design environment*. 25th Hawaii International Conference on System Sciences, Conference Proceedings, Vol. II, IEEE Computer Society Press (California), pp. 587-596, 1992.
- [Mark94] Markopoulos, P., Wilson, S., Johnson, P.. *Representation and use of task knowledge in a user interface design environment*. In: IEE Proceedings - Computers and Digital Techniques. 1994. Disponível em novembro de 2003 através do IEEE Xplore.
- [Mark97] Markopoulos, P., Gikas, S.. *Towards a Formal Model for Extant Task Knowledge Representation*. Cognitive Systems, Vol. 3-4, pp. 289-310. 1997
- [Mede03] de Medeiros, F. P. A.. *Projeto e implementação do módulo TAOS-Graph da ferramenta iTAOS para análise e modelagem da tarefa*. Dissertação de Mestrado em Informática. UFCG. Campina Grande, Brasil. Fevereiro de 2003.

- [Mede95] Medeiros, H.. *L'Utilisation d'un Langage Terminologique dans la Modélisation de Concepts Dynamiques*. Journées 1995. Project ACNOS, LRPS-BETA-LAG-INRIA, pp 04-05. Strasbourg, França. Setembro de 1995.
- [Mori02] Mori, G., Paternò, F., Santoro, C.. *CTTE: Support for Developing and Analyzing Task Models for Interactive System Design*. IEEE Transactions On Software Engineering, Vol. 28, No 9, pp. 797-813. Setembro de 2002.
- [Mori03] Mori, G., Paterno, F., Santoro, C.. *Tool Support for Designing Nomadic Applications*. In: Proceedings of IUI 2003. Miami, Florida. Janeiro de 2003. Disponível em junho de 2003 em <http://giove.cnuce.cnr.it/teresa/publications.html>.
- [Mura89] Murata, T. *Petri Nets: Properties, Analysis and Applications*. In: Proceedings of the IEEE, número 77 (4). pp. 541-580. Abril de 1989.
- [Myer96] Myers, B. A.. *A Brief History of Human Computer Interaction Technology*. Carnegie Mellon University School of Computer Science Technical Report CMU-CS-96-163 and Human Computer Interaction Institute Technical Report CMU-HCII-96-103. Dezembro de 1996.
- [Nava01] Navarre, D., Palanque, P., Paternò, F., Santoro, C., Bastide, R.. *A Tool Suite for Integrating Task and System Models through Scenarios*. In Revised Papers of 8th International Wordshop, DSV-IS 2001. Glasgow, Scotland, UK. Junho de 2001.
- [Nune02] Nunes, N. J., e Cunha, J. F.. *Towards Flexible Automatic Generation of User-Interfaces via UML and XMI*. In: Proceedings of the 5 Workshop Iberoamericano de Ingeniera de Requisitos y Ambientes Software (IDEAS'2002), La Habana, Cuba. 23 a 26 de abril. 2002.
- [Pari02] Paris, C., Linden, K. V., Lu, S.. *Automated Knowledge Acquisition for Instructional Text Generation*. In Proceedings of the 20th annual international conference on Computer documentation, pp. 142-151. Toronto. Canadá. 2002.
- [Pari03] Paris, C., Lu, S., Linden, K. V.. *Environments for the Construction and Use of Task Models*. To appear in The Handbook of Task Analysis, D. Diaper and N. Stanton (eds), Lawrence Earlbaum Associates, 2003. Disponível em junho de 2003 em <http://www.calvin.edu/~kvlinden/distributions/Paris-VL-Lu-HTA-draft.doc>.

- [Pate97] Paternò, F., Mancini, C., Meniconi, S.. *ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models*. Proceedings Interact'97, pp. 362-369. Sydney, Austrália. Julho de 1997.
- [Pate01] Paternò, F., Mori, G., Galiberti, R.. *CTTE: An Environment for Analysis and Development of Task Models of Cooperative Applications*. CHI'01 extended abstracts on Human factors in computer systems, pp. 21-22. Seattle. Washington. 2001.
- [Pate02] Paternò, F.. *Tools for Task Modelling: Where we are, Where we are headed*. In 1st International Workshop on Task Models and DIAgrams for user interface design - Tamodia 2002. Bucharest, Romania. Julho de 2002.
- [Penn00] Penner, R. R. and Steinmetz, E. S.. *Dynamic User Interface Adaptation Based on Operator Role and Task Models*. Proceedings of Systems, Man, and Cybernetics, Nashville, TN, October, 2000. Disponível em dezembro de 2002 em www.iterativity.com/papers/2000/DigbeSMCCConference/.
- [Puer01] Puerta, A., Eisenstein, J.. *A Representational Basis for User Interface Transformations*. In: Wiecha, Ch., Szekely, P. (eds): Proc. Of CHI'2001 Workshop "Transforming the UI for anyone, anywhere - Enabling an increased variety of users, devices, and tasks through interface transformations". Seattle. Abril de 2001.
- [Puer03] Puerta, A., Eisenstein, J.. *XIML: A Universal Language for User Interfaces*. Disponível em novembro de 2003 em www.ximl.org/Docs.asp.
- [Quei01] de Queiroz, J. E. R.. *Abordagem híbrida para a avaliação da usabilidade de interfaces com o usuário*. Tese de Doutorado, GIHM, UFCG, Campina Grande, Pb, Junho de 2001.
- [Rich03] Richter, K.. *Chapter 2 - Generic Interface Description using XML*. Disponível em dezembro de 2003 em <http://www.embassi.de/publi/veroeffent/Richter.pdf>.
- [Scai01] Scaico, A.. *Aplicação de um Modelo de Navegação de IHM ao Contexto de Sistemas Industriais*. Dissertação de Mestrado em Engenharia Elétrica. UFCG. Agosto de 2001.
- [Scap89] Scapin, D.L, Pierret-Goldbreich, C.. *Towards a Method for Task Description: MAD*. Unité de Recherche, INRIA, Rocquencourt, France, 1989.
- [Somm03] Sommerville, I.. *Engenharia de Software*. São Paulo. Addison Wesley. 2003.

- [Sous99] de Sousa, M. R. F.. *Avaliação Iterativa da Especificação de Interfaces com Ênfase na Navegação*. Tese de Doutorado, Coordenação de Pós-Graduação em Engenharia Elétrica - COPELE, UFPB, Campina Grande. Dezembro de 1999.
- [Souz99] Souza, C. S., Leite, J. C., Prates, R. O., Barbosa, S. D. J.. *Projeto de Interfaces de Usuário - Perspectivas Cognitivas e Semióticas*. Anais da Jornada de Atualização em Informática, XIX Congresso da Sociedade Brasileira de Computação. Rio de Janeiro. Julho de 1999. Disponível em dezembro de 2002 em www.dimap.ufrn.br/~jair/piu.
- [Sy00] Sy, O., Bastide, R., Palanque, P., Le, D., Navarre, D.. *PetShop : a CASE Tool for the Petri Net Based Specification and Prototyping of CORBA Systems*. Em Petri Nets 2000 - 21st International Conference on Application and Theory of Petri Nets. Aarhus, Dinamarca. Junho 2000. Disponível em junho de 2003 em www.daimi.au.dk/pn2000/proceedings/pn2000_tooldemos.pdf.
- [Tarb96] Tarby, J., Barthet, M.. *The Diane+ Method*. In: Proceedings of the 2nd International Workshop on Computer-Aided of User Interfaces CADUI 96. pp. 95-119. Namur. Junho de 1996. Disponível em novembro de 2003 em www.isys.ucl.ac.be/etudes/cours/iag3960/Tarby-CADUI96.pdf.
- [Turn00] Turnell, M. F. Q. V.. *Projeto de Interface Homem-Máquina*. Notas de Aula. UFCG, Campina Grande, PB. 2000.
- [Turn01] Turnell, M. F. Q. V., Scaico, A., Sousa, M. R. F., Perkusich, A.. *Industrial User Interface Evaluation Based On Coloured Petri Nets Modelling and Analysis*. Lecture Notes in Computer Science - Interactive Systems, LNCS 2220, p. 69-87, Germany, 2001.
- [Vand93] Vanderdonckt, J. Bodart, F.. *Encapsulating Knowledge for Intelligent Automatic Interaction Objects Selection*. in ACM Proc. of the Conf. on Human Factors in Computing Systems INTERCHI'93 (Amsterdam, 24-29 abril 1993), S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel & T. White (éds.), ACM Press, New York, 1993, pp. 424-429.
- [Vand01] Vanderdonckt, J., Bouillon, L., Souchon, N.. *Flexible Reverse Engineering of Web Pages with Vaquista*. In: Proc. WCRE'200: IEEE 8th Working Conference on Reverse Engineering. Stuttgart. Outubro de 2001.

- [Veer00] van der Veer, G. and van Welie, M.. *Task based groupware design: putting theory into practice*. Proceedings of the conference on Designing interactive systems, pp. 326-337. New York, USA. 2000.
- [Vögl03] Vögler, G.. *UIML - User Interface Markup Language - Introduction and Overview*. XML - Arbeitskreis der GI - Regionalgruppe München. Fevereiro de 2003.
- [W3C00] W3C XML Working Group. Extensible Markup Language (XML) 1.0 (Second Edition). W3C. Outubro de 2000. Disponível em junho de 2003 em <http://www.w3.org/TR/REC-xml#sec-intro>.
- [Weli98] van Welie, M., van der Veer, G. C., Eliëns, A.. *Euterpe - Tool support for analyzing cooperative environments*. In: Ninth European Conference on Cognitive Ergonomics. Limerick, Ireland. Agosto de 1998. Disponível em dezembro de 2002 em www.cs.vu.nl/~martijn/gta/docs/ecce9.pdf.
- [Weli99] van Welie, M.. *Euterpe Quick Manual and Quick Reference Guide*. Division Mathematics and Computer Science. Vrije Universiteit Amsterdam. Agosto de 1999.
- [Weli01] van Welie, M.. *Task-based User Interface Desing*. Ph.D. Thesis. Vrije Universiteit, Amsterdam. Abril de 2001. Disponível em junho de 2003 em <http://www.cs.vu.nl/%7Egerrit/gta/docs/Welie-PhD-thesis.pdf>.
- [Weli03] van Welie, M., van der Veer, G.C.. *Groupware Task Analysis*. In: Handbook of Cognitive Task Design, Eds: Erik Hollnagel, pp. 447-476. Lawrence Erlbaum Associates, New Jersey, US. 2003. Disponível em novembro de 2002 em www.welie.com/about.html .
- [Wils93] Wilson, S., Johnson, P., Kelly, C., Cunningham, J., Markopoulos, P. *Beyond hacking: a model based approach to user interface design*. In Proceedings of HCI'93, J. Alty, D. Diaper and S. Guest (eds), Cambridge University Press, pp. 217- 231, 1993.
- [Wils95] Wilson S., & Johnson P.. *Empowering Users in a Task-Based Approach to Design*. Proceedings of DIS'95, Symposium on Designing Interactive Systems, Ann Arbor, Michigan, ACM Press. pp. 25-31. Agosto de 1995.

Bibliografia Complementar

- [Azev00] Azevedo, P., Merrick, R., Roberts, D.. *OVID to AUIML - User-Oriented Interface Modelling*. In: Proceedings of Towards a UML Profile for Interactive System development (TUPIS'00). York. UK. 2000.
- [Baro02] Baron, M.. *De la conception à la construction d'application sûre*. Proceedings of the 14th French-speaking conference on Human-computer interaction, pp. 285-286. Poitiers, France. 2002.
- [Beck00] Beck, K.. *Extreme Programming Explained*. Reading, MA: Addison-Wesley, 2000.
- [Boda94] Bodart, F., Hennebert, A.-M., Leheureux, J.-M., Vanderdonckt, J.. *A Model-based Approach to Presentation: A Continuum from Task Analysis to Prototype*. Chapter 6, in "Interactive Systems: Design, Specification and Verification", F. Paterno (ed.), Focus on Computer Graphics Series, Springer-Verlag, pp. 77-94, Berlin, 1995.
- [Cons99] Constantine, L. L. and Lockwood, L. A. D.. *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*. Addison -Wesley, 1999.
- [Grif98] Griffiths, T., McKirdy, J., Forrester, G., Paton, N., Kennedy, J., Barclay, P., Cooper, R., Goble, C., Gray, P.. *Exploiting model-based techniques for user interfaces to databases*. VDB-4, pp. 21-46, Italy, May 1998.
- [Guer01] Guerrero, C. V., Lula, B. Jr.. *"Um Tutorial na Web" Obtenção do Modelo da Interação a partir do Modelo da Tarefa com o auxílio de Regras Ergonômicas*. Relatório Técnico RT DSC-002/2001. Universidade Federal da Paraíba. Campina Grande, Brasil. Outubro de 2001.
- [Guer03] C. V. S. Guerrero. *Proposta de um arcabouço para concepção e avaliação de interfaces ergonômicas*. Relatório - Projeto de Pesquisa I. COPELE, UFCG, Campina Grande PB, janeiro de 2003.
- [Jaco99] Jacobson, I., Booch, G. and Rumbaugh, J.. *The Unified Software Development Process*. Addison Wesley, 1999.

- [John93] Johnson, P., Wilson, S., Markopoulos, P., Pycok, J.. *ADEPT - Advanced Design Environment for Prototyping with Task Models*. INTERCHI'93 Conference Proceedings, Amsterdam: ACM, 1993.
- [Kafu00] Kafure, I. *Validação do Formalismo TAOS para a Análise da Tarefa no Contexto da Concepção de Interfaces Homem-Computador*. Dissertação de Mestrado. Universidade Federal da Paraíba. CCT. COPIN. Campina Grande, PB. Fevereiro de 2000.
- [Mich03] Michel, T.. *XForms - The Next Generation of Web Forms*. W3C Interaction domain. Agosto de 2003. Disponível em agosto de 2003 em <http://www.w3.org/MarkUp/Forms/>.
- [Paga02] Paganelli, P. and Paternò, F. *Automatic reconstruction of the underlying interaction design of web applications*. Proceedings of the 14th international conference on Software engineering and knowledge engineering, pp. 439-445. Ischia, Italy. 2002.
- [Pate99] Paternò, F., Mancini, C.. *Developing task models from informal scenarios*. CHI'99 extended abstracts on Human factors in computer systems, pp. 228-229. Pittsburg, Pennsylvania. 1999.
- [Pato94] Paton, N. W., al-Qaimari, G., Doan, K.. *On Interface Objects In Object-Oriented Databases*. Proc. 12th British National Conference on Databases (BNCOD), D.S. Bowers (ed), 153-169, Springer-Verlag, 1994.
- [Penn01] Penner, R. R. and Steinmetz, E. S.. *Model-Based Automation of the Design of User Interfaces to Digital Control Systems*. To appear in International Journal of Systems, Man, and Cybernetics. Disponível em junho de 2003 em www.iterativity.com/papers/2001/DIGBE.pdf.
- [Puer01] Puerta, A., Eisenstein, J.. *XIML: A Common Representation for Interaction Data*. In: Proceedings ACM IUI'01. pp. 214-215.
- [Shne98] Shneiderman, B.. *Designing the User Interface - Strategies for effective Human-Computer Interaction*. Addison Wesley Publishing Co. 3rd Edition. 1998.
- [Silv00] da Silva, P. P., Griffiths, T., Paton, N. W.. *Generating User Interface Code in a Model Based User Interface Development Environment*. Proc. Advanced Visual Interfaces, V. di Gesu, et al. (eds), ACM Press, 155-160, 2000.

[Sohn03] Sohn, T. Y. and Dey, A. K.. *iCAP: An Informal Tool for Interactive Prototyping of Context-Aware Applications*. CHI 2003. Ft. Lauderdale, Florida, USA. Abril de 2003. Disponível em maio de 2003 em guir.berkeley.edu/pubs/chi2k3/tsohn-chi2003-ip-icap-final.pdf

[Tamo03] Tamot help. Presente na ferramenta TAMOT3 versão 2.19.

[Usab03] Usability Net. *International standards for HCI and usability - ISO/IEC FDIS 9126-1*. Disponível em junho de 2003 em www.usabilitynet.org/tools/r-international.htm#9126-1.

Anexo A - Outros Suportes

Os suportes apontados no Quadro 46 compreendem o conjunto de ambientes e ferramentas que:

- não estão disponíveis para *download*;
- estão em fase de implementação; ou
- estão disponíveis porém não possuem informações suficientes sobre como usá-las.

Quadro 46 - Outros suportes apurados

Ferramenta	Observações
Damask	<p>Grupo Associado: James Lin Group for User Interface Research Universidade da Califórnia em Berkeley</p> <p>Objetivo: Suporte aos estágios iniciais de projetos de interfaces com o usuário para múltiplos dispositivos.</p> <p>Status: Proposta de Dissertação</p> <p>Referências: [Lin02b] http://guir.cs.berkeley.edu/projects/damask/proposal.pdf</p> <p>Observações: Não disponível para <i>download</i>.</p>
subArtic	<p>Grupo Associado: Scott Hudson e Ian Smith Graphics, Visualization and Usability Center</p> <p>Objetivo: <i>Toolkit</i> para a construção de interfaces com o usuário.</p> <p>Status: Versão 0.8e</p> <p>Referências: http://www.cc.gatech.edu/gvu/ui/sub_arctic/</p> <p>Observações: Disponível para <i>download</i>, porém documentação insuficiente sobre forma de funcionamento da aplicação.</p>
Adligo	<p>Grupo Associado: Said S. Elnaffar e T.C. Nicholas Graham Department of Computing and Information Science Queen's University</p> <p>Objetivo: Suporta a geração de ligações entre um modelo de tarefas em UAN e um modelo Arquitetural Clock (<i>Clock architectural</i></p>

	<p><i>model</i>)</p> <p>Status: Dissertação de Mestrado</p> <p>Referências: [Elna99] http://www.cs.queensu.ca/home/elnaifar/Publications/adligo.pdf</p> <p>Observações: Não disponível para <i>download</i>.</p>
Amulet	<p>Grupo Associado: User Interface Software Group Human Computer Interaction Institute</p> <p>Objetivo: Consiste em um ambiente para desenvolvimento de interfaces com o usuário para C++.</p> <p>Status: Versão 3.0</p> <p>Referências: http://www-2.cs.cmu.edu/~amulet/</p> <p>Observações: Disponível para <i>download</i>, porém documentação insuficiente acerca da forma de funcionamento da aplicação.</p>
CPN/Tools	<p>Grupo Associado: Department of Computer Science University of Aarhus</p> <p>Objetivo: Versão para windows da ferramenta Design/CPN</p> <p>Status: -</p> <p>Referências: [Beau00]</p> <p>Observações: Não disponível para <i>download</i>.</p>
FaceKit	<p>Grupo Associado: Roger King e Michael Novak Departamento de Ciência da Computação Universidade do Colorado</p> <p>Objetivo: <i>Toolkit</i> para o projeto de interfaces para <i>databases</i> orientados a objetos</p> <p>Status: -</p> <p>Referências: [King89]</p> <p>Observações: Não disponível para <i>download</i>.</p>
SUIDT	<p>Grupo Associado: Mickaël Baron e Patrick Girard Laboratoire d'Informatique Scientifique et Industrielle ENSMA</p> <p>Objetivo: Ferramenta que, através de técnicas de programação visual permite, baseando-se em modelos, o desenvolvimento de aplicações.</p> <p>Status: -</p> <p>Referências: [Baro02] http://www.lisi.ensma.fr/members/baron/</p> <p>Observações: Não disponível para <i>download</i>.</p>

Anexo B - Formalismos para a modelagem da tarefa

O presente t3pico vem ao encontro da necessidade de esclarecer detalhes sobre alguns dos formalismos pesquisados, tanto os mais conhecidos como aqueles que servem de base para algumas das ferramentas apuradas.

CTT

Segundo os autores, inicialmente pensou-se em utilizar o formalismo UAN, devido ao suporte a especifica33es hier3rquicas e a exist3ncia de operadores temporais que expressam os relacionamentos entre tarefas. Entretanto houve duas limita33es decisivas:

- em UAN toda a especifica33o 3 textual, o que tende a dificultar a leitura e interpreta33o dos modelos gerados; e
- suporte limitado para a obten33o de uma arquitetura de software.

Outra op33o foi a linguagem LOTOS que consiste em notaa3o internacional largamente utilizada por centros industriais e de pesquisa na especifica33o de sistemas. Por3m, devido a algumas dificuldades e falhas apresentadas no uso da notaa3o, pensou-se em um formalismo que se sobressaísse a estes problemas.

O *ConcurTaksTrees* (CTT) foi desenvolvido por Patern3 et al. e baseia-se tanto em UAN quanto em LOTOS [Pate97], atrav3s dele 3 possív3l descrever os objetivos do usu3rio decompondo-os de forma hier3rquica, iniciando-se pelo objetivo (nív3l mais alto), passando por tarefas e subtarefas intermedi3rias e chegando-se ao nív3l das a33es (nív3l mais baixo) (Figura 59).

As a33es podem ser classificadas em tr3s tipos: (I) cognitivas, (II) l3gicas ou (III) f3sicas. Os objetos est3o relacionados com as a33es e podem ser classificados em: percebív3is pelo usu3rio ou internos 3 aplica33o.

Para cada tarefa h3 um conjunto de atributos relacionados (Quadro 47).

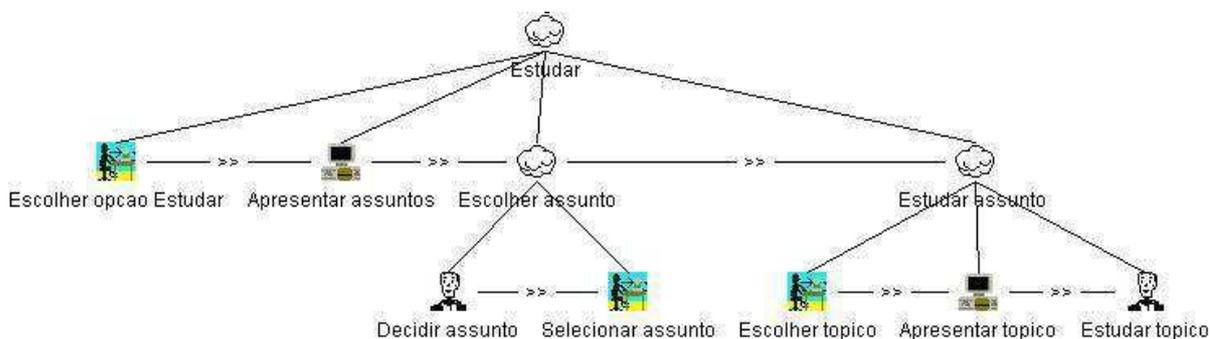


Figura 59 - Árvore de tarefas CTT

Quadro 47 - Atributos do CTT

Informações gerais	
Identificador	Nome dado à tarefa e que a identifica na árvore gerada
Nome	Nome extenso da tarefa
Categoria	Abstrata / Interação / Aplicação / Usuário
Tipo	Tipos pré-definidos para cada categoria, (sendo possível criar novos tipos) e que tem por finalidade auxiliar o projetista a agrupar as tarefas.
Frequência	Baixa / Média / Alta
Plataforma	PDA Desktop Cellphone Outra definida pelo usuário
Descrição	Descrição da tarefa
Iterativa	<i>Checkbox</i> permitindo definir a tarefa como iterativa
Opcional	<i>Checkbox</i> permitindo definir a tarefa como opcional
Parte de uma tarefa cooperativa	<i>Checkbox</i> permitindo definir a tarefa como parte de tarefa cooperativa
Pré-condição	Definição da pré-condição para a execução da tarefa
Objetos relacionados	
Nome	Nome do objeto
Classe	Texto Numérico Gráfico Imagem Posição
Tipo	Percebível / Aplicação
Acesso	Acessível / Modificável
Cardinalidade	Baixa / Média / Alta
Plataforma	Plataforma onde objeto é utilizado
Desempenho em tempo	
Min	Valor de tempo mínimo para a execução da tarefa
Max	Valor de tempo máximo para a execução da tarefa
Média	Valor de tempo médio para a execução da tarefa

No modelo gerado, as ações são classificadas em quatro tipos diferentes de tarefas:

- **user tasks** (tarefas do usuário), representando as tarefas executadas pelo usuário requerendo atividades cognitivas ou físicas;
- **application tasks** (tarefas da aplicação), representando tarefas executadas, exclusivamente e completamente, pela máquina;

- *interaction tasks* (tarefas de interação), representando as tarefas que requerem interação entre usuário e máquina;
- *abstract tasks* (tarefas abstratas), representando tarefas complexas de forma a necessitarem subtarefas que exigem ações tanto do usuário quanto da aplicação.

Para cada um dos tipos de tarefas, a ferramenta CTTE disponibiliza um símbolo, visando facilitar a visualização e compreensão do modelo gerado.

Operadores:

- *Independent Concurrency*, tarefas executadas de forma independente uma da outra;
- *Choice*, equivalente ao **ou exclusivo**, ou seja, apenas uma tarefa do subconjunto é executada;
- *Concurrency with information exchange*, tarefas executadas concorrentemente, porém sincronizadas através de troca de informações;
- *Order Independence*, todas as tarefas podem ser executadas, porém uma de cada vez;
- *Deactivation*, quando a execução de T_2 provoca a interrupção definitiva de T_1 ;
- *Enabling*, o término de execução de uma tarefa provoca o início da execução de uma outra tarefa;
- *Enabling with information passing*, o término de execução de uma tarefa T_1 provoca o início da execução de uma outra tarefa T_2 e a passagem de parâmetros de T_1 para T_2 ;
- *Suspend-resume*, execução de T_2 interrompe a execução de T_1 que só volta a executar após o término de T_2 ;
- *Iteration*, a tarefa executa repetitivamente até que outra tarefa a interrompa (quando retorna ao começo da iteração) ou a desative;
- *Optional tasks*, quando a execução da tarefa é opcional.

DIANE+

Estendido por Jean-Claude Tarby e Marie-France Barthet a partir do formalismo Diane, visa ser um método de desenvolvimento e projeto, a ser compreendido tanto por designers quanto por usuários, de forma a tornar possível a geração e manutenção automática de interfaces do usuário [Tarb96].

Possui como características principais:

- uso de notação gráfica explicitando quem é responsável pela execução da tarefa, seja ele o usuário, o sistema ou a interação entre ambos;
- possibilidade de representar vários atributos tanto de tarefas quanto do fluxo entre tarefas. Atributos, tais como: pré e pós-condições ou constantes;
- tarefas complexas podem ser representadas, de forma relativamente simples, através da combinação de outras tarefas.

Para tanto, além do uso de simbologia específica (Figura 60), o formalismo embasa-se em três elementos:

- **tarefas** que representam os objetivos. Cada tarefa pode conter um conjunto de atributos que a descreve;
- **conexões** que conectam uma tarefa a outra. As conexões permitem definir uma seqüência de execução, podendo ser de dois tipos:
 - * incondicional;
 - * condicional.
- **conectores booleanos** que através de operadores lógicos permitem controlar a rota do fluxo de execução. Possui três tipos:
 - * **Or** (ou);
 - * **And** (e);
 - * **Xor** (ou exclusivo).

Assim como outros formalismos que se propõem a modelar tarefas, Diane+ representa as tarefas como uma decomposição contendo relações lógicas e temporais (Figura 61). Porém um diferencial sobre o formalismo é que as informações são expostas de forma mais explícita

e formal, assim o diagrama resultante indica, explicitamente, se uma tarefa é executada pelo usuário final, pelo sistema ou pela combinação de ambos [Balb99].

Operação automática		Operação requerida	
Operação interativa		Operação opcional	
Operação manual		Operação obrigatória	
Iniciado pelo usuário		Precedência permanente	
Iniciado pelo sistema	Por padrão	Precedência indicativa	
Pré-condição (expressão booleana no início de eventos ou entrada de dados)		Operação automática com sub-operações restritivas	
Pós-condição		Operação interativa com sub-operações restritivas	
Evento (entrada ou saída)		Evento final	

Figura 60 - Formalismo Diane+ [Tarb96]

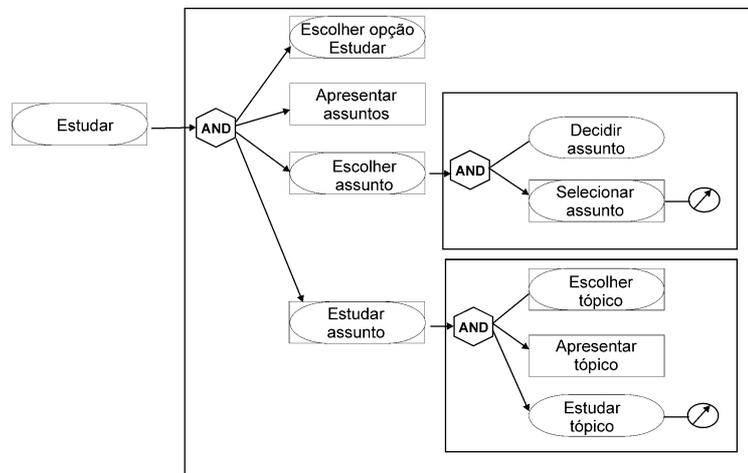


Figura 61 - Árvore de tarefas Diane+

Quadro 48 - Atributos Diane+

Atributos da Tarefa	
Nome	Nome que identifica a tarefa
Semântica (<i>semantics</i>)	Dado gerado automaticamente
Estilo	Interativo / Automática / Manual
Composição	Falso, para tarefas elementares Verdadeiro, para tarefas compostas ou complexas
Decomposição	Tarefas compostas possuem decomposição
Comentário	Descrição sobre a tarefa
Contexto da Tarefa	
<i>Dirty</i> (sujeira)	Falso para as tarefas não processadas pela <i>Language Generation Engine</i> Verdadeiro para as tarefas processadas
<i>Mandatory</i>	Falso para tarefas opcionais Verdadeiro para tarefas obrigatórias
minCard	Definição do tempo mínimo para a tarefa ser executada

maxCard	Definição do tempo máximo para a tarefa ser executada
Pré-condição	Requisitos que devem ser satisfeitos para execução da tarefa
Feedback	Retorno dado pelo sistema quando do término da tarefa
Terminal	Indicador que significa que a tarefa corresponde à tarefa terminal do objetivo

GTA

Método desenvolvido no ano de 1996 por van der Veer, Lenting e Bergevoet, fruto da combinação de aspectos de diversos métodos. O *Groupware Task Analysis* (GTA), essencialmente, consiste em um *framework* conceitual que especifica aspectos relevantes das tarefas que necessitam atenção quando *groupwares* são projetados [Weli01].

Para tanto, o formalismo abrange tanto a modelagem de tarefas de processos já existentes (classificadas como “*task model 1*”) quanto a modelagem de tarefas para processos que ainda deverão ser adotados (classificadas como “*task model 2*”) [Weli03].

O conjunto de termos, bem como os relacionamentos entre estes, sobre os quais o formalismo embasa-se, pode ser observado na Figura 62.

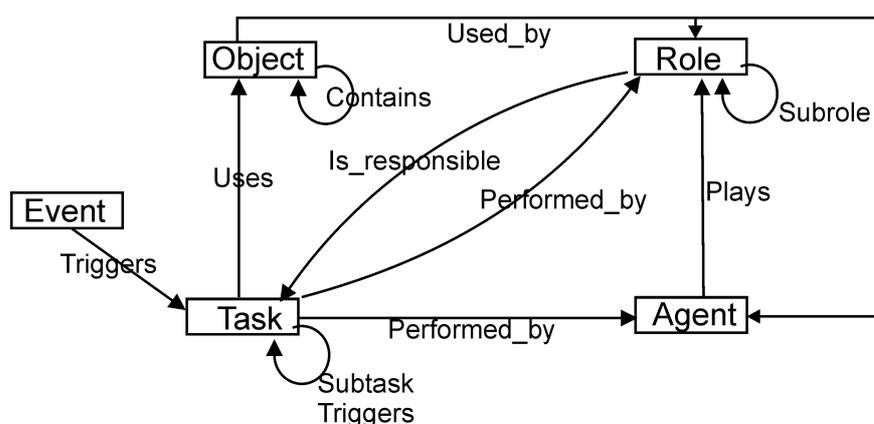


Figura 62 - Ontologia GTA [Weli98]

Os conceitos apontados na ontologia possuem, segundo [Weli98], a seguinte definição:

- **Object** (objetos), refere-se a entidades físicas ou não-físicas, possuindo atributos (nome do atributo e *value pairs*) e ações indicando o que pode ser feito com o objeto. Um objeto pode estar contido em outro objeto;
- **Agent** (agentes), são as entidades consideradas ativas, podendo ser um indivíduo, um grupo de indivíduos ou um grupo de componentes de software;

- **Role** (papéis), consiste em um conjunto de tarefas significativas executadas por um ou mais agentes;
- **Task** (tarefa), consiste em uma atividade executada pelos agentes com o intuito de alcançar um determinado objetivo. Uma tarefa pode ser composta por tarefas elementares (subtarefas) e uma tarefa pode iniciar (*trigger*) a execução de uma ou mais tarefas;
- **Event** (evento), consiste em uma mudança de estado em um ponto determinado do tempo. Os eventos influenciam a seqüência de execução de uma tarefa.

Já as relações entre os conceitos da ontologia, possuem a seguinte definição:

- **Uses** (usa), especifica quais e como os objetos são utilizados na execução de determinada tarefa;
- **Triggers** (iniciador, “gatilho”), consiste na base para a especificação do fluxo de tarefas;
- **Plays** (executa), indica a relação entre quais agentes executam quais papéis;
- **Performed_by** (executada por), indica a relação entre quais tarefas são executadas por quais agentes;
- **Subtask** (subtarefa), indica a decomposição de uma tarefa em subtarefas;
- **Subrole** (sub-papéis), permite a hierarquização dos papéis;
- **Is_responsible** (é responsável), indica uma tarefa para qual um papel é responsável;
- **Used_by** (usado por), permite indicar qual papel ou agente utilizam determinado objeto.

A representação do modelo embasa-se em dois tipos: árvore hierárquica (Figura 63) e descritores (Quadro 49 a Quadro 53).

Quanto aos operadores temporais, em [Weli03], os autores indicam como suficientes os seguintes relacionados:

- **Concorrência** (*concurrent*), as tarefas são realizadas concorrentemente;
- **Escolha** (*choice*), *one out of a set of tasks is done*. Equivalente ao “xor”.

- **Any**, *zero or more tasks of a set of tasks are done in no fixed order*. Equivalente ao “ou”;
- **Seqüência (successive)**, *one task is followed by another*;
- **Independência de ordem (AnyOrder)**, todas as tarefas de determinado conjunto são executadas sem haver ordem fixa;
- **Iteração (*)**, quando combinado com outros construtores é usado para expressar iteração.

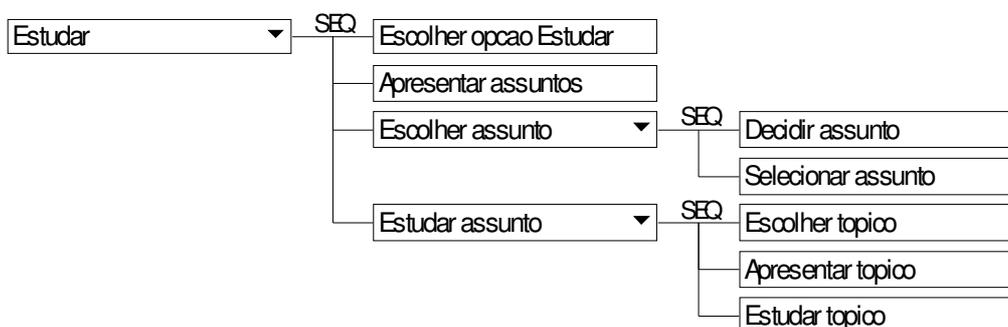


Figura 63 - Árvore de tarefas GTA

Quadro 49 - Descritor da Tarefa

Template da Tarefa	
Nome	Nome que identifica a tarefa
Objetivo	Descrição do objetivo da tarefa
Construtor	Definição do operador a ser usado com subtarefas diretamente relacionadas
Tipo	Complexa / Unitária / Interação
Relevância	Normal / Crítica / Perigosa / Opcional
Mental	<i>Checkbox</i>
Motor	<i>Checkbox</i>
<i>Media Support</i>	Se existe alguma figura ou animação relacionada
Comentário	Descrição sobre a tarefa
Condição de Início	Requisitos que devem ser satisfeitos para execução da tarefa
Condição de Parada	Requisitos que devem ser satisfeitos para finalizar a execução da tarefa
Estado Inicial	?
Estado Final	Resultado final da execução da tarefa
Duração	Tempo de execução da tarefa
Frequência	A frequência que a tarefa é realizada

Quadro 50 - Descritor do Objeto

Template do Objeto	
Nome	Nome que identifica o objeto
Propriedades	Constituída por lista de nome e valor
Usuários	Constituída por lista dos usuários
<i>Rights</i>	Constituída pelo conjunto de opções selecionáveis através <i>checkbox</i> :

	<ul style="list-style-type: none"> - Owner - Create - Destroy - Use/Inspect - Change
<i>Media Support</i>	Se existe alguma figura ou animação relacionada
Comentário	Descrição sobre o objeto

Quadro 51 - Descritor do Papel

<i>Role Template</i>	
Nome	Nome que identifica o papel
<i>Tasks Responsibilities</i>	Conjunto de tarefas
<i>Role Holders</i>	Agentes relacionados a este papel
<i>Media Support</i>	Se existe alguma figura ou animação relacionada
Comentário	Descrição sobre o papel

Quadro 52 - Descritor do Agente

<i>Agent Template</i>	
Nome	Nome que identifica o agente
Tipo	Individual / Organização / Sistema Computadorizado
<i>Roles</i>	Papéis relacionados a este agente
<i>Media Support</i>	Se existe alguma figura ou animação relacionada
Comentário	Descrição sobre o agente

Quadro 53 - Descritor do Evento

<i>Event Template</i>	
Nome	Nome que identifica o evento
<i>Triggers</i>	Conjunto de <i>triggers</i>
<i>Media Support</i>	Se existe alguma figura ou animação relacionada
Comentário	Descrição sobre o evento

MAD

Concebido por Scapin e Pierret-Golbreich [Scap89], o *Méthode Analytique de Description* (MAD) permite a geração de especificações de tarefas de forma a auxiliar as fases iniciais dos projetos, principalmente para a etapa de análise de requisitos.

Segundo van Welie, em [Weli01], a forma de modelar as tarefas através do MAD é bastante similar a utilizada pelo HTA⁴², havendo diferença na presença dos **construtores** e da inexistência de planos (Figura 64).

MAD possui os seguintes construtores:

⁴² HTA: *Hierarchical Task Analysis*.

- **SEQ**, corresponde a execução em seqüência ordenada e estrita das tarefas;
- **PAR**, seqüência de execução porém sem ordem pré-definida;
- **ALT**, escolha exclusiva de tarefas. (XOR);
- **SIM**, segundo a qual indica que várias tarefas são executadas ao mesmo tempo. Porém estas tarefas não trocam informações entre si, mas sim são executadas por agentes diferentes. Ex.: um incêndio é apagado por uma equipe enquanto outra faz o salvamento das vítimas [Cybi96].
- **BOUCLE**, tarefas iterativas.
- **FAC**, tarefas facultativas.

Além disto, para cada tarefa é associado um *template*, onde estão relacionados os detalhes da tarefa em questão (Quadro 54).

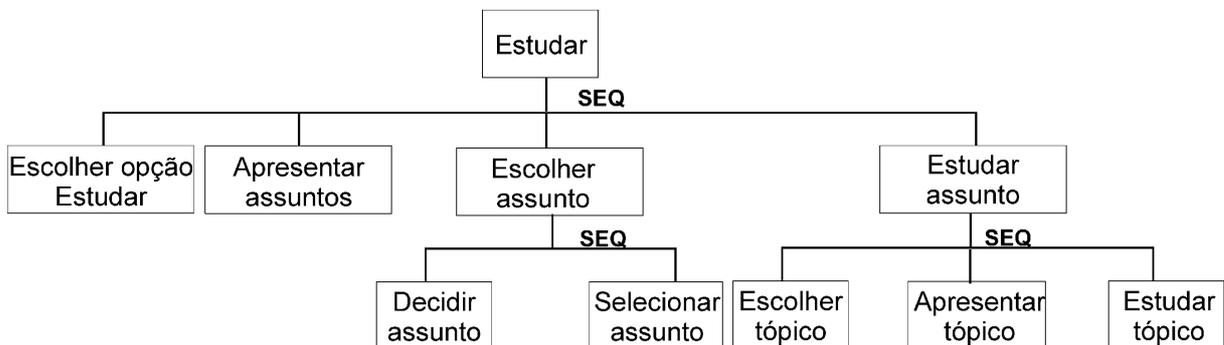


Figura 64 - Árvore de tarefas MAD

Quadro 54 - Descritores da Tarefa

TAREFA: Nome da Tarefa	
ESTADO INICIAL (I): relação dos objetos e parâmetros de entrada para a tarefa (ENTRADA)	ESTADO FINAL (F): relação dos objetos e parâmetros de saída da tarefa (SAÍDA)
OBJETIVO (G): objetivo a ser alcançado a partir da execução da tarefa	
PRÉ-CONDIÇÃO (NPrC): conjunto de predicados que estabelece as restrições sobre o estado dos objetos que são necessários e suficientes para a realização da tarefa	PÓS-CONDIÇÕES (PoC): conjunto de predicados que estabelece as restrições sobre o estado final, que deve ser satisfeito após a execução da tarefa
NÍVEL SUPERIOR: identificador da tarefa antecessora no grafo	TAREFA COMPOSTA: se a tarefa for composta é apresentada a estrutura da tarefa (<u>Estrutura da Tarefa:</u> construtor e seus parâmetros)
TAREFA ELEMENTAR: se a tarefa for elementar contém uma descrição da ação a ser executada	

MAD*

Modele Analytique de Description de tacheS utilisateurs orient specificAtion d'inteRface ou MAD STAR consiste em extensão do método MAD [Hamm95] e possui como principais objetivos:

- solucionar ambigüidades existentes no modelo MAD; e
- aproximar o modelo MAD para o domínio de concepção de interfaces;

Sendo uma extensão do MAD, também MAD* descreve e modela as tarefas de forma hierárquica e através de descritores (Figura 65). A Figura 66 possibilita verificar os principais conceitos associados ao formalismo.

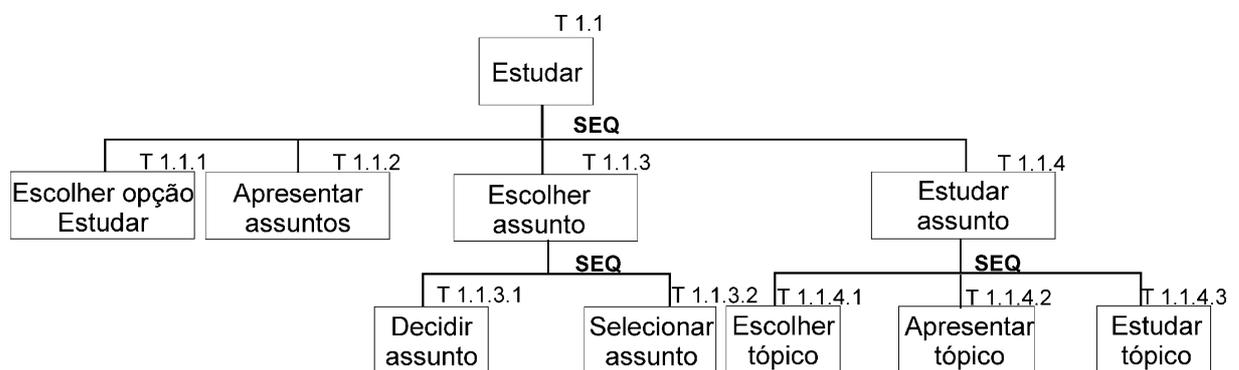


Figura 65 - Árvore de tarefas MAD*

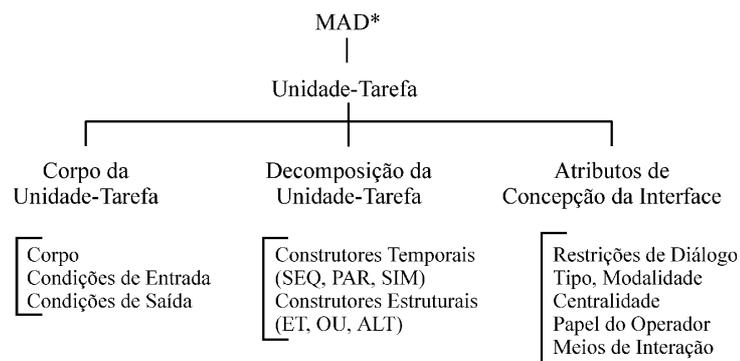


Figura 66 - Conceitos MAD* [Hamm95]

Onde:

- **Unidade-Tarefa (UT)** corresponde a um objetivo, podendo ser de três níveis distintos:
 - UT de alto nível ou Tarefa Raiz: objetivo global da tarefa;
 - UT de intermediário ou Tarefa Intermediária: consistindo nas tarefas intermediárias da execução;

- UT de baixo nível ou Tarefa Elementar: consistindo nas ações da tarefa.
- **Corpo da Unidade-Tarefa:** consiste nos atributos da tarefa (corpo), bem como os atributos de entrada e saída da tarefa;
- **Decomposição da Unidade-Tarefa:** consiste nas relações inter-tarefas, tanto estruturais quanto temporais;
- **Atributos de Concepção da Interface:** aspectos que objetivam auxiliar na exploração da árvore de tarefas através do uso de recomendações ergonômicas de forma a orientar as primeiras decisões de concepção para a futura interface.

Operadores utilizados pelo MAD*:

- **E**, segundo o qual devem ser executadas todas as subtarefas sem haver uma ordem fixa. Tida como paralela independente.
- **OU**, segundo o qual deve ser executado pelo menos uma das subtarefas;
- **ALT**, segundo o qual deve ser executado somente uma das subtarefas. (XOR);
- **SEQ**, segundo o qual as tarefas são executadas de forma seqüencial;
- **PAR**, tarefas que são executadas de forma dependente, na forma de *suspend/resume*, ou seja, uma tarefa é suspensa para que outra trabalhe;
- **SIM**, tarefas que iniciam e finalizam ao mesmo tempo. Implicando uma divisão de atividades entre operadores.

Quadro 55 - Descritores MAD*

Corpo da Unidade-Tarefa	
Corpo da Tarefa	
Número da tarefa	Número que identifica a tarefa no contexto geral
Nome	Nome da tarefa
Objetivo	Objetivo da tarefa
Modo	Definição da modalidade da tarefa (ex.: facultativa ou interativa)
Ator	Definição do usuário que atuará sobre a tarefa
Prioridade	Valor numérico definindo a prioridade de execução da tarefa
Interruptibilidade	Definição se a tarefa pode ou não ser interrompida e em caso afirmativo, como será dado o reinício da tarefa
Estado de execução	Podendo estar “em espera” ou “em execução”
Condições de entrada da tarefa	
Estado do mundo inicial	Relação dos objetos e parâmetros de entrada para a tarefa

Pré-condições / condições desencadeantes	Conjunto de predicados que estabelece as restrições sobre o estado dos objetos que são necessários e suficientes para a realização da tarefa
Condições de saída da tarefa	
Estado do mundo final	Relação dos objetos e parâmetros de saída da tarefa
Pós-condições / resultados da execução	Conjunto de predicados que estabelece as restrições sobre o estado final, que deve ser satisfeito após a execução da tarefa
Decomposição da Unidade-Tarefa	
Relação	Definição do operador temporal
Lista de subtarefas	Conjunto das subtarefas da tarefa
Nível superior	Identificador da tarefa antecessora no grafo
Tarefa Elementar	Se a tarefa for elementar contém uma descrição da ação a ser executada
Atributos orientados à concepção da interface	
Restrição de diálogo	Podendo estar em espera ou interrompida
Tipo	Tipo da tarefa (ex.: mental ou sensória-motor)
Modalidade	Modalidade da execução (ex.: interativa)
Centralidade	Descrição da frequência e entidades relacionadas à execução
Papel do operador	Descrição do papel do operador na execução da tarefa
Meios de interação	Descrição da forma interação

TAOS

Concebido por J. H. de Medeiros, consiste em formalismo para a aquisição e representação do conhecimento baseado na modelagem do domínio.

Segundo apontado em [Cord03], a linguagem considera a existência de comportamentos estáticos e dinâmicos, assim distribuídos:

- conceitos estáticos (objetos, métodos e situações): quando mantêm suas características durante um tempo pré-estabelecido;
- conceitos dinâmicos (processos, ações e planos): quando ocorre uma evolução de uma determinada situação em um determinado intervalo de tempo.

Sua taxonomia pode ser observada na Figura 67 onde os nós da árvore representam as classes e os arcos representam as relações entre as classes [Mede03].

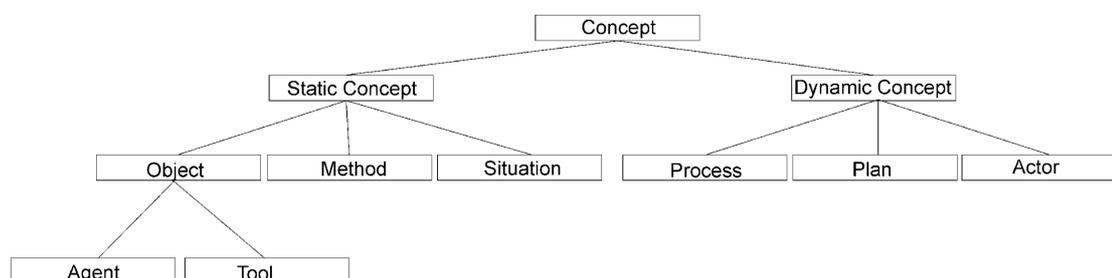


Figura 67 - Meta Modelo TAOS [Mede03]

Apesar de ser um formalismo validado sob o domínio da biologia molecular, o fato de embasar-se em sistema baseado em conhecimento, permite sua utilização na análise e modelagem da tarefa do usuário [Mede03], tanto o é que, Kafure o validou segundo MAD [Cord03].

Após a validação o *Task Action Oriented System* (TAOS) foi estendido de forma a melhor suportar a concepção de interfaces do usuário. Para isto, mudou-se o termo *Plan* e *Sub-Plan* para *Task* e *Sub-Task*, além da inclusão de novos atributos orientados à concepção da interface.

O formalismo representa os dados apurados com o usuário através de descrição hierárquica formal (Figura 68), acrescida de um conjunto de descritores (Quadro 56 ao Quadro 61).

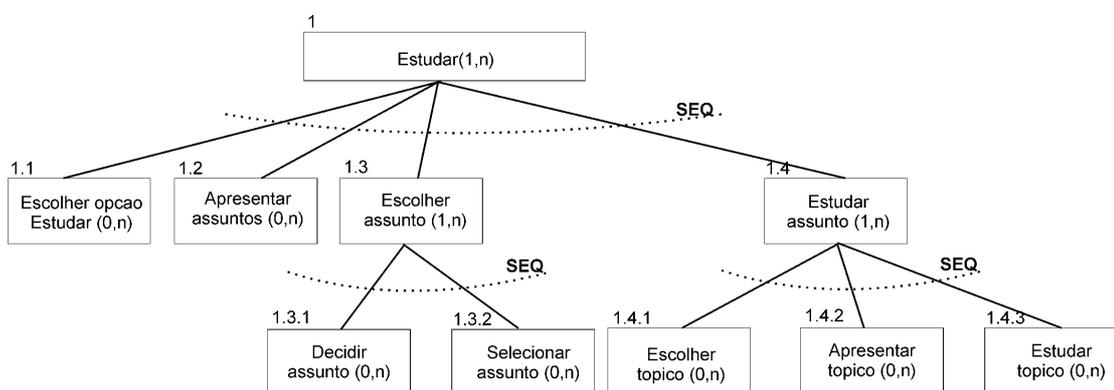


Figura 68 - Árvore tarefas TAOS

Quadro 56 - Descritores de tarefas complexas

Nome	Nome da tarefa
Número	Número gerado automaticamente e que identifica a tarefa na árvore
Ocorrência	(0,0) / (0,1) / (0,n) / (1,1) / (1,n)
Descrição	Descrição da tarefa
Pré-situação	Apontador para objeto do tipo situação
Pós-situação	Apontador para objeto do tipo situação
Ações	Conjunto de ações da tarefa
Subtarefas	Conjunto de subtarefas da tarefa
Como realizar	Expressão que representa o método
Prioridade	Elemento relacionado à simulação
IntERRUPTIBILIDADE	Não interrompível Interrompível com reprise no início Interrompível com reprise em curso
Duração	Calculada com base nos valores apresentados para as tarefas elementares da tarefa
Especificação da Interface orientada a atributos	
Tipo da tarefa	Sensório-motora / Mental / Verbal
Modalidade da tarefa	Manual / Automática / Interativa
Importância da tarefa	Alta / Média / Baixa
Frequência da tarefa	Alta / Média / Baixa

Quadro 57 - Descritor da tarefa elementar ou ação

Nome	Nome da tarefa
Número	Número gerado automaticamente e que identifica a tarefa na árvore
Ocorrência	(0,0) / (0,1) / (0,n) / (1,1) / (1,n)
Descrição	Descrição da ação
Pré-situação	Apontador para objeto do tipo situação
Pós-situação	Apontador para objeto do tipo situação
Lista de agentes	Relação de atores envolvidos na ação
Adicionar ou selecionar agentes	Permite adicionar um agente
Lista de ferramentas	Relação de ferramentas envolvidas na ação
Adicionar ou selecionar ferramentas	Permite adicionar uma ferramenta
Prioridade	Elemento relacionado à simulação
IntERRUPTIBILIDADE	Não interrompível Interrompível com reprise no início Interrompível com reprise em curso
Duração	Definição do tempo, em segundos, necessários para executar a tarefa
Especificação da Interface orientada a atributos	
Tipo da tarefa	Sensório-motora / Mental / Verbal
Modalidade da tarefa	Manual / Automática / Interativa
Importância da tarefa	Alta / Média / Baixa
Frequência da tarefa	Alta / Média / Baixa

Quadro 58 - Descritor de pré e pós-situação

Nome	Nome da pré ou pós-situação
Descrição	Descrição da pré ou pós-situação
Objetos	Objetos relacionados
Adicionar, abrir ou remover objetos	Permite editar os objetos
Restrição	Expressão que permite controlar os objetos
Adicionar restrição	Permite adicionar restrição
Como Atingir	Expressão que representa o método
Adicionar tarefa ou ação ao Como Atingir	Permite adicionar tarefa ou ação à expressão

Quadro 59 - Descritor do método

Nome	Nome do método
Descrição	Descrição do método
Corpo	Expressão que representa o método

Quadro 60 - Descritor do agente

Nome	Nome do agente
Descrição	Descrição do agente
Competências	Relação de ações realizadas pelo agente
Experiência com a tarefa	Alta / Média / Baixa
Experiência computacional	Alta / Média / Baixa
Remover atributos adicionais	Permite retirar os atributos adicionais inseridos
Adicionar atributos adicionais	Permite adicionar atributos ao descritor

Quadro 61 - Descritor da ferramenta

Nome	Nome da ferramenta
Descrição	Descrição da ferramenta
Ator	Relação de agentes associados
Emprego	Relação de ações associadas

Como operadores temporais, o formalismo possui [Mede03]:

- **SEQ**, subtarefas e/ou ações devem ser executados em seqüência;
- **OR**, ao menos uma das subtarefas e/ou ações deve ser executada;
- **XOR**, apenas uma das subtarefas e/ou ações deve ser executada;
- **AND**, todas as subtarefas e/ou ações devem ser executadas sem importância da ordem de execução;
- **SIM**, subtarefas e/ou ações podem ser executadas simultaneamente;
- **PAR**, subtarefas e/ou ações são executados paralelamente, não necessariamente ao mesmo tempo.

TKS

A teoria parte do princípio que o conhecimento que o usuário possui sobre a tarefa é estruturado e pode ser representado como uma *Task Knowledge Structure* (TKS) [Mark97].

Ainda segundo Markopoulos et al., as principais vantagens no uso da teoria são:

- o modelo gerado é preciso e internamente consistente;
- o modelo pode ser validado pelo analista;
- o modelo pode ser compreendido por outras pessoas além do analista.

Como elementos principais, tem-se:

- a estrutura representa a decomposição dos **objetivos**;
- **procedimentos** são compostos de ações, o que pode ser compreendido por outra forma, procedimentos são ações compostas;
- a **ação** corresponde ao elemento mais básico da execução de uma tarefa;
- ações são aplicadas a **objetos**, que por sua vez refletem o conhecimento do domínio da tarefa e são descritos através de atributos e relacionamentos entre outros objetos.

Em [Mark97] os autores indicam que a estrutura de objetivos do usuário e o conjunto de objetos do TKS são modelados através de dois processos paralelos envolvendo as ações

(Figura 69). Partindo-se, portanto, do objetivo do usuário e sua decomposição até as ações, das quais obtém-se os objetos.

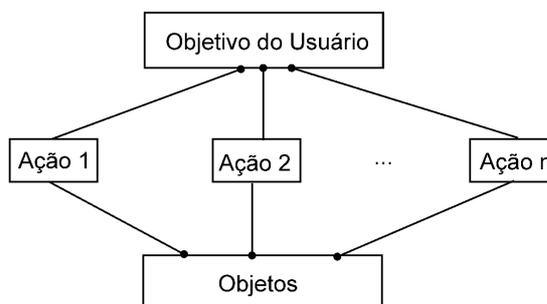


Figura 69 - Estrutura abstrata TKS [Mark97]

Em [Mark94] os autores apontam que as expressões existentes na linguagem LOTOS são utilizadas no formalismo TKS como forma de evitar imprecisão e ambigüidades. No mesmo artigo, é apontado o subconjunto , extraído das definições da linguagem, que foi utilizado para o estudo de caso apresentado.

Conjunto básico de operadores extraído de [Bolo87]:

- **Choice** (\square), equivalente ao “ou exclusivo”;
- **Parallel composition - general case** ($\square\square$), tarefas independentes com pontos sincronizados para troca de informação;
- **Parallel composition - pure interleavingI** ($\square\square\square$), tarefas executadas de forma independente e assíncrona, porém com possibilidade de trocar informações;
- **Full synchronization** (\parallel), quando dois processos são forçados a ocorrer juntos;
- **Sequential composition** (\gg), tarefas são executadas em ordem seqüencial rígida;
- **Disabling** ($\square>$), execução de T_2 interrompe definitivamente a execução de T_1 .