

Universidade Federal de Campina Grande
Coordenação de Pós-Graduação em Informática

Dissertação de Mestrado

Abordagem de Desenvolvimento Evolutivo de Interface do Usuário
Baseada em Modelos e Múltipla Prototipagem: FastInterface

Karolyne Maria Alves de Oliveira

Campina Grande

Agosto – 2008

**Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Informática**

**Abordagem de Desenvolvimento Evolutivo de Interface do Usuário
Baseada em Modelos e Múltipla Prototipagem: FastInterface**

Karolyne Maria Alves de Olivera

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande como requisito parcial para obtenção do grau de mestre em Informática (MSc).

**Profº Bernardo Lula Júnior, DSc.
(Orientador)**

**Área de Concentração: Ciência da Computação
Linha de Pesquisa: Engenharia de Software**

**Campina Grande - Paraíba
Agosto 2008**

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

O48a

2008 Oliveira, Karolyne Maria Alves de.

Abordagem de desenvolvimento evolutivo de interface do usuário baseada em modelos e múltipla prototipagem: FastInterface / Karolyne Maria Alves de Oliveira. — Campina Grande, 2008.
158f. : il.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.

Referências.

Orientador: Prof. Dr. Bernardo Lula Júnior.

1. Interface do Usuário. 2. Prototipagem. 3. Metodologia de Concepção de Interface. 4. Desenvolvimento Baseado em Modelos. I. Título.

CDU 004.5(043)

**"ABORDAGEM DE DESENVOLVIMENTO DE INTERFACE DO USUÁRIO BASEADA EM
MODELOS E MÚLTIPLA PROTOTIPAGEM: FASTINTERFACE"**

KAROLINE MARIA ALVES DE OLIVEIRA

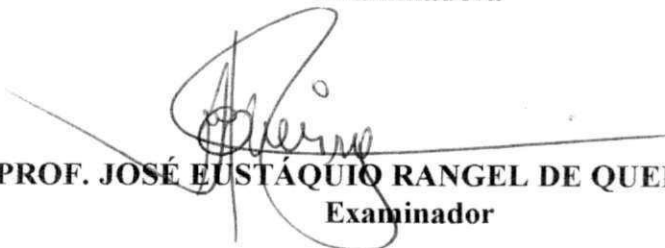
DISSERTAÇÃO APROVADA EM 22.08.2008



PROF. BERNARDO LULA JÚNIOR, Dr.
Orientador



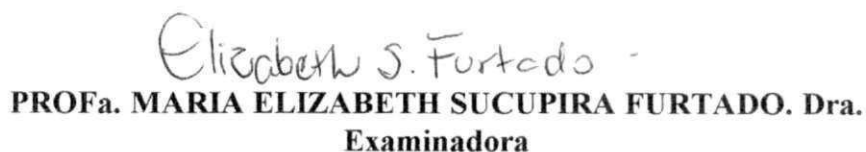
PROFa. FRANCILENE PROCÓPIO GARCIA, D.Sc
Examinadora



PROF. JOSÉ EUSTÁQUIO RANGEL DE QUEIROZ, D.Sc
Examinador



PROF. LAFAYETTE BATISTA MELO, Dr.
Examinador



PROFa. MARIA ELIZABETH SUCUPIRA FURTADO, Dra.
Examinadora

CAMPINA GRANDE – PB

"Nada temas, Eu estou contigo... Eu te fortaleço e venho em teu socorro, Eu te amparo com minha destra vitoriosa" (Isaias 41,10)

*Dedico este trabalho a Deus por toda orientação
e força.*

Agradecimentos

Agradeço, primeiramente, a **Deus**, por toda providência com a qual guiou este trabalho e por toda fonte de inspiração.

Aos meus pais, **Severino e Odete**, por toda atenção, carinho e confiança. Tenho certeza de que seus exemplos de perseverança e principalmente de amor me guiarão por toda vida.

À minha querida irmãzinha **Ana Cristina**, a grande, por ser um reflexo de bondade, humildade e presteza.

À minha **família**, por sempre entender minhas ausências e ser um sustentáculo em todas as horas.

À **Comunidade Consolação Misericordiosa**, na figura da minha formadora **Ana Veruska e grupo de formação**, que nos momentos de tribulação e dificuldades sempre caminharam ao meu lado.

Ao meu orientador **Bernardo**, por toda colaboração e dedicação durante toda caminhada do mestrado.

Aos professores do DSC, principalmente aos professores **Eustáquio e Francilene** por toda contribuição. Aos professores do CEFET-PB por toda confiança e abertura.

Às minhas amigas do peito e para sempre, **Lígia e Michelle**, pelo companheirismo, amor e união.

A **Gustavo Nascimento**, pelo companheirismo e apoio constante.

A todos os amigos que fiz durante o mestrado, em especial a **Zé de Guga e Mila**, por todo apoio e por serem essas pessoas de um coração sem igual; a **Danilo**, por todo incentivo e colaboração durante o decorrer deste trabalho; às queridas **Amanda, Lili, Andréa** por grandes momentos de descontração; e, por fim, as minhas companheiras de trabalho **Márcia e Yuska**.

A todos os fizeram parte da grande família FastInterface, entre eles: **Giuseppe, Anselmo, Claudiana, Luiz, Rodrigo, Carlos Diego, Jânio, Diénert, Gabriela, Nielson e Rafael**. Levarei todos para sempre em meu coração.

A todos meus colegas de trabalho da **DATAPREV**, em especial aos que participaram do estudo de caso.

Por fim, a todos que contribuíram, de forma direta ou indireta, para que este trabalho tivesse êxito.

Resumo

Este trabalho propõe uma abordagem de desenvolvimento de interface com usuário baseado em modelos e múltipla prototipagem evolutiva. A utilização deste tipo de abordagem apresenta-se como fator determinante para adoção efetiva de abordagens baseadas em modelos. Neste caso, protótipos de baixa, média e alta fidelidade podem ser utilizados em consonância com as diferentes fases do processo de desenvolvimento. Essas fases estão definidas no *framework* de referência unificado para desenvolvimento de interfaces do usuário multi-plataforma, também denominado *Cameleon*. A abordagem proposta foi refletida em MEDITE, uma metodologia de concepção de interface que preconiza o uso de diferentes técnicas de prototipagem e pode ser vista sob o prisma de *Cameleon*. Para dar suporte a esta abordagem e agilizar o processo de concepção, um ambiente de desenvolvimento de interface foi desenvolvido e validado dentro do escopo deste trabalho, o FastInterface.

Palavras-Chaves: Desenvolvimento de Interface com usuário Baseado em Modelos, Ambientes de Desenvolvimento Baseados em Modelos, Prototipagem, Metodologia de Concepção de Interface.

Abstract

This work proposes a user interface development approach based on models and multiple evolving prototyping. The utilization of this type of approach consists of a decisive factor to the effective adoption of models based approaches. In this case, low, medium and high fidelity prototypes may be utilized in consonance with the different phases of the development process. Those phases are defined in the unified reference framework for multi-target user interfaces development, named *Cameleon*. Such approach led to MEDITE, which is a methodology to the conception of interfaces that points out to the use of different prototyping techniques and may be seen from the *Cameleon's* perspective. To support this approach and speed up the interface conception process, a user interface development environment, the FastInterface, was conceived and validated in the scope of this work.

Key-Words: Model Based Interface Development, Model Based Interface Development Environment, Prototyping, User Interface Conception Methodology.

Sumário

INTRODUÇÃO.....	16
1.1. CENÁRIO TÉCNICO-CIENTÍFICO	16
1.1.1. <i>Desenvolvimento de Interfaces do Usuário Baseado em Modelos</i>	18
1.1.1.1 Desenvolvimento de Interfaces do Usuário Baseado no Domínio	18
1.1.1.2 Desenvolvimento de Interfaces do Usuário Baseado na Tarefa.....	19
1.1.1.3 Prototipagem da Interface com usuário.....	19
1.1.2. <i>Ambientes de Desenvolvimento de Interfaces do Usuário Baseado em Modelos</i>	23
1.1.3. <i>Antecipação de Representação Visual em Abordagens Baseadas em Modelos</i>	25
1.2. DEFINIÇÃO DO PROBLEMA	27
1.3. OBJETIVOS DO TRABALHO	28
1.3.1. <i>Objetivo geral</i>	28
1.3.2. <i>Objetivos específicos</i>	28
1.4. RELEVÂNCIA	29
1.5. METODOLOGIA DE DESENVOLVIMENTO	30
1.6. ESTRUTURA DA DISSERTAÇÃO.....	31
ABORDAGEM DE DESENVOLVIMENTO DE INTERFACE COM USUÁRIO BASEADA EM MODELOS E MÚLTIPLA PROTOTIPAGEM EVOLUTIVA	32
2.1. MEDITE	32
2.1.1. <i>Tarefas e Conceitos em MEDITE</i>	35
2.1.2. <i>Interface Abstrata do Usuário - Especificação Conceitual Parcial da Interação</i>	36
2.1.2.1. Aspectos de Apresentação.....	37
2.1.2.2. Aspectos de Navegação.....	39
2.1.3. <i>Protótipo de Média Fidelidade em MEDITE</i>	42
2.2. ABORDAGEM DE DESENVOLVIMENTO BASEADO EM MODELOS UTILIZANDO MÚLTIPLA PROTOTIPAGEM EVOLUTIVA.....	44
2.2.1. <i>Extensões nos modelos de MEDITE</i>	45
2.2.2. <i>Protótipo de Alta Fidelidade</i>	48
2.2.3. <i>Necessidade de Integração</i>	50
2.3. CONCLUSÕES.....	51
PROJETO E IMPLEMENTAÇÃO DE HI-FY	53
3.1. PROPÓSITO DE HI-FY.....	53
3.2. METODOLOGIAS DE DESENVOLVIMENTO	54
3.3. TECNOLOGIAS UTILIZADAS.....	56
3.3.1. <i>Framework JHotDraw</i>	56
3.3.2. <i>API JDOM</i>	57
3.3.3. <i>Padrões de projeto</i>	57
3.4. ARTEFATOS.....	58
3.4.1. <i>Requisitos Funcionais</i>	58
3.4.2. <i>Requisitos Não-Funcionais</i>	59
3.4.3. <i>Perfil do Usuário</i>	59
3.4.4. <i>Objetivos de Usabilidade</i>	59
3.4.5. <i>Modelo da Tarefa</i>	60
3.4.6. <i>User Stories e Testes de Aceitação</i>	60
3.4.7. <i>Protótipo da interface</i>	61
3.4.8. <i>Projeto Arquitetural de Hi-Fy</i>	62
3.5. FERRAMENTA HI-FY	64

3.5.1. Modos de Visualização	64
3.5.2. Edições	66
3.6. CONCLUSÃO	66
PROJETO E IMPLEMENTAÇÃO DE FASTINTERFACE	68
4.1. PROPÓSITO DE FASTINTERFACE.....	68
4.2. PROJETO E IMPLEMENTAÇÃO DO AMBIENTE FASTINTERFACE.....	69
4.2.1. Requisitos Funcionais.....	69
4.2.2. Protótipo da interface.....	70
4.2.3. Projeto arquitetural.....	70
4.3. ASPECTOS DE INTEGRAÇÃO	74
4.3.1. Controle da Interface Gráfica.....	75
4.3.2. Persistência e Transformação entre os Modelos.....	77
4.3.3. Mecanismo de Manutenção da Consistência entre os Modelos.....	78
4.3.3.1. Consistência entre o Modelo da Tarefa e Modelo de Interface Abstrata do Usuário	78
4.3.3.2. Consistência entre o Modelo de Interface Abstrata do Usuário e o Modelo de Interface Concreta do Usuário	80
4.4. FASTINTERFACE	80
4.4.1. Múltiplos Projetos	80
4.4.2. Edições em Modelos.....	81
4.4.3. Manutenção da Consistência entre os Modelos.....	82
4.4.4. Modos de Visualização	82
4.5. CONCLUSÕES.....	84
ESTUDO EXPERIMENTAL SOBRE O USO DO AMBIENTE FASTINTERFACE	85
5.1. PERSPECTIVAS DE AVALIAÇÃO	85
5.2. INSPEÇÃO DE CONFORMIDADE AO PADRÃO ISO 9241	86
5.2.1. Resultado da Inspeção de Conformidade de FastInterface ao Padrão ISO 9241-14: Diálogos via menus	87
5.2.2. Resultado da Inspeção de Conformidade de FastInterface ao Padrão ISO 9241-16: Diálogos via manipulação direta.....	88
5.2.3. Resultado da inspeção de conformidade de FastInterface ao Padrão ISO 9241-17: Diálogos via Formulários.....	89
5.2.4. Taxas de adoção.....	90
5.3. ESTUDO DE CASO.....	91
5.3.1. Preparação do estudo de caso	Error! Bookmark not defined.
5.3.1.1. Primeira Etapa – Levantamento de Requisitos para Sistema Consultor.....	92
5.3.1.2. Segunda Etapa – Análise e Modelagem da Tarefa para Sistema Consultor.....	93
5.3.1.3. Terceira Etapa – Preparação de Material de Nivelamento.....	93
5.3.1.4. Quarta Etapa – Planejamento da Avaliação	94
5.3.1.5. Quinta Etapa – Identificação e Recrutamento de Usuários de Teste	95
5.3.2. Estratégia de realização do estudo de caso.....	95
5.3.3. Análise dos resultados obtidos.....	96
5.3.3.1. Análise dos indicadores objetivos.....	96
5.3.3.2. Análise dos indicadores subjetivos.....	99
5.4. SONDAAGEM DA SATISFAÇÃO SUBJETIVA DO USUÁRIO.....	103
5.4.1. Análise do questionário USer– Perfil do Usuário	104
5.4.2. Análise do questionário USE.....	104
5.5. CONCLUSÕES.....	108
CONCLUSÃO E TRABALHOS FUTUROS.....	110
6.1. CONTEXTUALIZAÇÃO GERAL DA DISSERTAÇÃO	110
6.2. CONCLUSÃO	112
6.3. PROPOSIÇÕES PARA TRABALHOS FUTUROS.....	113

APÊNDICE A – MODELO DA TAREFA COM USO DO FORMALISMO TAOS PARA O SISTEMA CONSULTOR.....	120
APÊNDICE C – MATERIAL DE NIVELAMENTO	124
APÊNDICE D – REQUISITOS FUNCIONAIS DO SISTEMA CONSULTOR.....	128
APÊNDICE E – TUTORIAL DE USO DOS AMBIENTES FASTINTERFACE E TERESA	131
APÊNDICE F – ROTEIROS DE TESTE DE USABILIDADE PARA AMBIENTES FASTINTERFACE E TERESA.....	143
APÊNDICE G – E-MAIL PARA RECRUTAMENTO DE USUÁRIOS PARA O ESTUDO DE CASO	151
APÊNDICE H - QUESTIONÁRIO USER (USERSKETCHER).....	152
APÊNDICE I - QUESTIONÁRIO USE (USERSATISFACTIONENQUIRER).....	153

Lista de Figuras

FIGURA 1- <i>FRAMEWORK CAMELEON</i>	24
FIGURA 2- FLUXO DE <i>MEDITE</i>	33
FIGURA 3 – PERSISTÊNCIA DA FERRAMENTA <i>ITAOS</i>	35
FIGURA 4 – RASTREABILIDADE ENTRE OS MODELOS DA TAREFA, ROTEIRO E INTERAÇÃO	36
FIGURA 5 - OBTENÇÃO DOS ESTADOS DE UM <i>STATECHART</i> A PARTIR DO MODELO DE INTERAÇÃO (RODRIGUES, 2005)	40
FIGURA 6 – REPRESENTAÇÃO DO ESPAÇO INICIAL GERADA PELA FERRAMENTA <i>SMILE</i>	43
FIGURA 7 – TELA DA FERRAMENTA <i>SMILE</i> DURANTE EDIÇÃO DO PROTÓTIPO DE MÉDIA FIDELIDADE.....	44
FIGURA 8 – ESQUEMA DO TRATAMENTO DADO À VISÃO FUNCIONALIDADE	47
FIGURA 9 – CRIAÇÃO DE TRANSIÇÕES PARA TAREFA COM APENAS UM FILHO	47
FIGURA 10 - REPRESENTAÇÃO <i>CUI</i> A PARTIR DA <i>AUI</i>	49
FIGURA 11 – CORRESPONDÊNCIA ENTRE OS NÍVEIS DE ABSTRAÇÃO PROPOSTOS EM <i>CAMELEON</i> E AS FASES E ARTEFATOS DE <i>MEDITE</i>	50
FIGURA 12 - FLUXO DO PROCESSO DE DESENVOLVIMENTO <i>EASYPROCESS</i>	55
FIGURA 13 – VISÃO DE PROTÓTIPOS DE DIFERENTES FIDELIDADES DA FERRAMENTA <i>HI-FY</i>	62
FIGURA 14 – ARQUITETURA DO MÓDULO <i>HI-FY</i>	62
FIGURA 15 - DIAGRAMA DE CLASSES DA CAMADA DE NEGÓCIO DO <i>HI-FY</i> COM ÊNFASE NO MODELO <i>CUI</i>	65
FIGURA 16 – MODO DE VISUALIZAÇÃO ÁRVORE DE PROTÓTIPO	64
FIGURA 17 – MODO DE VISUALIZAÇÃO PROTÓTIPO	65
FIGURA 18 – MODO DE VISUALIZAÇÃO <i>SPLIT</i>	65
FIGURA 19 – EDIÇÃO DE PROTÓTIPO DE ALTA FIDELIDADE EM <i>HI-FY</i>	66
FIGURA 20 – PROTÓTIPO DE ALTA FIDELIDADE DE <i>FASTINTERFACE</i>	70
FIGURA 21 – INTERAÇÃO ENTRE CAMADAS E MÓDULOS DE <i>FASTINTERFACE</i>	71
FIGURA 22 – RELAÇÃO ENTRE <i>JHD</i> E <i>FASTINTERFACE</i>	72
FIGURA 23 – DIAGRAMA CONCEITUAL DE CLASSES DO PACOTE <i>FAST.BUSINESS</i>	73
FIGURA 24 – RELAÇÃO ENTRE <i>JDOM</i> E CAMADA DE PERSISTÊNCIA DE <i>FASTINTERFACE</i>	74
FIGURA 25– ESTRUTURA DO CONTROLADOR DE MÓDULOS	76
FIGURA 26 – ORGANIZAÇÃO DOS ARQUIVOS DO TIPO <i>FASTINTERFACEXML</i>	77
FIGURA 27 – ÁRVORE DE PROJETOS EM <i>FASTINTERFACE</i>	80
FIGURA 28 – ÁREA DE EDIÇÃO DE MODELOS NO <i>FASTINTERFACE</i>	81
FIGURA 29 - AMBIENTE <i>FASTINTERFACE</i> E AS ABAS DOS MÓDULOS.....	82
FIGURA 30 – PROJETO ABERTO NO MODO DE JANELA DE ABAS EXTERNA	83
FIGURA 31 – PROJETO ABERTO NO MODO DE ABAS INTERNAS À JANELA PRINCIPAL.....	83
FIGURA 32 – INTERFACE FINAL GERADA PELO AMBIENTE <i>TERESA</i> PARA TAREFA CADASTRAR EMPRESA E EFETUAR <i>LOGIN</i>	102
FIGURA 33 – INTERFACE FINAL GERADA PELO AMBIENTE <i>FASTINTERFACE</i> PARA A TAREFA CADASTRAR EMPRESA.....	103
FIGURA 34 - INTERFACE FINAL GERADA PELO AMBIENTE <i>FASTINTERFACE</i> PARA A TAREFA EFETUAR <i>LOGIN</i>	103

Lista de Quadros

QUADRO 1 - VISÃO DOS MB-IDES PELO PRISMA DO <i>FRAMEWORK CAMELEON</i>	25
QUADRO 2 – REGRAS DE MAPEAMENTO DO MODELO DA TAREFA (MTA) PARA MODELO DE ROTEIRO (MRO)	38
QUADRO 3 – EVENTOS ACEITOS PELO SISTEMA DADA A OCORRÊNCIA DA TAREFA	41
QUADRO 4 – COMBINAÇÕES POSSÍVEIS <i>OCORRÊNCIA</i> VERSUS <i>MÉTODO</i>	42
QUADRO 5 – NOVAS REGRAS DE MAPEAMENTO DO MODELO DA TAREFA PARA MODELO DE ROTEIRO.....	46
QUADRO 6 – REGRAS DE TRANSFORMAÇÃO ENTRE MODELO DE INTERFACE ABSTRATA DO USUÁRIO (AUIMODEL) E MODELO DE INTERFACE CONCRETA DO USUÁRIO (CUIMODEL).....	48
QUADRO 7 – DESCRIÇÃO DAS PRINCIPAIS CLASSES E INTERFACES DE JHD QUE FORAM UTILIZADAS	56
QUADRO 8 – EXEMPLO DE ESPECIFICAÇÃO DE <i>USER STORY</i> E TESTES DE ACEITAÇÃO PARA A FERRAMENTA HI-FY.....	61
QUADRO 9 – SEQUÊNCIA DE AÇÕES DE MANUTENÇÃO DA CONSISTÊNCIA AUI - MT	78
QUADRO 10 - OPERAÇÕES DE MANUTENÇÃO DA CONSISTÊNCIA AUI – MT	79
QUADRO 11 – FALHAS DE CONFORMIDADE DE FASTINTERFACE RELACIONADAS AO PADRÃO ISO 9241-14	87
QUADRO 12 – FALHAS DE CONFORMIDADE DE FASTINTERFACE RELACIONADAS AO PADRÃO ISO 9241-16	89
QUADRO 13 – FALHAS DE CONFORMIDADE DE FASTINTERFACE RELACIONADAS AO PADRÃO ISO 9241-17	90
QUADRO 14 – REQUISITOS FUNCIONAIS DO SISTEMA <i>CONSULTOR</i>	92
QUADRO 15 - PLANO DE AVALIAÇÃO.....	94
QUADRO 16 – RESULTADO DA EXECUÇÃO DAS ATIVIDADES PREVISTAS NO ROTEIRO DE TESTE.....	98

Lista de Tabelas

TABELA 1– TAXAS DE ADOÇÃO	91
TABELA 2– MENSURAÇÃO DO DESEMPENHO A PARTIR DOS INDICADORES QUANTITATIVOS	97
TABELA 3 – DADOS SUMARIZADOS A RESPEITO DAS CARACTERÍSTICAS DOS USUÁRIOS DE TESTE	104
TABELA 4–DADOS SUMARIZADOS COM RELAÇÃO À FACILIDADE DE USO DOS AMBIENTES FASTINTERFACE E TERESA	105
TABELA 5 – DADOS SUMARIZADOS COM RELAÇÃO À CONCORDÂNCIA DE AFIRMAÇÕES SOBRE O USO DOS AMBIENTES FASTINTERFACE E TERESA	106

Capítulo 1

Introdução

Este capítulo tem como finalidade apresentar o estado da arte a respeito do desenvolvimento de interface com usuário, em especial da abordagem baseada em modelos. Em seguida, apresenta-se o problema que este trabalho se propõe a resolver, bem como os objetivos e relevância desta dissertação e, por fim, a metodologia utilizada.

1.1. Cenário técnico-científico

Aplicações interativas são basicamente compostas por duas partes principais: o núcleo funcional, que contém as várias funções semânticas (ou métodos) das aplicações, e a Interface com Usuário (IU), que dá acesso a essas funções. Durante muito tempo, o foco de pesquisas na área de Engenharia de Software foi a construção de sistemas focados na funcionalidade, processamento lógico e organização de dados. A usabilidade era colocada muitas vezes, de maneira errônea, como uma questão da futura interface gráfica que poderia ser abordada após ser desenvolvida a parte principal da funcionalidade. Sabe-se, hoje, que a IU é um componente indispensável para qualquer sistema interativo, pois, determina quão fácil será a inserção de dados, a navegação entre as telas e o controle das funções semânticas do software. Além do mais, mesmo que o software tenha sido equipado com funções robustas e bem projetadas, se a IU for de baixa qualidade, essas funções podem ser inexploradas ou até mesmo inutilizadas ((Nielsen, 1993), (Ferre, 2001), (Coyette, 2007)).

Myers, Hudson e Pausch (2000), mostra que a codificação da IU pode representar entre **50% e 70%** do total de Linhas de Código (LOC, do inglês *Lines of Code*) da aplicação. Com relação ao ciclo de vida, ao tempo gasto no projeto, assim como à implementação e avaliação, representam, respectivamente, 45%, 50% e 37% do total do tempo de desenvolvimento, o que reporta a uma média de 44% do tempo total para o desenvolvimento da IU. Tendo em mente esses dados que remetem à importância do projeto da interface, a indústria e a comunidade científica têm dedicado significativo esforço na procura de novas técnicas que possam reduzir o esforço e o tempo de obtenção de uma interface com usuário com alto grau de usabilidade. Neste contexto, por exemplo, diversos editores gráficos, também conhecidos como “*builders*”, foram desenvolvidos a fim de auxiliar o processo de concepção de interfaces do usuário de maneira menos onerosa. Apesar de se mostrarem úteis para a construção de IU, essas ferramentas por si só não garantem a produção de uma interface de qualidade. Há necessidade de uma metodologia que possa guiar as equipes de projeto durante as diferentes etapas do ciclo de vida do desenvolvimento de IU para a obtenção de um produto com alto grau de usabilidade (Coyette, 2007).

Diversos grupos de pesquisa vêm investindo tempo e recursos na formalização de métodos de desenvolvimento de IU que levem em consideração aspectos de usabilidade. Nesse sentido, houve um estímulo para o uso da Abordagem Baseada em Modelos (*Model-Based Approach*), que consiste em estabelecer uma arquitetura que encoraje o uso intensivo de modelos como meio de gerar, de forma automatizada, seu código. Esta abordagem tem um largo contraste com abordagens tradicionais onde o software é codificado sem nenhum modelo ou especificação, e pouco contraste com as abordagens em que o software é especificado e manualmente codificado através dessas especificações (Limbourg e Vanderdonckt, 2004, p. 155). As informações contidas nos diversos modelos envolvidos são utilizadas para representar formalmente um agrupamento de conceitos, estruturas de representação, e uma série de primitivas e termos que podem ser usados para explicitamente capturar as várias formas de conhecimento sobre a IU e sobre sua aplicação interativa usando abstrações apropriadas. O uso deste tipo de abordagem traz grandes benefícios, como: (i) possibilidade de fácil automatização do processo através do uso de ferramentas específicas; (ii) consistência e reutilização; e (iii) possibilidade de desenvolvimento iterativo.

1.1.1. Desenvolvimento de Interfaces do Usuário Baseado em Modelos

Os processos de desenvolvimento de interface com usuário baseados em modelos fundamentam-se, tipicamente, na existência de modelos tais como **modelo da tarefa, do usuário, do domínio e da interação** e na definição de mapeamentos entre eles. Pode-se encontrar na literatura diversas propostas de *frameworks* que auxiliam a concepção de interface com usuário em uma abordagem baseada em modelos (Szekely, 1996; Stary, 2000; Griffiths *et. al.*, 2001). A princípio, a maioria dessas propostas tinha como objetivo a geração de interfaces do usuário através de um **modelo de domínio**. Atualmente, a maioria das abordagens guia seu desenvolvimento através do **modelo da tarefa**. Algumas destas propostas serão introduzidas nas sub-seções a seguir, apresentando os prós e contras da utilização de cada uma delas (Montero e López-Jaquero, 2006).

1.1.1.1 Desenvolvimento de Interfaces do Usuário Baseado no Domínio

Modelo de Domínio encapsula importantes entidades do domínio de uma aplicação, tais como, seus atributos, métodos e relacionamentos. Os elementos do modelo de domínio possuem informações relevantes para seleção dos elementos de interação que farão parte da IU. Exemplos de atributos de elementos do domínio são: o tipo de dado, o tamanho, o mínimo e o máximo valor, etc. Durante o processo de transformação que gera a interface final, o mapeamento é estabelecido pra definir, por exemplo, qual *widget* deve ser utilizada para representar um atributo do tipo inteiro para entrada de dados.

Muitas ferramentas dão suporte a esse tipo de desenvolvimento da IU, a saber: Janus (Balzert *et. al.*, 1996); OlivaNova (Molina, 2004); Teallach (Griffiths *et. al.*, 2001), para aplicação desktop; e, WebRadio (Ceri, Fraternali e Bongio, 2000) e VisualWade (Gómez, 2004), para aplicações web.

Essas ferramentas, que fazem uso do modelo de domínio para a geração da IU, produzem interfaces finais muito complexas, visto que os usuários podem ver muitos elementos de uma única vez. Da mesma forma, quando não se faz o uso de um modelo da tarefa, o diálogo entre as telas fica limitado e restrito, produzindo interfaces, de certa maneira, estáticas. Utilizar informações apenas do modelo de domínio não possibilita a detecção do

grau de relacionamento entre os atributos existente em uma determinada classe nem a importância dos elementos para a interface final.

1.1.1.2 Desenvolvimento de Interfaces do Usuário Baseado na Tarefa

Modelo da Tarefa especifica o que o usuário faz, como faz, e quando faz. Esta especificação descreve as tarefas que o usuário executa quando usa a aplicação e como essas tarefas se relacionam entre si. A maioria das abordagens de desenvolvimento de interface baseadas em modelos define o modelo da tarefa como artefato base a partir do qual, por transformações sucessivas, a interface é produzida. Desta forma, as informações provenientes do modelo da tarefa são exploradas para automaticamente ou interativamente derivar a estrutura navegacional (modelo de diálogo) e de apresentação de uma aplicação (Montero e López-Jaquero, 2006).

Para dar suporte a modelagem da tarefa, foram desenvolvidos vários formalismos tais como: MAD, TKS, TAOS (Medeiros *et. al.* 2000), CTT (Kieras e Polson 1985), TATAR (Schulungbaum e Elwert, 1996) e GOMS (Card, Moran e Newell, 1983); e, algumas ferramentas de suporte a esses formalismos entre as quais se pode destacar: CTTE (CTTE, 2002), que dá suporte ao formalismo CTT; e, iTAOS (Medeiros *et. al.* 2002), que auxilia a modelagem com o formalismo TAOS.

Ao se comparar o desenvolvimento baseado na tarefa com o baseado no domínio, pode-se destacar que o primeiro incorpora informações a respeito da forma como o usuário irá desempenhar uma tarefa através das várias telas do sistema, indicando os relacionamentos temporais e restrições entre essas tarefas. Assim, esse tipo de abordagem considera, de maneira mais enfática, aspectos de usabilidade, sobreposição de telas, agrupamento dos elementos de interação, entre outros. Já o desenvolvimento baseado no domínio dá destaque aos atributos das classes do domínio, aos tipos e validação de dados e na ligação dos objetos de negócio com os objetos interativos.

1.1.1.3 Prototipagem da Interface com usuário

O desenvolvimento baseado em modelos, como descrito anteriormente, está apoiado em dois aspectos:

- Modelagem da Interface com Usuário, o que representa um avanço significativo na área de Interface Humano-Computador.
- Transformação entre os Modelos, considerado bastante promissor na área de Engenharia de Software (ES) uma vez que serve como elo de ligação entre descrições abstratas do software e sua real implementação (Brown, 1997; Vanderdonckt e Berquim, 1999).

A transformação entre modelos obedece duas etapas: transformação modelo-para-modelo (M2M) e modelo-para-código (M2C) (Model-Driven Architecture Guide, 2003). A atividade de edição de código gerado é muito sensível a erros, visto que o entendimento do código é complexo e pode ferir os padrões que foram utilizados na construção baseada em modelos. Além do mais, alguns aspectos como usabilidade, consistência entre a IU e seus modelos, corretude e o benefício de estar livre de erros podem ser feridos (Abrahão, Iborra e Vanderdonckt, 2007; Pederiva *et. al.*, 2007).

Sob esse ponto de vista, se faz necessário o uso de alguma representação visual que possibilite a edição dos modelos ainda em fase de projeto. Essa visualização e edição podem ser feitas através de alguma técnica de prototipagem. A prototipagem consiste na criação de uma versão preliminar da futura interface (protótipo), de forma que tanto o usuário como o projetista possam identificar, antecipadamente, problemas ocorridos em seu projeto, tanto do ponto de vista funcional, como do ponto de vista de usabilidade.

Segundo Berkun (2000), os protótipos podem ser produzidos segundo dois modelos:

- Evolutivo: usado quando os requisitos do sistema não estão bem definidos. Assim, o protótipo evolui à medida que novos requisitos são descobertos;
- Descartável: usado quando os requisitos estão bem definidos e busca-se a validação dos mesmos;

A abordagem evolutiva vem ao encontro da necessidade de se agregar um processo sistemático de concepção e prevê o desenvolvimento da interface de maneira incremental.

Com relação às técnicas de prototipagem, estas podem ser divididas em termos de fidelidade, que é o grau de similaridade entre o protótipo e a interface final da aplicação, quais sejam:

- Técnicas *Low-Fi* (Baixa Fidelidade) - essa família de técnicas é geralmente usada no estágio de análise de requisitos para validar os requisitos com o usuário em abordagens centradas no usuário (Constantine e Lockwood., 1999). Sua construção

normalmente se dá através da técnica de desenho a mão livre ou utilizando ferramentas simples como lápis, papel e material de escritório.

- Técnicas *Me-Fi* (Média Fidelidade) - consistem na implementação computadorizada de uma aplicação limitada funcionalmente, contendo apenas as funções essenciais para avaliar alguns cenários específicos. As características dos protótipos de média fidelidade consistem na união das características positivas dos protótipos de baixa fidelidade (fácil e rápido de construir e editar) com a possibilidade de automatização e simulação.
- Técnicas *Hi-Fi* (Alta Fidelidade) - essas técnicas são usadas para criação de versões preliminares de uma interface com usuário com um aceitável grau de qualidade. Esses tipos de técnicas produzem um protótipo de interface com usuário que é próximo à futura interface final. Esses protótipos são representações executáveis (código), construídos com o uso de uma linguagem de programação (ou ferramentas de apoio) e contêm as principais funcionalidades presentes na interface do futuro sistema. Eles definem, claramente, os aspectos estéticos (padrão, fonte, cor, tamanhos de botões, etc.) e os componentes de navegação. Esse tipo de protótipo está presente nas diversas metodologias de concepção de interface baseada em modelos como último artefato a ser construído antes da interface final.

O uso isolado de algum tipo de protótipo traz algumas melhorias tanto para o processo de concepção de interface quanto para o produto em si (a interface), como por exemplo:

- (i) o uso de protótipos de baixa fidelidade, quando feito no início do processo de desenvolvimento, melhora a comunicação entre os membros da equipe de projeto devido ao uso de uma linguagem visual simples que não requer conhecimento técnico ou habilidade específica (melhoria no processo). A melhoria na comunicação facilita o entendimento dos requisitos e do domínio do problema por parte da equipe de projeto. O resultado é uma interface que corresponde aos requisitos especificados (melhoria no produto);
- (ii) o uso de protótipos de média fidelidade, quando feito em uma fase intermediária do processo de desenvolvimento, antecipa o processo iterativo entre o usuário e a interface em desenvolvimento, possibilitando a identificação prematura de falhas de usabilidade ou insatisfações por parte do usuário (melhoria no processo). Como resultado, temos um produto final com um número reduzido de falhas sem requerer

investimento de tempo e recurso em re-trabalho para a correção das falhas em uma fase posterior do processo de desenvolvimento da interface com usuário (melhoria no produto);

- (iii) o uso de protótipos de alta fidelidade, quando feito no final do processo de desenvolvimento, permite que treinamentos sejam realizados a fim de tornar o produto, que ainda não é o final, familiar aos seus futuros usuários (melhoria no processo). Como resultado, ao disponibilizar o produto final, seus usuários poderão fazer uso efetivo do mesmo, uma vez que a curva de aprendizado já foi anteriormente minimizada (melhoria no produto).

No entanto, embora as melhorias no processo e no produto sejam evidentes com o uso de protótipos no processo de concepção de interfaces do usuário, seu uso isolado não estimula a adoção da abordagem baseada em modelos, visto que:

- (i) como o uso de protótipos de baixa fidelidade se dá no início do processo e sua representação ainda é rudimentar, a sua utilização exclusiva não possibilita a total previsibilidade de como será a interface final;
- (ii) o uso exclusivo de protótipos de média fidelidade, apesar de melhorar a previsibilidade e possibilitar a avaliação da interface final (aspectos visuais e de navegação), ainda exige a realização de testes em cima de modelos declarativos durante as fases iniciais e finais do processo; e por fim,
- (iii) o uso restrito de protótipos de alta fidelidade adia consideravelmente a possibilidade de prever a interface final e impõe que avaliações sejam feitas em cima de modelos declarativos nas fases iniciais e intermediárias do processo.

Com foi visto, o uso isolado destes tipos de protótipos adotados dificulta a aplicação de técnicas de avaliação da usabilidade que, em algum momento, ainda necessitará ser feita em cima de modelos declarativos. O uso de protótipos de baixa e de alta fidelidade, mesmo que em conjunto, dificulta que técnicas de avaliação da interface final sejam eficazes visto que não é possível garantir que exista uma correlação direta (mapeamento) de sua aplicação em protótipos de baixa e de alta fidelidade devido a grande distância no nível de fidelidade existente entre eles.

1.1.2. Ambientes de Desenvolvimento de Interfaces do Usuário Baseado em Modelos

O grande ganho de se trabalhar com modelos é a possibilidade de agilização do desenvolvimento através de geradores automáticos. Basicamente, visa-se definir algoritmos que façam mapeamento entre modelos que representam tarefas realizadas por um usuário em um **modelo de interface** (Libório *et. al.*, 2005). Como forma a dar suporte a esse tipo de desenvolvimento de IU, foram criados ambientes que têm como objetivo auxiliar os projetistas em atividades que envolvem a construção e ligação de uma coleção de modelos: são os denominados Ambientes de Desenvolvimento de Interface com Usuário Baseado em Modelos (MB-IDEs) (Calvary, 2007).

Para apresentar os MB-IDEs existentes, é preciso que todos se encaixem em uma mesma referência. Foram selecionados alguns esforços representativos e significantes feitos nos ambientes existentes que suportam a abordagem baseada em modelos. Esses esforços estão apresentados de acordo com um *framework* de referência que representa os vários níveis de abstração e modelos que podem ser utilizados em um processo de desenvolvimento de interface com usuário. O desenvolvimento baseado em modelos e na tarefa do usuário está estruturado basicamente em quatro níveis de abstração, definidos pelo *Framework* de Referência Unificado para Desenvolvimento de Interface Multi-Plataforma (Calvary, Coutaz e Thevenin, 2003)¹, referido neste trabalho como *framework Cameleon*, o qual pode ser observado na Figura 1.

O *framework Cameleon* está dividido em três partes: **Modelos Ontológicos**, **Processo de Desenvolvimento** e **Processo de Execução**. Modelos Ontológicos são meta-modelos que são independentes de qualquer domínio ou sistema interativo e descrevem um determinado problema. Os modelos que representam o domínio do problema e o contexto de uso servem como entrada do processo de desenvolvimento, já os modelos observados guiam o processo de adaptação em tempo de execução. O processo de desenvolvimento explica como produzir uma Interface com usuário para cada contexto de uso típico. Engloba os processos de **Tarefas e Conceitos**, que descreve as várias tarefas as quais serão realizadas e os conceitos que serão requeridos para que estas tarefas possam ser executadas; **Interface Abstrata do Usuário** (do inglês **AUI**), que define os elementos de interação abstratos e o diálogo entre eles; **Interface Concreta do Usuário (CUI)**, que faz uso de elementos de

¹ Muitos trabalhos a respeito de Coyette, Kieffer e Vanderdonckt. (2007), Montero e López-Jaquero (2006), Limbourg e Vanderdonkt (2004), Griffiths *et. al.*(2001) utilizam o termo *framework Cameleon* para designar o *framework* de Referência Unificado.

interação concretos (*widgets*) para um contexto de uso particular; e, **Interface Final com o Usuário (FUI)**, que é tipicamente o código da interface em alguma linguagem interpretada ou compilada.

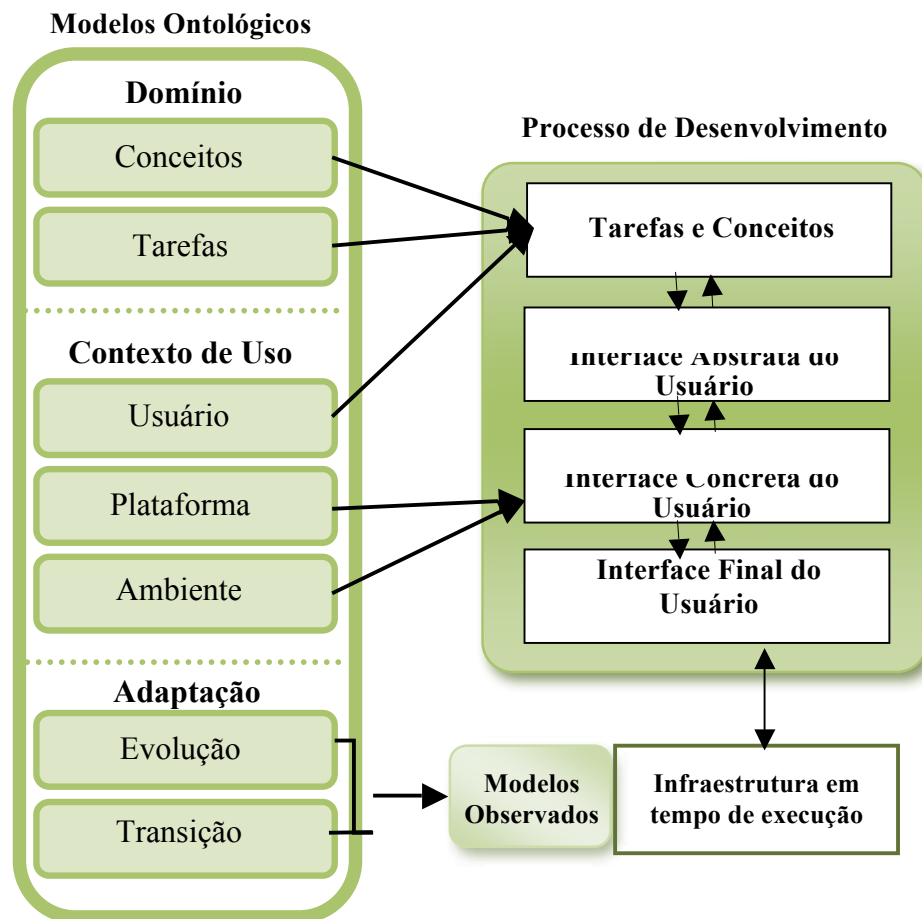


Figura 1- Framework Cameleon

Vários são os ambientes que dão suporte a este tipo de abordagem, entre os quais podemos destacar Mastermind (Szekely, 1996), TADEUS (Stary, 2000), Teallach (Griffiths *et. al.*, 2001), TransformiXML (Limbourg e Vanderdonckt, 2004, p. 155) e TERESA (Mori, Paternò e Santoro, 2004). Estes ambientes exploram informações estruturais do modelo tarefa, como relacionamentos temporais, para gerar uma “série de ativações” (do inglês *activation set*), que é usada para gerar automaticamente o diálogo e os elementos do modelo da interação. Cada um destes MB-IDES dão suporte a métodos e formalismos diferentes e se propõem a gerar IU para diferentes contextos de uso. O Quadro 1 apresenta uma visão dos MB-IDES sob o prisma do framework *Cameleon*.

Quadro 1 - Visão dos MB-IDES pelo prisma do *framework Cameleon*

	Conceitos e Modelo da Tarefa	Interface Abstrata	Interface Concreta	Interface Final
Mastermind	Modelo da Tarefa Hierárquico	Descritores	Protótipo Low-Fi	Código C++
TADEUS	TATAR	Script	Protótipo Hi-Fi	-
Teallach	Modelo da Tarefa Hierárquico	Script	Protótipo Hi-Fi	Código Java Swing
TransformiXML	CCT	Representação visual rudimentar (não editável)	Protótipo Hi-Fi	XHTML
Teresa	CCT	Descritores Lógicos Múltiplos	Protótipo Hi-Fi	HTML

Percebe-se, observando o quadro, que os MB-IDES tardam em utilizar técnicas de prototipagem em seus processos e preconizam apenas o uso de um tipo de fidelidade. O uso de alguma técnica de prototipagem, geralmente prototipagem de alta fidelidade, é realizado apenas no nível de abstração CUI. Para MB-IDES, nos níveis anteriores, não há nenhuma técnica de prototipagem envolvida, de modo que com que os projetistas têm que lidar com informações especificadas em descritores ou arquivos de script.

1.1.3. Antecipação de Representação Visual em Abordagens Baseadas em Modelos

Como visto, os principais problemas encontrados em abordagens de desenvolvimento de interface com usuário baseada em modelos estão relacionados às dificuldades na sua adoção por parte dos projetistas devido a dois aspectos importantes citados por Myers, Hudson e Pausch (2000):

- (i) o uso de modelos não está associado a uma representação visual que possibilite vislumbrar a interface final que será gerada a partir dos modelos utilizados. Essa característica faz com que a interface final do usuário seja imprevisível durante todo, ou quase todo, o seu processo de desenvolvimento;
- (ii) o uso exclusivo de modelos declarativos nas fases iniciais e intermediárias do processo implica na necessidade de se aplicar técnicas de avaliação da interface final em cima desse tipo de modelo, o que impossibilita a avaliação de aspectos

visuais e de navegação simultaneamente, o que seria adiado para estágios finais do processo de desenvolvimento.

No intuito de resolver estes problemas, o trabalho de Montero e López-Jaquero (2006), sobre o ambiente IDEALXML, se destacou no sentido de oferecer representações visuais ao projetista no desenvolvimento de interface com usuário em uma abordagem baseada em modelos. O ambiente IDEALXML se propõe a oferecer um protótipo de alta fidelidade no nível de abstração AUI e, para tal, fornece as funções de visualização e simulação. No entanto, no nível abstrato, ainda não existe informações suficientes de modo a gerar uma interface próxima ao que o usuário deseja, como é o propósito de um protótipo de alta fidelidade. Não é possível detectar os objetos concretos e nem tratar a questão da disposição dos elementos interativos na tela. A grande dificuldade de utilização deste ambiente está no fato de não apresentar uma técnica capaz de oferecer ao projetista a flexibilidade de edição de maneira rápida e eficaz para uma fase intermediária do processo. A utilização de um protótipo de alta fidelidade remete ao fato que a interface está em seu estágio mais próximo à interface final do sistema, até mesmo incorporando estilo, o que também não é possível para este caso.

Por sua vez, o trabalho de Coyette, Kieffer e Vanderdonckt. (2007) apresenta um estudo sobre o uso de múltiplas técnicas de prototipagem em uma abordagem baseada em modelos, o que o autor denomina *multi-fidelidade*. Nesta abordagem, técnicas de baixa, média e alta fidelidade são utilizadas no nível de abstração CUI a fim de serem utilizadas em momentos diferentes, dependendo do grau de detalhamento que se deseja trabalhar. Este processo é suportado por uma ferramenta denominada SketchXML. O uso deste tipo de técnica, no entanto, não favorece a antecipação de representação visual em uma fase intermediária do processo de desenvolvimento da IU baseado em modelos, fato que corrobora para a permanência dos problemas supracitados.

MEDITE (Rodrigues, Lula Jr. e Suárez, 2005), uma metodologia baseada em modelos e na tarefa do usuário, define em seu fluxo um processo de prototipagem de média fidelidade em um nível intermediário (nível AUI) e pressupõe o uso de um protótipo de alta fidelidade ao final do processo de concepção (nível CUI). MEDITE tem como suporte computacional as ferramentas iTAOS (Medeiros, Lula Jr. e Cordeiro, 2002), para modelagem da tarefa; MAPA (Rodrigues, 2005), para mapeamento entre o modelo da tarefa e o modelo de interface abstrata do usuário (AUI); e, SMILE (Aguiar *et. al.*, 2007), para geração de protótipo de média fidelidade (AUI). Porém, mesmo preconizando o uso de duas técnicas de prototipagem em uma abordagem baseada em modelos, MEDITE não oferece uma evolução entre os

diferentes protótipos utilizados. A prototipagem de alta fidelidade não apresenta nenhuma técnica formal, o que acarreta em manipulações subjetivas do projetista, fato este que não favorece a resolução dos problemas elencados por Myers, Hudson e Pausch (2000).

1.2. Definição do problema

O uso conjunto de protótipos de baixa, média e alta fidelidade em fases diferentes (e adequadas) do processo de desenvolvimento da interface com usuário pode facilitar a aceitação do uso deste tipo de abordagem, visto que:

- (i) a transição entre diferentes níveis de representação visual durante todas as fases do processo (inicial, intermediária e final) pode permitir que os projetistas tenham um certo grau de previsibilidade da interface final, que será gerada ao final do processo, em suas diferentes fases;
- (ii) o constante uso de representações visuais da interface pode permitir que técnicas de avaliação de interface sejam empregadas em todas as fases do processo, minimizando (ou até mesmo evitando) a necessidade de realizar avaliações de interface em cima de modelos declarativos.

No entanto, como visto na seção anterior, mesmo com a definição de diferentes técnicas de prototipagem em diferentes estágios no seu processo de desenvolvimento, os protótipos em MEDITE não são utilizados de maneira evolutiva.

Para que haja prototipagem evolutiva em MEDITE, é preciso garantir que:

- regras de transformação entre MT e AUI sejam passíveis de englobar as diferentes formas de entendimento do problema;
- protótipos editáveis e simuláveis em um nível intermediário (nível AUI) exijam um modelo de diálogo capaz de prover a navegabilidade entre os diferentes elementos;
- a criação de protótipos de alta fidelidade seja feita com técnicas de transformação AUI – CUI que levam em consideração informações dos protótipos de média fidelidade;

- todas as etapas sejam encaradas de maneira integrada com a utilização de um modelo de interface e uma linguagem de especificação.

Dado o exposto, levanta-se as seguintes questões dentro do escopo de utilização da metodologia MEDITE:

- É possível criar métodos para geração de múltiplos protótipos de maneira evolutiva?
- O uso de múltipla prototipagem evolutiva facilita o processo de concepção de interface com usuário?
- É possível criar um ambiente que integre e suporte esse tipo de abordagem?

1.3. Objetivos do trabalho

1.3.1. Objetivo geral

O objetivo geral deste trabalho é propor uma abordagem de desenvolvimento de interfaces do usuário baseada em modelos e múltipla prototipagem evolutiva. Para que seja possível a validação do trabalho, a abordagem deverá refletir em uma metodologia de concepção de interfaces do usuário, denominada MEDITE, que foi escolhida por agrupar seus processos de acordo com o *framework* de referência unificado para desenvolvimento de interfaces com o usuário e prover o uso de diferentes técnicas de prototipagem. Deve-se disponibilizar, da mesma forma, um ambiente de desenvolvimento que possa refletir a abordagem proposta.

1.3.2. Objetivos específicos

- Localização e definição de diferentes técnicas de prototipagem em MEDITE sob a perspectiva dos diferentes níveis de abstração propostos no *framework* de referência *Cameleon*;
- Ajuste no conjunto de regras de transformação do modelo da tarefa para o modelo de interface abstrata do usuário (modelo conceitual parcial da interação) definido por MEDITE a fim de abranger o mapeamento de diferentes formas de modelagem;

- Definição de um processo formal de construção de protótipos de alta fidelidade em MEDITE que leve em consideração informações provenientes de protótipos de média fidelidade;
- Definição e implementação da ferramenta Hi-Fy para automatizar o processo de geração automática, edição e simulação de protótipos de alta fidelidade e geração da interface final;
- Definição de uma linguagem de especificação no formato XML para persistência do modelo da interface com usuário em seus diferentes níveis de abstração;
- Definição e implementação de um ambiente (FastInterface) que integre as ferramentas até então desenvolvidas para dar suporte à metodologia MEDITE em seus diferentes níveis de abstração;
- Validação da abordagem proposta através da aplicação de uma avaliação multidimensional.

1.4. Relevância

A disponibilização de um ambiente único e integrado de desenvolvimento de interfaces do usuário baseado em modelos utilizando múltipla prototipagem pode evidenciar que:

- o tipo de abordagem traz benefícios com relação à produtividade, considerando tempo e esforço para projeto e desenvolvimento de interface;
- o projetista terá menor esforço para utilizar uma ferramenta que tenha uma infraestrutura mais próxima à sua realidade de desenvolvimento, que alie os requisitos de usabilidade com o poder do uso de modelos.
- a manutenção da consistência entre os diversos modelos previstos na metodologia MEDITE seja realizada em tempo real de projeto.

Há ainda alguns aspectos relevantes, entre os quais podemos destacar:

- Fomentar a concepção e implementação de métodos e modelos de transformação de protótipo de média fidelidade para o protótipo de alta fidelidade.
- Fomentar a especificação de linguagem para geração automática da interface final levando em consideração a plataforma Web.

- Consolidação do GIHM da UFCG (Grupo de Interface Homem-Máquina da Universidade Federal de Campina Grande), pois com a expansão da utilização da abordagem proposta e a utilização de MEDITE com suas ferramentas de suporte, o GIHM adquire maior visibilidade entre a comunidade interessada em concepção de interfaces.

1.5. Metodologia de desenvolvimento

A metodologia adotada na realização deste trabalho de dissertação pode ser decomposta nas seguintes etapas, a saber:

1. Revisão bibliográfica, a fim de identificar o estado da arte das técnicas de concepção de interface baseadas em modelos e abordagens que utilizassem protótipos de diferentes níveis de fidelidade;
2. Análise do *framework Cameleon*, visando tê-lo como referência para identificar as lacunas nas técnicas de concepção elencadas no estado da arte;
3. A partir das lacunas identificadas no levantamento do estado da arte, foi feita uma nova proposta de *framework* que preenche algumas delas, o *framework* FastInterface;
4. Estudo da metodologia MEDITE, a fim de dotá-la de uma abordagem de desenvolvimento baseada em modelos e múltipla prototipagem evolutiva, ou seja, adequá-la ao *framework* FastInterface;
5. Elaboração de uma técnica de geração automática de protótipos de alta fidelidade baseado nos protótipos de média fidelidade em MEDITE, disponibilizando uma ferramenta denominada Hi-Fy;
6. Planejamento, projeto e implementação do ambiente de desenvolvimento de interface com usuário baseado em modelos e múltipla prototipagem evolutiva que refletisse os processos definidos em MEDITE, esse ambiente foi denominado FastInterface;
7. Validação da abordagem e do ambiente de desenvolvimento de interfaces FastInterface foi feita por meio de um estudo de caso com testes de usabilidade realizados por usuários, frente a outra abordagem;

8. Realização da Inspeção de Conformidade do ambiente FastInterface às Partes 14 – *Diálogo via menus*, 16 – *Diálogo via manipulação direta* e 17 – *Diálogo via Formulários* do Padrão ISO-9241.
9. Todas as etapas descritas acima foram documentadas e o relato dessas atividades constitui esta dissertação.

1.6. Estrutura da dissertação

A dissertação está organizada em 6 capítulos. No **Capítulo 1**, é apresentado o cenário técnico-científico, a definição do problema, os objetivos do trabalho, sua relevância e a metodologia utilizada.

No **Capítulo 2**, é descrita a abordagem de desenvolvimento de interface com usuário baseada em modelos e múltipla prototipagem evolutiva tendo sua instanciação na metodologia MEDITE.

No **Capítulo 3**, trata-se do projeto e implementação de uma ferramenta de suporte à geração automática, edição e simulação de protótipos de alta fidelidade em MEDITE. Essa ferramenta se adequará à fase CUI, de *Cameleon*, a fim de prover a evolução de protótipos de média para de alta fidelidade.

No **Capítulo 4**, apresenta-se o projeto de um ambiente integrado de desenvolvimento de interface com usuário, FastInterface, que apóia a abordagem proposta. São descritos os aspectos de implementação da ferramenta e integração entre os diferentes módulos. Da mesma forma, descreve-se os mecanismos de transformação e persistência entre os modelos envolvidos na concepção da interface, bem como a manutenção da consistência entre eles.

No **Capítulo 5**, apresenta-se a análise dos resultados obtidos com a utilização de FastInterface em um estudo de caso para validação do ambiente frente a outra abordagem. Também foi realizada, dentro do escopo deste capítulo, uma inspeção de conformidade com as Partes (14 - *Diálogos via menus*, 16 - *Diálogo via manipulação direta* e 17 – *Diálogo via formulários*) do Padrão ISO-9241.

Por fim, no **Capítulo 6** são apresentadas as conclusões da pesquisa e sugestões para trabalhos futuros.

Capítulo 2

Abordagem de Desenvolvimento de Interface com usuário Baseada em Modelos e Múltipla Prototipagem Evolutiva

Este capítulo tem como objetivo descrever a abordagem de desenvolvimento de interface com usuário baseada em modelos e múltipla prototipagem proposta neste trabalho. Para isto, será apresentada MEDITE, metodologia a qual será estendida a fim de incorporar a abordagem proposta. Da mesma forma, serão evidenciadas as extensões realizadas nos processos e fases da metodologia em questão.

2.1. MEDITE

MEDITE é uma metodologia baseada na tarefa do usuário, orientada a modelos, iterativa e incremental para auxiliar projetistas no processo de concepção de interfaces do usuário. O processo de desenvolvimento definido por MEDITE inclui a geração de protótipos de média e de alta fidelidade para representação visual da AUI e CUI, respectivamente, conforme *Cameleon*.

O fluxo completo de MEDITE possui quatro fases, quais sejam: (i) fase 1, modelagem da tarefa; (ii) fase 2, geração da especificação conceitual parcial da interação; (iii) fase 3, geração da especificação total da interação; e, (iv) fase 4, geração da interface final, como evidenciado na Figura 2.

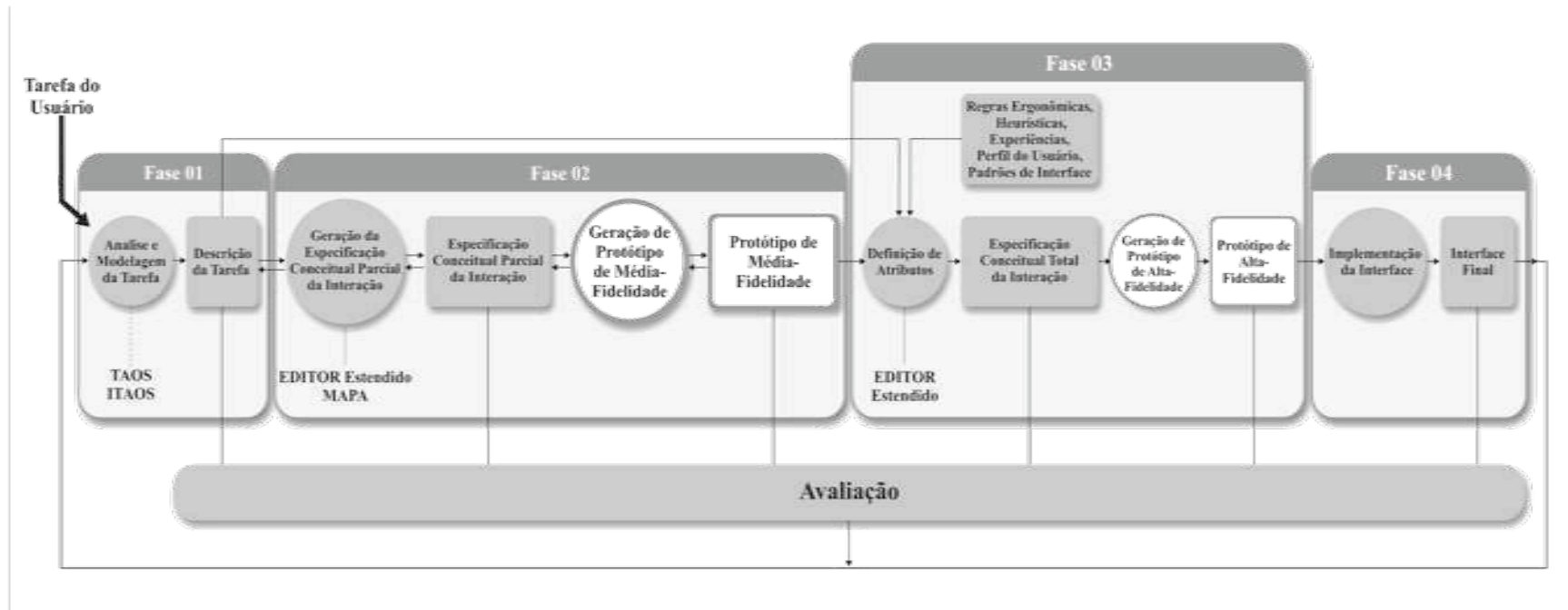


Figura 2- Fluxo de MEDITE
(Fonte: Aguiar, 2007)

A fase 1 (Tarefa e Conceito - em *Cameleon*) define o processo de *análise e modelagem da tarefa do usuário* e para tanto utiliza o formalismo TAOS (*Task and Action Oriented System*) (Medeiros, Kafure e Lula Jr., 2000). Esta fase tem o suporte computacional da ferramenta iTAOS (*interface TAOS*) (Medeiros, Lula Jr. e Cordeiro, 2002) que permite representar a tarefa do usuário a partir de uma árvore hierárquica de Tarefas, Sub-tarefas e Ações. A saída de iTAOS é um arquivo XML com a descrição da tarefa do usuário segundo o formalismo TAOS.

A fase seguinte (*fase 2*), geração da especificação conceitual parcial da interação (modelo de interface abstrata do usuário), faz o mapeamento dos elementos do modelo de tarefa em elementos do modelo da interação (Suárez, Lula Jr. e Barros, 2004). A entrada desse processo é o arquivo XML contendo a descrição da tarefa gerado pela ferramenta iTAOS (*interface TAOS*), a saída desse processo é computado a partir dos algoritmos de mapeamento, denominado MAPA (*MAPping Algorithm*), contendo uma descrição conceitual parcial da interação segundo o modelo EDITOR Estendido (Rodrigues, 2005). Essa descrição parcial da interação define os elementos de apresentação (Espaços, Visões e Objetos de interação) e o diálogo entre eles (Statecharts) extraídos da descrição da tarefa, onde: objetos de interação são quaisquer elementos que possibilitam que o usuário interaja diretamente com o sistema. Posteriormente, há um processo de geração de esboço de interface, na forma de protótipos de média fidelidade, através da especificação conceitual parcial da interação e regras de detecção automática dos tipos de objetos de interação pelo uso dos operadores da tarefa. Este processo dispõe de uma ferramenta chamada SMILE (*Sketch Manipulation Integrated with Less Effort*).

A próxima fase da metodologia (*fase 3*) prevê um refinamento da especificação conceitual parcial da interação, incorporando conhecimentos ergonômicos e heurísticas de projeto, para a construção de uma especificação conceitual completa da interação. Esta fase carece de formalização e ainda não possui suporte computacional.

A fase final da metodologia (*fase 4*) consiste na geração de um protótipo executável da interface, o que atualmente também carece de suporte computacional para sua realização.

Os processos previstos em MEDITE, sob o prisma de *Cameleon*, serão detalhados com maior riqueza de detalhes nas sub-seções a seguir.

2.1.1. Tarefas e Conceitos em MEDITE

A modelagem da tarefa permite a descrição dos objetivos dos usuários e dos meios utilizados no processo de realização de uma determinada tarefa, segundo o formalismo TAOS. A tarefa deve ser descrita em termos de objetivos, procedimentos, objetos, decomposição em sub-tarefas, restrições, etc.. Este processo tem o suporte da ferramenta iTAOS que é dividida em dois módulos: TAME (Cordeiro, 2003), parte funcional da ferramenta; e, GRAPH (Medeiros, 2003), que representa a parte gráfica do formalismo. iTAOS propõe um modelo de representação hierárquico de tarefas e sub-tarefas que considera tanto o comportamento estático quanto o comportamento dinâmico de um dado domínio através de dois tipos de entidades ou conceitos: os conceitos estáticos (objetos, métodos e situações) e os conceitos dinâmicos (processos, ações e planos).

A descrição da tarefa é persistida através de dois arquivos XML: um obedece uma DTD específica que reflete as informações das tarefas do usuário através de regras previstas pelo módulo TAME e o formalismo TAOS; e, o outro, com a descrição gráfica dos nodos da árvore hierárquica das tarefas conforme especificação do *framework* gráfico JHotDraw 6.0 (JHotDraw, 2004), como mostra a Figura 3.

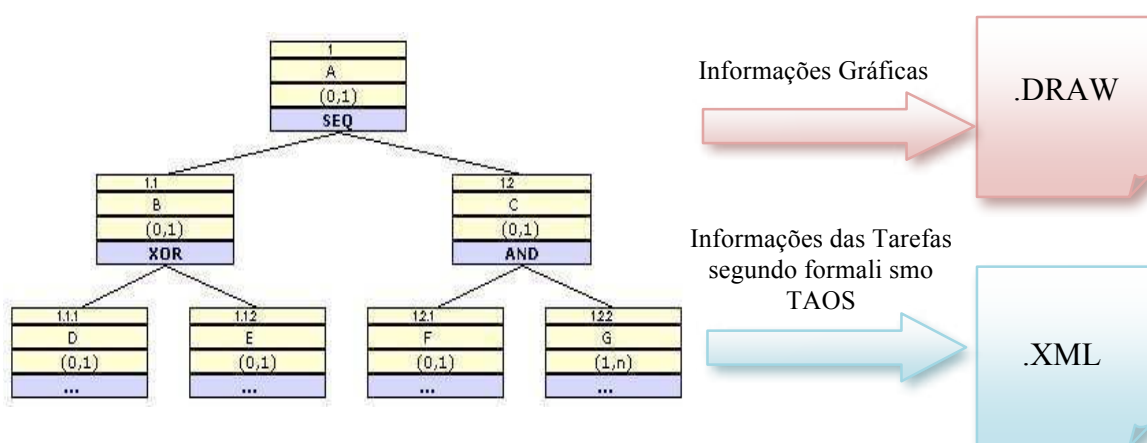


Figura 3 – Persistência da ferramenta iTAOS

2.1.2. Interface Abstrata do Usuário - Especificação Conceitual Parcial da Interação

MAPping Algorithm (MAPA) é uma ferramenta que tem como objetivo fazer o mapeamento entre os elementos presentes no modelo da tarefa em elementos do modelo da interação (especificação conceitual parcial da interação) segundo o Modelo Editor Estendido que contempla aspectos de apresentação e de navegação/diálogo. Para que esta operação aconteça é necessária a instanciação de um modelo intermediário, denominado modelo de roteiro. MAPA dispõe de um mecanismo de rastreabilidade entre os modelos da tarefa e da interação que permite que mudanças realizadas no modelo da interação sejam automaticamente refletidas no modelo da tarefa e vice-versa, mantendo os modelos sempre coerentes. Para que a manutenção da coerência seja possível, um conjunto de regras descritas por (Rodrigues, Lula Jr. E Suarez, 2005) devem ser aplicadas aos modelos. A Figura 4 ilustra o mecanismo para a manutenção da coerência entre os elementos dos modelos envolvidos.

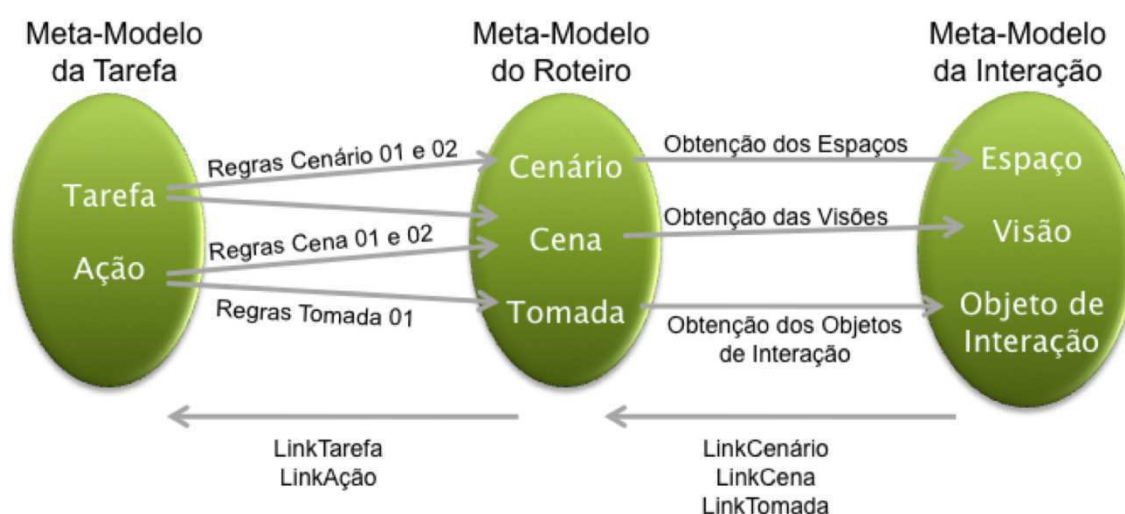


Figura 4 – Rastreabilidade entre os modelos da tarefa, roteiro e interação

O mapeamento entre os modelos acontece, a partir da aplicação de um conjunto de regras, em duas etapas:

- elementos do modelo da tarefa mapeados em elementos do modelo de roteiro;
- elementos do modelo de roteiro mapeados em elementos do modelo da interação.

Onde:

- Elementos do Modelo da Tarefa:
 - Tarefa: é uma seqüência de sub-tarefas ou ações organizadas de maneira a atingir um objetivo.
 - Ação: é uma ação elementar, atômica.
- Elementos do Modelo de Roteiro:
 - Cenário: elemento intermediário que faz o elo de ligação entre uma tarefa, no modelo da tarefa, e a representação do espaço do modelo da interação.
 - Cena: elemento intermediário que faz o elo de ligação entre uma tarefa ou ação, no modelo da tarefa, e a representação da visão do modelo da interação.
 - Tomada: elemento intermediário que faz o elo de ligação entre uma tarefa ou ação, no modelo da tarefa, e a representação do objeto de interação do modelo da interação.
- Elementos do Modelo da Interação:
 - Espaço: define a visualização e manipulação de um conjunto de visões em um quadro (ou janela)
 - Inicial: representa a tela inicial da aplicação
 - de Direcionamento: provê a navegação de um espaço até outros espaços de interação
 - de Interação: pelo menos um de seus elementos está associado a uma tarefa elementar da aplicação
 - Visão: define o modo com é reagrupamento e a visualizado um conjunto de objetos de interação
 - Objeto de Interação: define uma forma de visualizar e manipular um objeto da aplicação a partir de objetos interativos de uma biblioteca virtual.

2.1.2.1. Aspectos de Apresentação

O modelo de roteiro é obtido através da aplicação de um conjunto de regras que estão descritas na Quadro 2.

Quadro 2– Regras de Mapeamento do Modelo da Tarefa (MTa) para Modelo de Roteiro (MRo)

Regra	Descrição da Regra
Regra Tomada01	Se a Tarefa é uma <Ação> (MTa) defina uma <Tomada> (MRo) associada à Tarefa
Regra Cena01	Se a Tarefa é uma <Tarefa> (MTa) cuja decomposição apresente pelo menos uma <Ação> (MTa), defina uma <Cena> (MRo) associada à Tarefa
Regra Cenário01	Se a Tarefa é uma <Tarefa> (MTa) cuja decomposição apresente pelo menos uma <Tarefa> (MTa) que esteja associada a uma <Cena> (MRo), defina um <Cenário> (MRo) associado à Tarefa
Regra Cenário02	Se a Tarefa é uma <Tarefa> (MTa) que está associada a uma <Cena> (MRo) e pelo menos um de seus irmãos é uma <Tarefa> (MTa) associada a um <Cenário> (MRo), associe à Tarefa um <Cenário> (MRo)
Regra Cena02	Se a Tarefa é uma <Ação> (MTa) que seja filha de uma <Tarefa> (MTa) que esteja associada a um <Cenário> (MRo), associe à Tarefa uma <Cena> (MRo)
Regra Tomada02	Se a Tarefa é uma <Tarefa> (MTa) que não seja filha da Tarefa Raiz (MTa) e que esteja associada a um <Cenário> (MRo) e a uma <Cena> (MRo), associe esta Tarefa a uma Tomada (MRo) (Será no Modelo de Interação um Objeto de Link)
Regra Normalização	Se a Tarefa T, que não seja a raiz, possui pelo menos uma filha associada com a um <Cenário>, uma <Cena> e a um <Objeto de Interação>, associe T a um <Cenário>, uma <Cena> e a um <Objeto de Interação>

O modelo conceitual parcial da interação (MIO) é obtido através de mapeamento direto entre os elementos do modelo de roteiro e de interação, os quais: Cenário (MRo) vira Espaço (MIO); Cena (MRo) vira Visão (MIO); e, Tomada (MRo) vira Objeto de Interação (MIO).

2.1.2.2. Aspectos de Navegação

A representação da navegação (diálogo) do modelo de interação é obtida através de uma máquina de estados (*statecharts*) (Rodrigues, 2005) a partir da qual é possível definir os aspectos relacionados ao comportamento da futura interface com usuário, por exemplo: a navegação entre os diferentes Espaços da AUI, a necessidade de habilitar/desabilitar um conjunto de Objetos de Interação ou Visões em um determinado momento, a necessidade de definir uma ordem para a execução de ações quando o usuário interage com a interface, etc. (Aguiar, 2007).

Basicamente os elementos principais dos *statecharts* são: Estados, Transições e Ações. Os *Estados* são condições de existência distinguíveis que persistem por um período de tempo. *Transições* são a maneira como os objetos mudam de estados em resposta a um evento. *Ações* são comportamentos atômicos, ou operações, processados em uma máquina de estados quando se entra ou se sai de um evento.

O componente *Diálogo* do modelo EDITOR Estendido é obtido de elementos do modelo da tarefa (TAOS) e do componente Apresentação do modelo EDITOR. A partir do componente de apresentação, identificam-se os estados que compõem o *statechart*. Os eventos e as transições possíveis são adquiridos do modelo da tarefa através de alguns de seus atributos.

Identificação dos Estados do *Statechart*

Os *estados* do *statechart* são obtidos através dos elementos do componente *Apresentação* do modelo da interação. O modelo da interação está representado sob a forma de árvores EDITOR, portanto, de cada árvore será obtido um pequeno trecho do *statechart* que representa a interface do sistema. O elemento *Espaço* será associado a um *estado composto* (super estado). Cada elemento *Visão* será, também, associado a um *estado composto*. Este *estado composto* (associado ao elemento *Visão*) será um sub-estado do estado associado ao elemento *Espaço*. Cada elemento *Objeto de Interação* será associado a um *estado simples*, sendo este um sub-estado do *estado composto* associado ao elemento *Visão*.

A Figura 5 ilustra a representação dos elementos de interação como estados do *statechart*. O estado maior (laranja) representa um *Espaço*, composto por dois sub-

estados. Estes sub-estados representam as *Visões* (beges), que são compostos por estados simples (verdes), representando os *Objetos de Interação*.

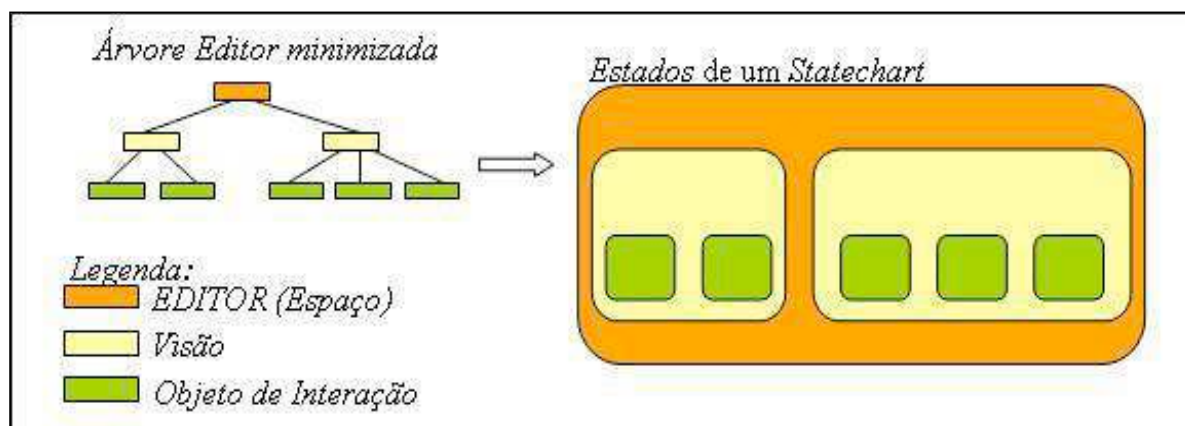


Figura 5 - Obtenção dos estados de um *Statechart* a partir do modelo de interação (Rodrigues, 2005)

Identificação dos Eventos e Transições do *Statechart*.

Os eventos que poderão disparar transições são eventos gerais, tais como, eventos de *realização* ou *não-realização* de uma condição, uma vez que não está se lidando com objetos concretos (*textfield*, *combobox*, etc..), o que não remete aos tipos de eventos que cada componente em específico pode disparar.

* Eventos

No modelo da tarefa existe um atributo *ocorrência*² de realização de uma determinada tarefa, que indica os possíveis eventos para cada tipo de elemento do modelo EDITOR Estendido.

O Quadro 3 apresenta a relação entre a ocorrência e os eventos possíveis para cada elemento do modelo da interação.

² A ocorrência pode assumir os valores: (0,0), não deve ser realizada nenhuma vez; (0,1), pode não ser realizada ou realizada apenas uma vez; (0,n), pode não ser realizada ou realizada inúmeras vezes; (1,1), deve ser realizada uma única vez; e, por fim, (1,n) deve ser realizada no mínimo uma vez.

Quadro 3 – Eventos aceitos pelo sistema dada a ocorrência da tarefa

		Elementos do Modelo da Interação		
		<i>Objetos de Interação</i>	<i>Visão</i>	<i>EDITOR (Espaço)</i>
Ocorrência da tarefa associada	<i>(0,0)</i>	eProcedimentoNãoRealizado – ePNR	eVisãoNãoRealizada – eVNR	EspaçoNãoRealizado – eENR
	<i>(0,1)</i>	eProcedimentoNãoRealizado – ePNR eProcedimentoRealizado – ePR	eVisãoNãoRealizada – eVNR	eEspaçoNãoRealizado – eENR
	<i>(0,n)</i>	eProcedimentoNãoRealizado – ePNR eProcedimentoRealizado – ePR eRepetirProcedimento – eRP	eVisãoNãoRealizada – eVNR eRepetirVisão – eRV	eEspaçoNãoRealizado – eENR eRepetirEspaço – eRE
	<i>(1,1)</i>	eProcedimentoRealizado – ePR	-	-
	<i>(1,n)</i>	eProcedimentoRealizado – ePR eRepetirProcedimento – eRP eRealizarLink - eRL	eRepetirVisão – eRV	eRepetirEspaço – eRE

Os eventos ePNR, eVNR e eENR indicam que a tarefa associada a um *Objeto de Interação*, uma *Visão* ou um *Editor* (Espaço), respectivamente, não foi realizada, e portanto, tais eventos serão sempre acompanhados de uma ação que cancela todas as ações e atividades do estado de origem, assim, as assinaturas destes eventos serão acompanhados pela cláusula: <evento>/ **AnulaAçõesEAtividadesDaOrigem()**;

O evento eRealizarLink (eRL) só é admissível a um objeto de interação, se o mesmo, for responsável pela realização de um link a um outro *Editor* (Espaço).

* Transições

As transições entre estados do statechart podem ser: entre estados internos a um super estado (*estado composto*), ou seja, o diálogo intra-*Editor*; ou entre estados internos a um super estado “A” para um outro super estado “B”, sendo este o diálogo inter-Editores.

O diálogo intra-*Editor* é obtido através dos atributos método das tarefas associadas a cada um dos elementos da árvore EDITOR. O valor do atributo método é uma expressão formada pela utilização dos operadores SEQ, OR, XOR e AND, conforme foi especificado no Modelo da Tarefa, que representam quais as seqüências, restrições e condições de realização de uma tarefa, implicando nas mesmas condições de realização (transição) dos estados do statechart.

A combinação entre os atributos ocorrência (que nos fornece os eventos possíveis a cada um dos elementos do modelo EDITOR) e método (seqüência, restrições e condições das transições) fornece todas as transações internas possíveis ao super estado de um *statechart*, ou seja, transações intra-EDITOR. O Quadro 4 apresenta as possíveis combinações entre a ocorrência e os métodos na geração de transições.

Quadro 4 – Combinações possíveis ocorrência versus método

		Método de Realização das Sub-tarefas.					
		SEQ	XOR	OR	AND	SIM	PAR
Ocorrência	(0,0)	-	-	-	-	-	-
	(0,1)	X	X	X	-	X	X
	(0,n)	X	X	X	-	X	X
	(1,1)	X	X	X	X	X	X
	(1,n)	X	X	X	X	X	X

As transições inter-EDITOR são obtidas através da hierarquia das árvores EDITOR (Espaços). Uma árvore EDITOR é dita hierarquicamente superior a outra, se a tarefa associada ao seu elemento Editor estiver em um nível mais alto do que a tarefa associada ao elemento Editor da outra árvore.

Existem dois tipos de diálogos inter-Editores possíveis:

- árvore hierarquicamente superior para uma árvore hierarquicamente inferior – transição disparada pelo evento eRL do objeto de interação responsável pelo link entre as árvores.
- árvore hierarquicamente inferior para uma árvore hierarquicamente superior – transição disparada pelo evento ePR do objeto de interação indicando a conclusão de uma tarefa.

2.1.3. Protótipo de Média Fidelidade em MEDITE

Baseado nas informações do modelo abstrato de interface com usuário (apresentação + diálogo) disponibilizadas pela ferramenta MAPA, é possível gerar uma representação visual capaz de antecipar o processo de prototipagem em uma etapa intermediária do projeto de

interface. Este protótipo pode contemplar aspectos de edição e simulação, assim como rege a definição de um protótipo de média fidelidade.

Este processo está contemplado em MEDITE e possui suporte computacional da ferramenta SMILE. Por se tratar de uma ferramenta de geração e uso de protótipos de média fidelidade, SMILE fundamenta-se em algumas características essenciais para ferramentas desta natureza, a saber: (i) rapidez e facilidade de construir e modificar o protótipo com baixo investimento de tempo e recurso; (ii) ausência da necessidade de habilidade técnica específica por parte dos projetistas; (iii) possibilidade de explorar diferentes alternativas de projeto; (iv) melhoria na comunicação da equipe de projeto; (v) interação direta entre o usuário e o sistema; (vi) manutenção do histórico do projeto; (vii) reuso de partes do projeto; e, por fim, (viii) possibilidade de realização de testes de usabilidade e de treinamento.

A geração do protótipo de média fidelidade é feita com base nos diferentes elementos do modelo de interação. O Espaço Inicial é o espaço base para todos os outros e, por sua definição, possui duas Visões: Visão Funcionalidade, que contém objetos de interação os quais representam as funcionalidades básicas da aplicação; e, Visão Orientação, que possui objetos que remetem às informações descritivas do sistema (opcional). A Visão Funcionalidade equivale a uma barra de menu e, como tal, está presente em todas as telas do sistema, já a Visão Orientação é substituída pelo conteúdo dos demais Espaços existentes. A Figura 6 apresenta a representação do Espaço Inicial gerada pela ferramenta SMILE.

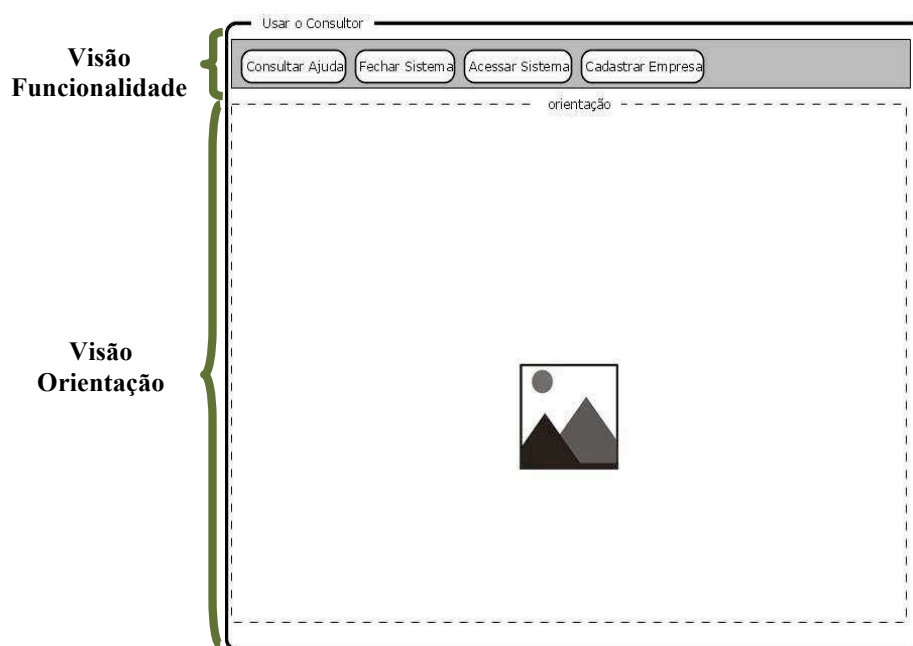


Figura 6 – Representação do Espaço Inicial gerada pela ferramenta SMILE

As principais funções de SMILE são: (i) Inserir OIs, Visões e Espaços; (ii) Excluir OIs, Visões e Espaços; (iii) Mudar tipo de OIs; (iv) Editar Esboço; (v) Exportar Visualização; (vi) Navegar; e (vii) Simular. SMILE disponibiliza, ainda, uma função automática para a manutenção do histórico do design que realiza salvamento do estado atual do esboço enquanto o usuário o manipula através da geração de arquivos de backup. A Figura 7 apresenta uma tela de SMILE durante a edição do protótipo de média fidelidade da interface com usuário para o sistema Consultor, descrito mais adiante.

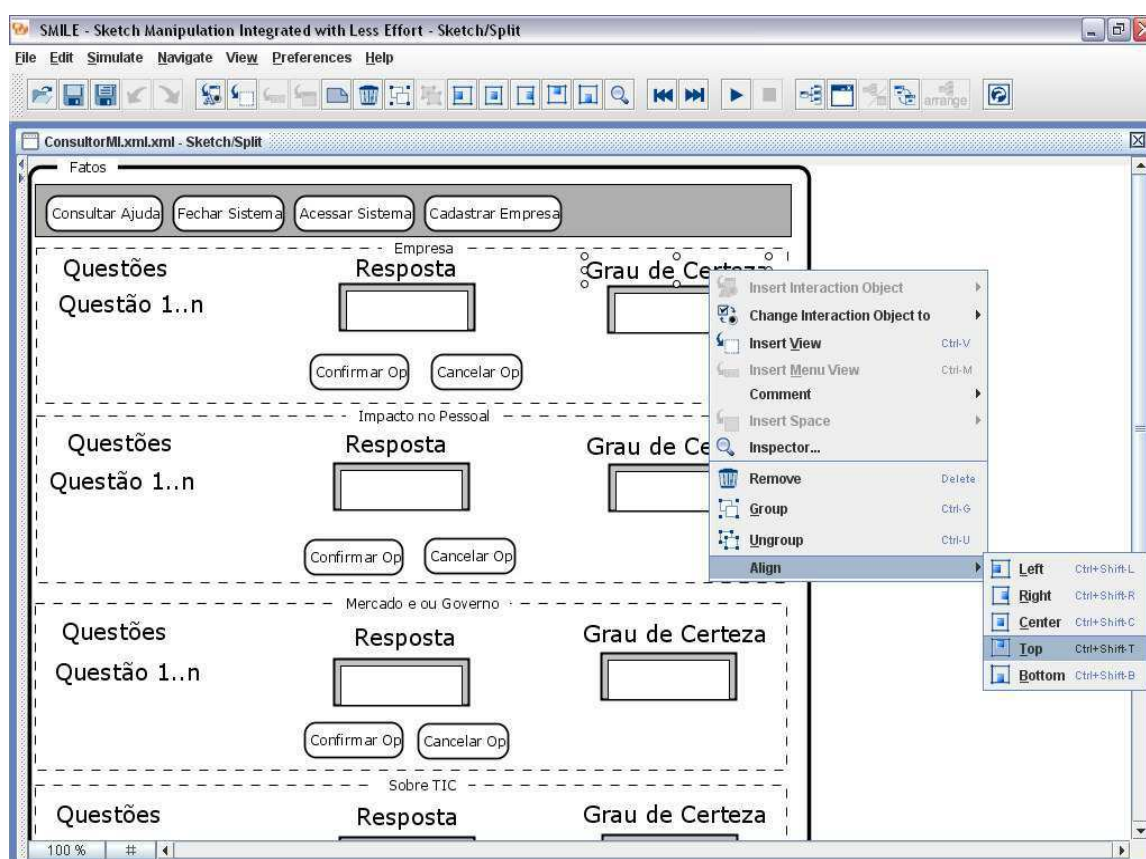


Figura 7 – Tela da ferramenta SMILE durante edição do protótipo de média fidelidade

2.2. Abordagem de Desenvolvimento Baseado em Modelos Utilizando Múltipla Prototipagem Evolutiva

Como foi visto, em MEDITE apenas uma técnica de prototipagem é utilizada, a de média fidelidade, em uma etapa intermediária do processo de concepção baseado em modelos. A

prototipagem de alta fidelidade é feita de maneira informal e leva apenas em consideração a experiência do projetista.

O uso exclusivo de apenas uma técnica de prototipagem, de maneira formal, não resolve as dificuldades enfrentadas pelos projetistas na adoção de uma abordagem de desenvolvimento baseada em modelos. Assim, percebe-se a necessidade de oferecer técnicas de formalização do protótipo de alta fidelidade e que ao mesmo tempo leve em consideração aspectos provenientes dos protótipos de média fidelidade.

Para oferecer prototipagem evolutiva em uma abordagem baseada em modelos, é necessário extensões nos modelos existentes em MEDITE de modo que o formalismo utilizado em cada etapa possa suportar as diferentes formas de modelagem e manipulação, uma vez que está se manipulando representações visuais. Da mesma forma, essas manipulações também devem resultar em operações de manutenção da coerência entre os modelos, o que também resulta em uma extensão no mecanismo de consistência. Nas subseções a seguir, serão apresentadas as extensões que foram realizadas nos modelos provindos de MEDITE e o processo de formalização do protótipo de alta fidelidade, ambas atividades executadas no escopo deste trabalho de dissertação.

2.2.1. Extensões nos modelos de MEDITE

A geração automática de protótipo de média fidelidade em MEDITE se inicia com a modelagem da tarefa. As informações provenientes da etapa *Conceitos e Tarefas* é representada por dois arquivos, um com o XML com informações das tarefas de acordo com o formalismo TAOS, e o outro com informações gráficas, provinda do *framework* gráfico JHotDraw. O arquivo com a descrição da tarefa serve como entrada da ferramenta MAPA, que por sua vez gera o modelo de interação. O fato da ferramenta iTAOS trabalhar com dois arquivos impede a visualização gráfica do modelo da tarefa quando este é editado pela ferramenta MAPA quando da realização da manutenção da consistência. Assim, a especificação do modelo da tarefa foi modificada para incorporar aspectos gráficos e descritivos das tarefas em um único arquivo. Para que a ferramenta iTAOS desse suporte a este tipo de mudança, foi realizado um refatoramento na parte gráfica do sistema.

O modelo da interação gerado pelo MAPA serve como entrada da ferramenta SMILE, que proporciona a geração e edição de protótipos de média fidelidade. Percebeu-se que as regras de transformação entre o modelo da tarefa e modelo de interação eram restritas apenas para alguns cenários de geração de representação visual. Para que a prototipagem de média fidelidade pudesse acontecer de maneira eficaz, uma extensão nas regras de mapeamento foi realizada. Essa extensão levou em consideração tanto aspectos de apresentação quanto de navegação do modelo de interação. O Quadro 5 apresenta as novas regras de mapeamento entre o modelo da tarefa e modelo de roteiro. A geração do modelo de interação é feita pelo mapeamento direto entre os elementos do modelo de roteiro e os elementos do modelo da interação.

Quadro 5 – Novas Regras de Mapeamento do Modelo da Tarefa para Modelo de Roteiro

Regra	Descrição da Regra
Regra Provisória	Se uma tarefa é Tarefa (MTa) e existe algum de seus filhos que seja Tomada, Cena e Cenário (MRo), essa Tarefa será Cenário, Cena e Tomada.
Regra Normalização 02	Se existe alguma tarefa (MTa) sem representação no modelo de roteiro e essa tarefa possui algum filho Cenário (MRo), atribua à tarefa Cenário, Cena e Tomada e o filho também será Cenário, Cena e Tomada.
Regra Normalização 03	Para Cenas Puras (MRo) que possuem filhos Cenário então a Cena será Cenário, Cena e Tomada e o filho será Cenário Cena e Tomada.
Regra Normalização 04	Para Cenário que possuem filhos Cenário Puro, então o filho Cenário Puro será Cenário, Cena e Tomada.
Regra Normalização 05	Se uma Tarefa (MTa) é filha da raiz e é Cena Pura (MRo), então essa tarefa virará Cenário e Tomada e seus filhos Cena e Tomada.

Após a aplicação das regras de mapeamento, um tratamento é feito à visão funcionalidade no sentido de incorporar as tarefas folhas que são filhas da tarefa raiz como tomadas dessa visão. Este tratamento foi preciso pois apenas os objetos que faziam *links* para outros espaços pertenciam à visão funcionalidade e as tarefas ações eram representadas como visões e tomadas independentes. Esta característica era indesejável uma vez que apenas a visão funcionalidade pode ser apresentada em todos os espaços, isto fazia com que as visões e tomadas independentes ficassem inutilizadas. A Figura 8 representa o tratamento que foi dado à visão funcionalidade.

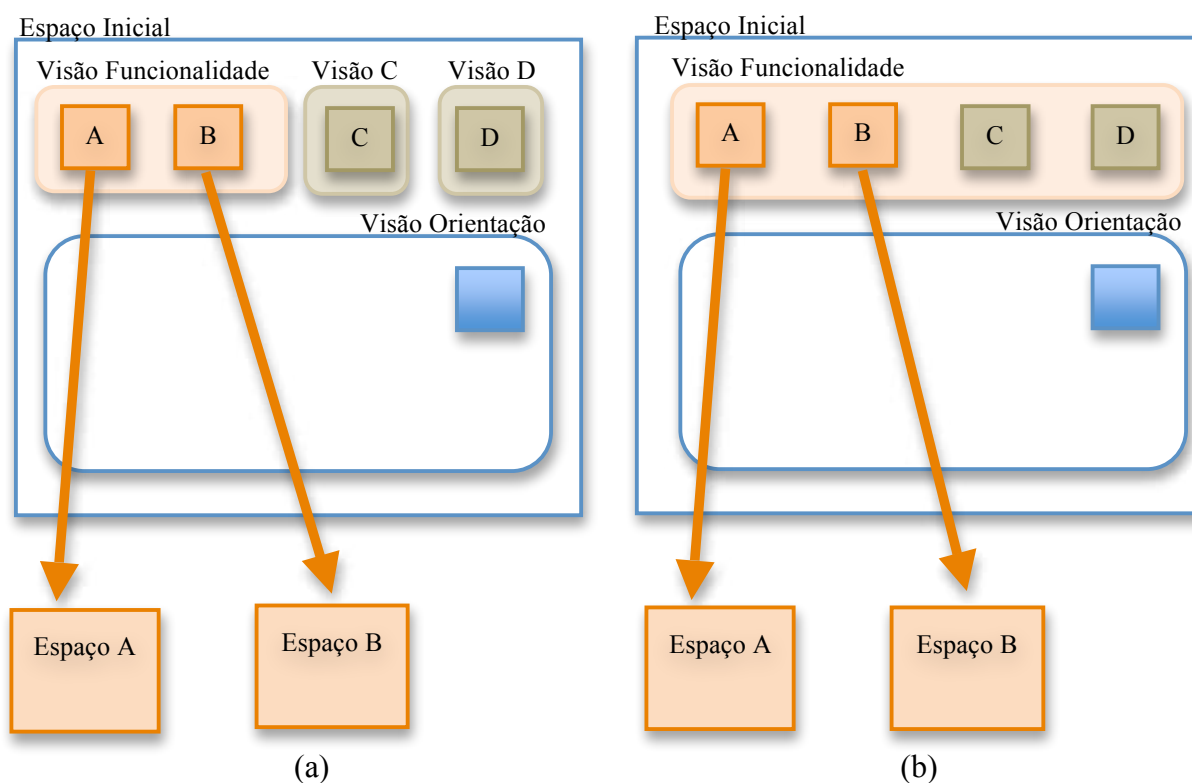


Figura 8 – Esquema do tratamento dado à visão funcionalidade. (a) Espaço Inicial antes do tratamento da visão funcionalidade. (b) Espaço Inicial com tratamento na visão funcionalidade

Com relação à navegação, algumas regras foram incorporadas de modo a garantir algumas operações de simulação. Essas regras foram inseridas principalmente ao diálogo intra-Editor. Como descrito, o diálogo intra-Editor é composto pela combinação entre a ocorrência e o método; porém, quando existiam tarefas com apenas um filho, não era gerada nenhuma transição para os elementos criados neste cenário. Assim, neste caso, foi inserida a criação de uma transição de entrada para a única tarefa filha existente, e a transição de saída para a tarefa que representa o espaço imediatamente superior. O esquema da navegação sem atributo método é apresentado na **Figura 9**.

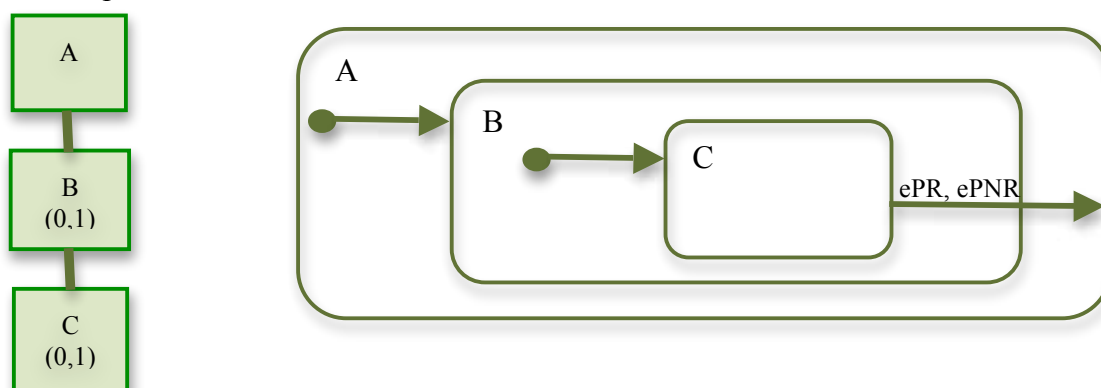


Figura 9 – Criação de transições para tarefa com apenas um filho

2.2.2. Protótipo de Alta Fidelidade

A própria definição da etapa *Interface Concreta do Usuário (CUI)*, em *Cameleon*, define a instanciação de elementos de interação concretos. A transformação do modelo existente na AUI em um modelo da CUI consiste basicamente na ligação entre os objetos abstratos e os objetos concretos pertinentes a uma determinada plataforma. Tendo em mente que a AUI, em MEDITE, é representada sob a forma de um protótipo de média fidelidade, é necessário a criação de um modelo CUI que possa ser representado sob a forma de um protótipo de alta fidelidade provendo, assim, múltipla prototipagem evolutiva em uma abordagem baseada em modelos.

No Quadro 6, contém as regras de transformação entre a AUI e CUI, em MEDITE, de modo que o modelo CUI possa dar suporte a criação e edição de protótipos de alta fidelidade.

Quadro 6 – Regras de Transformação entre Modelo de Interface Abstrata do Usuário (AUIModel) e Modelo de Interface Concreta do Usuário (CUIModel)

Regra	Descrição da Regra
Regra CUI01	Os elementos do modelo da interação (AUIModel) serão denominados UIComponents (CUIModel). Todo espaço (AUIModel) será equivalente a um elemento Window (CUIModel); Toda visão (AUIModel) será equivalente a um Box (CUIModel); Todo objeto de interação (AUIModel) será equivalente a um UIElement (CUIModel). O UIElement pode ser do tipo: <ul style="list-style-type: none"> • UICheckbox • UICommandButton • UIFigure • UIRadio • UISelectionbox • UITextArea • UITextField
Regra CUI02	O atributo de coordenada dos elementos do modelo interação (AUIModel) terá relação direta com as coordenadas dos elementos UIComponents (CUIModel).
Regra CUI03	A criação de atributos de navegação e validação dos UIComponents (CUIModel) serão extraídos do Diagrama de Estados (<i>Statecharts</i>) do modelo de interação (AUIModel).
Regra CUI04	Todo UIComponent (CUIModel) receberá um estilo padrão que poderá ser editado a nível individual ou agrupado por categorias de acordo com os elementos do modelo da interação (AUIModel).

As informações provenientes dos protótipos de média fidelidade são aspectos de localização e representações visuais dos elementos interativos. Desta forma, os protótipos de alta fidelidade devem incorporar o que já foi editado pelos projetistas e validado pelo usuário do sistema em relação aos protótipos de média fidelidade. Após essa correlação, define-se os aspectos estéticos (padrão, fonte, cor, tamanhos de botões, etc.) e os componentes de navegação.

O esquema de transformação do modelo da interação (AUIModel), representado sob a forma de um protótipo de média fidelidade, para o modelo de interface concreta (CUIModel), representado sob a forma de um protótipo de alta fidelidade, é apresentado na **Figura 10**. Em primeiro lugar, o esquema define uma transformação de objetos interativos abstratos em objetos concretos de uma determinada plataforma. Para efeitos deste trabalho, utilizou-se a plataforma Web por ser a plataforma mais utilizada nos dias atuais. A seguir, é definido um alinhamento dos objetos, uma vez que a edição na etapa anterior é à mão livre. Com relação à navegação, agrega-se aspectos como foco inicial da tela, botões de links funcionais e restrições reais. Por fim, utiliza-se o estilo padrão para o qual o software seria desenvolvido. Este estilo pode ser editado de modo a deixar o protótipo mais agradável ao uso.

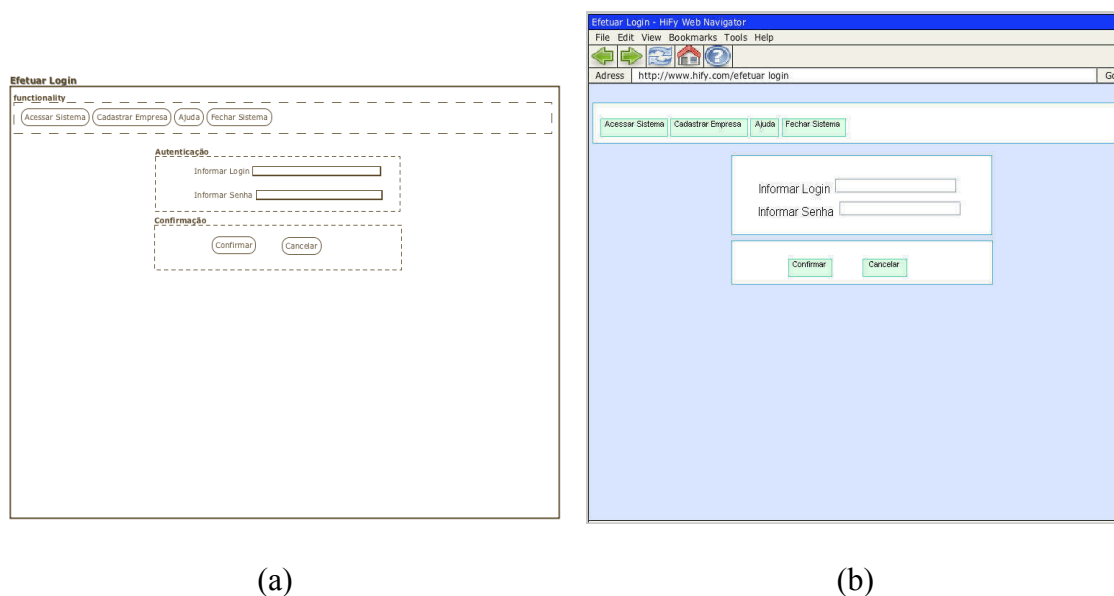


Figura 10 - Representação CUI a partir da AUI. (a) Protótipo de média fidelidade. (b) Protótipo de alta fidelidade

2.2.3. Necessidade de Integração

Mesmo com as ferramentas existentes, a criação e a edição dos diversos modelos envolvidos na concepção de IU em MEDITE ainda são muito onerosas visto que o processo é realizado de maneira independente e todas as operações de manipulação e transformação ficam a cargo do projetista. Uma forma de se reduzir, ou mesmo eliminar, estas dificuldades é fornecer uma infra-estrutura que garanta a ligação e consistência entre estes modelos de forma automatizada e em tempo real de projeto.

O desenvolvimento de IU baseada em modelos, em especial MEDITE, segue o processo definido no *framework* de referência unificado *Cameleon*. A Figura 11 apresenta a correspondência entre os níveis de abstração do processo de desenvolvimento proposto em *Cameleon* e as fase e artefatos de MEDITE.

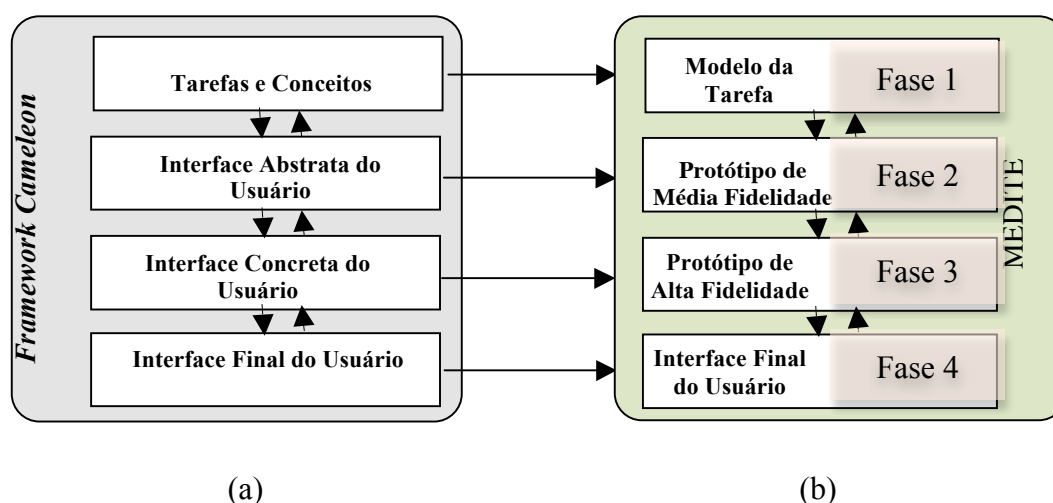


Figura 11 – Correspondência entre os níveis de abstração propostos em *Cameleon* e as fases e artefatos de MEDITE. (a) Framework Cameleon. (b) MEDITE.

Com a especificação formal de um protótipo de alta fidelidade a partir de um protótipo de média fidelidade em uma abordagem baseada em modelos, foi possível garantir a infra-estrutura necessária para automatização do processo de desenvolvimento de interface com usuário que resolve os problemas expostos no capítulo 1 e fazer uma correlação direta entre os níveis de abstração de *Cameleon* com as diferentes técnicas de prototipagem. A infra-estrutura ora apresentada denominar-se-á *framework FastInterface*.

2.3. Conclusões

Neste capítulo, apresentou-se o uso evolutivo de protótipos de média e alta fidelidade em fases diferentes (e adequadas) do processo de desenvolvimento da interface com usuário. Esta abordagem facilita a aceitação do uso de uma abordagem baseada em modelos por parte dos projetistas de interface, visto que:

- (i) a transição entre diferentes níveis de representação visual durante todas as fases do processo (inicial, intermediária e final) permite que os projetistas tenham um certo grau de previsibilidade da interface final, que será gerada ao final do processo, em suas diferentes fases;
- (ii) o uso constante de representações visuais da interface permite que técnicas de avaliação de interface sejam empregadas em todas as fases do processo, minimizando (ou até mesmo evitando) a necessidade de realizar avaliações de interface em cima de modelos declarativos.

Assim, com o objetivo de incentivar/facilitar o uso de uma abordagem baseada em modelos para o processo de desenvolvimento de interface com usuário, propõe-se neste trabalho a aplicação da técnica de múltipla prototipagem evolutiva nos processos definidos em MEDITE baseados nos processos apresentados por *Cameleon* através da definição e utilização de protótipos de média e alta fidelidade. Esta infra-estrutura será denominada *framework FastInterface*.

Foi apresentado que o nível AUI contém informações sobre os elementos interativos e sobre a navegação entre eles (Coyette *et. al.*, 2006) e deve ser derivado do nível precedente, *Tarefas e Conceitos*. Essas informações são suficientes para a produção de uma representação na forma de um protótipo de média fidelidade da interface com usuário. Esse tipo de protótipo é gerado a partir da especificação do modelo da tarefa. Desta maneira, é um protótipo que incorpora as funcionalidades capturadas na etapa anterior, podendo ser editado e simulado, porém, não possui aspectos estéticos nem é dependente de nenhuma plataforma específica.

Para o nível CUI, o uso de protótipos de alta fidelidade é adequado, pois em momentos avançados do processo de desenvolvimento é importante considerar representações visuais mais refinadas, que contemplem localização, tipos de objetos concretos, navegação e aspectos estéticos, tais como: fonte, cor, tamanho dos elementos, e etc. A partir do protótipo de alta

fidelidade, treinamentos podem ser realizados, assim como testes de usabilidade mais completos e abrangentes.

Dessa forma, constatou-se que é possível utilizar múltiplas técnicas de prototipagem em uma metodologia compatível com o *framework Cameleon*. MEDITE, então passa a contemplar um processo evolutivo do uso de protótipos de interface com usuário.

Para fornecer suporte computacional a esta abordagem, necessita-se desenvolver uma ferramenta para geração e edição do modelo CUI e geração da interface final. Tal como, disponibilizar um ambiente que integre todos as ferramentas existentes e proveja manutenção da consistência em tempo real de projeto. A ferramenta e ambiente serão denominados respectivamente Hi-Fy e FastInterface e serão descritas nos próximos capítulos.

Capítulo 3

Projeto e Implementação de Hi-Fy

Este capítulo tem como objetivo descrever Hi-Fy, uma ferramenta que foi projetada e desenvolvida no escopo desta pesquisa para a geração automática, a edição e a simulação de protótipos de alta fidelidade. Essa ferramenta está inserida no contexto de desenvolvimento de interface com usuário baseado em modelos, adequando-se à fase 3 de MEDITE (CUI em *Cameleon*). Como Hi-Fy será incorporada enquanto módulo ao MB-IDE FastInterface, grande parte das decisões de projeto e implementação tomadas para Hi-Fy foram também previstas para o ambiente em questão. Desta forma, será abordado neste capítulo alguns artefatos gerados ao longo do desenvolvimento da ferramenta, em seguida, expor algumas de suas funcionalidades através da apresentação de telas do sistema.

3.1. Propósito de Hi-Fy

A metodologia MEDITE provê três ferramentas de suporte aos seus processos: iTAOS, para modelagem da tarefa; MAPA, para geração do modelo AUI de acordo com o modelo da tarefa; e, SMILE, para representação visual da AUI sob a forma de um protótipo de média fidelidade. Para que o ambiente FastInterface possa contemplar o mapeamento entre os elementos do modelo AUI e os elementos do modelo CUI, permitir edição e simulação do modelo CUI a partir da edição e simulação de protótipos de alta fidelidade, e por fim, permitir geração da interface final, é preciso o desenvolvimento de um módulo que possa executar essas atividades. Esse módulo será denominado Hi-Fy, acrônimo da técnica de prototipagem Hi-Fi.

O modelo CUI descrito neste trabalho contempla instanciação de objetos concretos para uma determinada plataforma, aspectos de estilo e fonte dos protótipos gerados, agregando informações provenientes da etapa anterior do processo. Desta forma, a concepção do módulo Hi-Fy é o ponto chave para favorecer a uma abordagem evolutiva de prototipagem em um processo baseado em modelos.

Hi-Fy também deve disponibilizar a opção de geração da interface final em alguma linguagem interpretada ou compilada. Esta geração deve refletir o protótipo editado e validado na etapa CUI. A plataforma e linguagem escolhidas para primeira versão do módulo foi a plataforma Web e linguagem HTML, respectivamente, por se tratarem de tecnologias bastante difundidas no cenário de desenvolvimento de software atual e pela simplicidade em construí-los.

3.2. Metodologias de desenvolvimento

Hi-Fy e FastInterface são projetos acadêmicos que foram implementados através de uma parceria entre a Universidade Federal da Paraíba (UFCG) e o Centro Federal de Educação Tecnológica da Paraíba (CEFET-PB), sob a vigência de um projeto de iniciação científica. Tendo em vista o caráter acadêmico e ágil, e uma vez que a natureza deste tipo de projeto remetia a curto prazo de finalização, era necessário oferecer um método de desenvolvimento que garantisse agilidade para o desenvolvimento, mas que incluísse em seu fluxo aspectos relacionados com a qualidade da interface com usuário. Para este caso, optou-se pelo uso de um processo de desenvolvimento de software, denominado easyProcess (YP) (Garcia *et. al.*, 2004), integrado à metodologia MEDITE, de modo que o sistema a ser desenvolvido pudesse contemplar ganhos tanto do ponto de vista funcional como de usabilidade.

O YP consiste em um processo de desenvolvimento de software simplificado, apoiado em práticas do XP, AM e RUP, para propiciar a construção de software de forma ágil. YP propõe a construção de artefatos necessários ao contexto acadêmico, no qual é preciso realizar avaliações constantes da evolução dos alunos na aplicação do processo, “simulando” a realidade fora da universidade.

O fluxo de YP compreende as seguintes fases: (i) **Definição de Papéis**; (ii) **Conversa com o Cliente**, que consiste na definição de uma visão sobre o processo de negócio e no levantamento dos requisitos funcionais e não-funcionais, do perfil do usuário e objetivos de usabilidade; (iii) **Inicialização**, que é composta das atividades de

modelagem da tarefa, levantamento das *user stories* e dos testes de aceitação, geração do protótipo da interface, elaboração do projeto arquitetural e geração do modelo lógico de dados (quando necessário); (iv) **Planejamento**, que define os planos de release e de interação; (v) **Implementação**, que consiste na construção do código da aplicação e de testes. As Reuniões de Acompanhamento são determinantes na avaliação da implementação realizada durante uma iteração e fundamentais para o planejamento das iterações e releases posteriores. E, por fim, (vi) **disponibilização da Versão do Sistema**. O fluxo de YP pode ser visto na Figura 12.

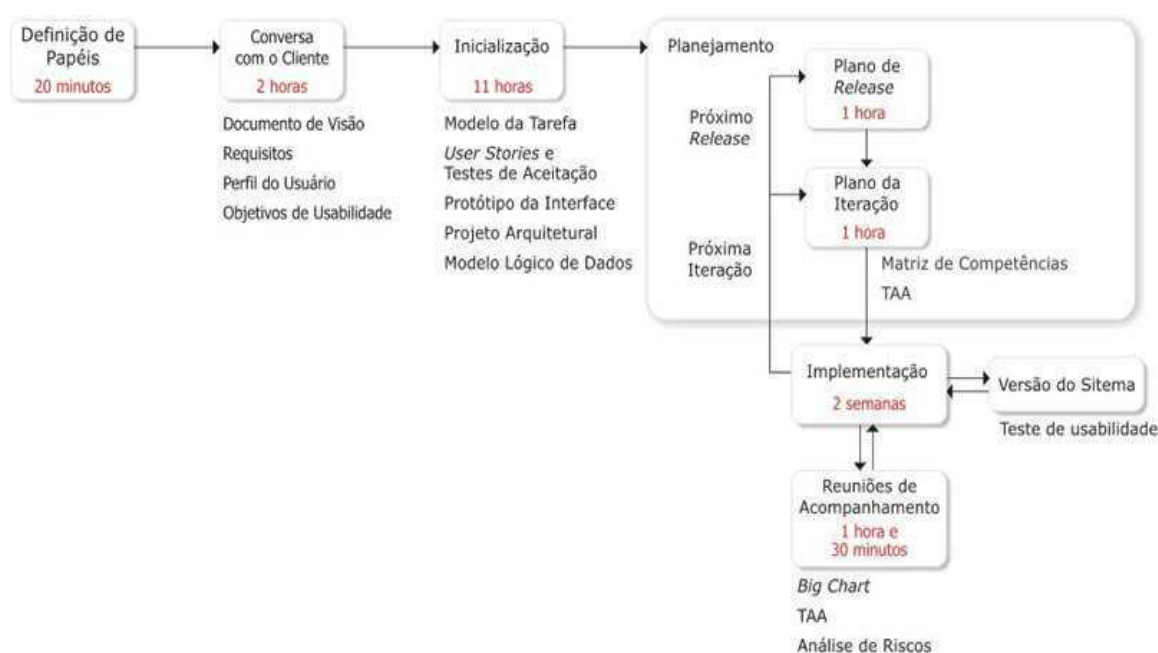


Figura 12 - Fluxo do processo de desenvolvimento easYProcess
 Fonte: easYProcess, 2003

Apesar de YP contemplar aspectos de usabilidade, a respeito do: perfil do usuário, objetivos de usabilidade, modelo da tarefa e protótipo de interface, estes artefatos não possuem uma correlação direta. Por esta razão optou-se por fazer uso da metodologia MEDITE como forma de prover aparato ferramental e metodológico para concepção da interface com usuário de maneira automatizada. Desta forma, o desenvolvimento da interface das ferramentas foi feito, tomando-se como pressuposto as extensões realizadas em MEDITE no que diz respeito à utilização de protótipos de alta fidelidade.

3.3. Tecnologias utilizadas

Tanto a ferramenta Hi-Fy como o ambiente FastInterface foram desenvolvidos na plataforma J2SE 6 (Java, 2007). Para a camada gráfica, foi feito uso do *framework Open Source JHotDraw 7.0.8* (JHotDraw, 2004). Para a camada de dados, utilizou-se a API JDOM (JDOM, 2004), cujo maior benefício é fornecer funções de manipulação em arquivos XML (leitura e gravação). As ferramentas disponibilizam um mecanismo de ajuda que foi implementado com o suporte da API Java Help 2.0 (Java Help System, 2007). Com relação à automatização de testes, utilizou-se o *framework JUnit* (JUnit, 2001), para garantir as funcionalidades das unidades de código, e o *framework Open Source EasyAccept* (EasyAccept, 2005), o qual auxilia na criação e execução de testes de aceitação.

3.3.1. Framework JHotDraw

JHotDraw (JHD) (JHotDraw, 2004) é *framework* que oferece um conjunto de componentes para construção de editores gráficos, permitindo a elaboração de esboços, diagramas e desenhos artísticos.

O uso da API de JHD remete à utilização de classes que atendem a determinados cenários. Para efeito das ferramentas em questão, vale salientar, de forma simplificada, as principais interfaces e classes que foram implementadas ou especializadas, as quais podem ser descritas no Quadro 7.

Quadro 7 – Descrição das principais classes e interfaces de JHD que foram utilizadas

Classes	Descrição
Application	Responsável pelo ciclo de vida da aplicação, gerenciando seu(s) projeto(s).
AbstractApplication	Implementação da interface Application com todos os métodos vazios.
DefaultMDIApplication	Corresponde a uma aplicação com suporte a múltiplos projetos.
ApplicationModel	Reúne as ações da aplicação e responsável por construir/inicializar componentes como projetos, menus e toolbars.
DefaultApplicationModel	Implementação da interface ApplicationModel usada em DefaultMDIApplication.
Project	É a unidade de trabalho da aplicação, composto de no mínimo um <i>DrawingView</i> . Está ligado diretamente a um arquivo.
AbstractProject	Implementação da interface Project com todos os métodos vazios.

3.3.2. API JDOM

JDOM (*Java Document Object Model*) é uma API JAVA *Open Source* responsável pela manipulação de documentos XML. Ele provê uma representação estruturada do documento e define um meio pelo qual a estrutura pode ser acessada por programas permitindo-os alterar sua forma, estilo e conteúdo.

Os documentos XML têm sua estrutura baseada em árvore cujos elementos, atributos e conteúdos são previamente definidos e são visto como nodos desta. A API de JDOM permite validação, criação, manipulação e persistência dos documentos de acordo com seu esquema ou descrição.

Uma vez carregado o documento em memória, através de um *Parser* (não disponibilizado por JDOM), são executadas as manipulações desejadas. De posse da árvore, pode-se persisti-la em formato XML desde de que sejam atendidas as restrições estabelecidas.

3.3.3. Padrões de projeto

Os padrões de projeto (Gamma *et. al.*, 2000) visam facilitar a reutilização de soluções de projeto e de código. Também acarretam um vocabulário comum de projeto, facilitando comunicação, documentação e aprendizado dos sistemas de software. Assim, esta seção visa apresentar os padrões de projeto utilizados na construção das ferramentas Hi-Fy e FastInterface.

Do ponto de vista arquitetural, as ferramentas Hi-Fy e FastInterface fizeram uso do padrão MVC (*Model View Controller*). O maior ganho de se utilizar este padrão é a independência da parte funcional do sistema com a parte da interface com usuário, isto faz com que seja possível o reuso de partes do código em outros contextos de uso. O padrão MVC propõe a criação de três camadas, a saber: (i) Camada de Modelo (*Model*), que compreende os itens do domínio do problema; (ii) Camada de Visão (*View*), a qual define a representação visual dos itens do modelo na interface com usuário; e, (iii) Camada de Controle (*Controller*), responsável por todo processamento do sistema, onde deve ser feita a comunicação entre o modelo e a visão.

Tendo em vista a Camada de Visão, JHD utiliza vários padrões de projeto, portanto, a aplicação que usa esse *framework* acaba por implementar muitos desses padrões, entre eles: *Command*, usado na criação de ações para itens dos menus e barra de ferramentas;

AbstractFactory, usado na criação dos diversos painéis e menus presentes na aplicação; e, *Memento*, usado na construção da estrutura de *Undo/Redo*;

As ferramentas consideram ainda os seguintes padrões: *Expert*, *Creator*, *Controller* (padrões de responsabilidade (GRASP)) e *Facade* (Estrutural).

3.4. Artefatos

Os sub-itens seguintes têm por finalidade apresentar os principais artefatos definidos em YP e gerados ao longo do desenvolvimento da ferramenta Hi-Fy. Alguns destes artefatos também foram utilizados na construção do ambiente FastInterface.

3.4.1. Requisitos Funcionais

Dentre as principais funcionalidades do módulo Hi-Fy, podem-se destacar: (i) Realização de leitura do modelo AUI; (ii) Realização de transformações e correlações entre AUI e CUI; (iii) Realização de leitura do modelo CUI previamente existente; (iv) Visualização do modelo CUI através de protótipos de alta fidelidade sob diferentes visões, quais sejam: Árvore de Telas e Protótipo Corrente; (v) Navegação entre os protótipos existentes; (vi) Edição de protótipos; (vii) Realização de salvamento do modelo CUI; (viii) Geração do modelo FUI; e, (iv) Exportação da FUI na linguagem HTML.

A função de edição envolve a manipulação direta nos estilos dos elementos interativos dos protótipos de alta fidelidade (*Window*, *Box*, *UIElement*). A edição de estilo pode ser feita sob uma de duas perspectivas: Edição de Estilo em Grupo, que permite a consistência da apresentação do elemento editado através dos diferentes protótipos; ou Edição de Estilo Individual, que favorece a edição do elemento para um contexto de uso particular em uma determinada tela do sistema. Entre as opções de edição, podem-se destacar: (i) Edição de cor de fundo, para os elementos *Window*, *Box* e *UIButton*; (ii) Edição de estilos de bordas, para os elementos *Windows*, *Box* e *UIButton*; e, (ii) Edição de Fonte, para os elementos *UIElements*.

3.4.2. Requisitos Não-Funcionais

Os requisitos não-funcionais de Hi-Fy e FastInterface são: (i) implementação utilizando JAVA 1.6; (ii) utilização do *framework JHotDraw* como base para a implementação da interface gráfica; (iii) utilização da API JDOM para viabilizar a persistência em formato XML; (iv) implementação de uma *suite* de testes de unidade a partir do uso do *framework JUnit*; (v) disponibilização da documentação Javadoc completa; (vi) internacionalização; e, (vii) disponibilização de mecanismo de ajuda.

3.4.3. Perfil do Usuário

Os usuários da ferramenta Hi-Fy, tal como do FastInterface, são pessoas envolvidas com o projeto de interfaces com o usuário. Estes podem seguir a metodologia MEDITE com o suporte computacional oferecido até o momento (iTAOS, MAPA, SMILE), para o caso do Hi-Fy, ou fazer uso do *framework* FastInterface.

A equipe de concepção de interfaces com o usuário, geralmente multidisciplinar, envolve desenvolvedores, projetistas visuais, clientes, usuários, especialistas em interfaces do usuário, entre outros. Percebe-se que os usuários do Hi-Fy e FastInterface possuem conhecimentos distintos, experiências profissionais diversas e dominam diferentes partes do conhecimento necessário para a construção da interface do sistema que estão desenvolvendo.

3.4.4. Objetivos de Usabilidade

Objetivos de usabilidade remetem a um conjunto de metas de usabilidade que devem ser mensuráveis e serem alcançados pelo sistema que se está desenvolvendo. Estas metas referem-se: à eficiência, à eficácia, à segurança, ao aprendizado e à memorização do sistema.

Dentre os objetivos de usabilidade levantados para a ferramenta Hi-Fy e ambiente FastInterface, tem-se: (i) Reduzir a taxa de erros: Número de tarefas concluídas sem falhas; (ii) Adequar conteúdo (terminologia e simbologia): Aprendizado mantido mesmo com uso pouco freqüente do sistema; (iii) Aumentar a satisfação subjetiva do usuário: Confrontar a

primeira impressão do usuário (Teste de Usabilidade 1) com sua opinião após uso prolongado do sistema (Teste de Usabilidade 2); (iv) Manter clareza na estrutura: Número de ações incorretas, de erros repetidos e consulta à ajuda (*online* ou *off-line*); (v) Ser atrativo ao usuário: Sondar a satisfação subjetiva do usuário ao utilizar o sistema; e, (vi) Possuir telas simples: Observar dificuldades de navegação.

3.4.5. Modelo da Tarefa

A análise e modelagem da tarefa é um processo que descreve um conjunto de tarefas com o objetivo de entender o funcionamento cognitivo dos usuários. Em MEDITE, o modelo da tarefa é o artefato resultante da fase 1 (Tarefas&Conceitos em *Cameleon*). Por utilizar MEDITE como metodologia para concepção de interface com usuário, o modelo da tarefa tanto de Hi-Fy quanto FastInterface foi construído com base no formalismo TAOS e com o suporte ferramental do iTAOS.

3.4.6. *User Stories* e Testes de Aceitação

User Stories são especificações acerca das funcionalidades do sistema que se deseja implementar. A especificação de cada *User Story* é seguida da definição de um conjunto de Testes de Aceitação, os quais servem para descrever o que o cliente aceita como válido. Para o Hi-Fy foram descritos 14 *User Stories* e 34 Testes de Aceitação. Alguns Testes de Aceitação foram automatizados com o auxílio do *framework* EasyAccept.

EasyAccept que tem por objetivo facilitar o entendimento do cliente a respeito dos testes de aceitação, para ele possa participar de concepção dos testes e verificar os resultados. Os testes são escritos numa linguagem simples de *script* que não exige um alto nível de conhecimento sobre programação. Quando os testes são executados, eles chamam os métodos da camada de regras de negócio e retornam se os resultados obtidos são os esperados ou não.

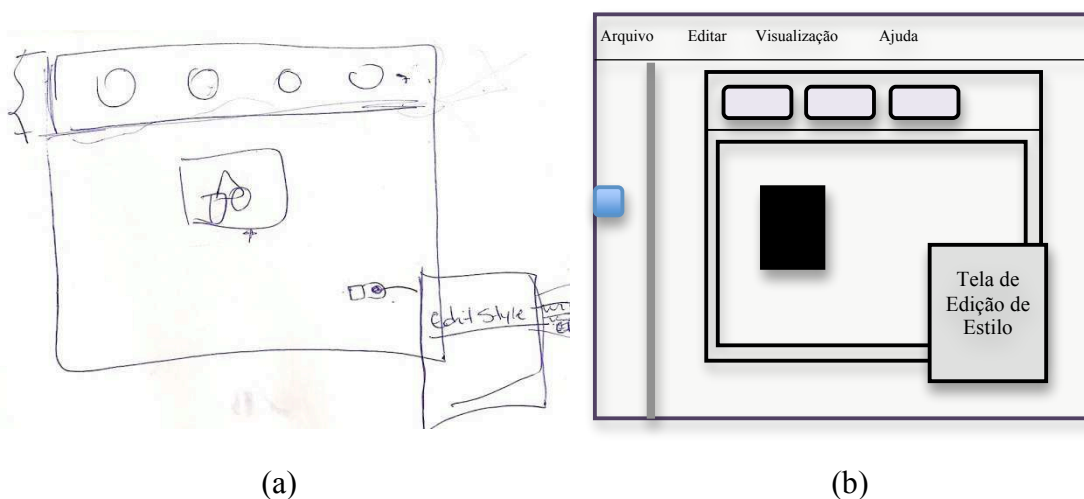
No Quadro 8, apresenta-se um exemplo de especificação de *User Story* e Testes de Aceitação para a ferramenta Hi-Fy.

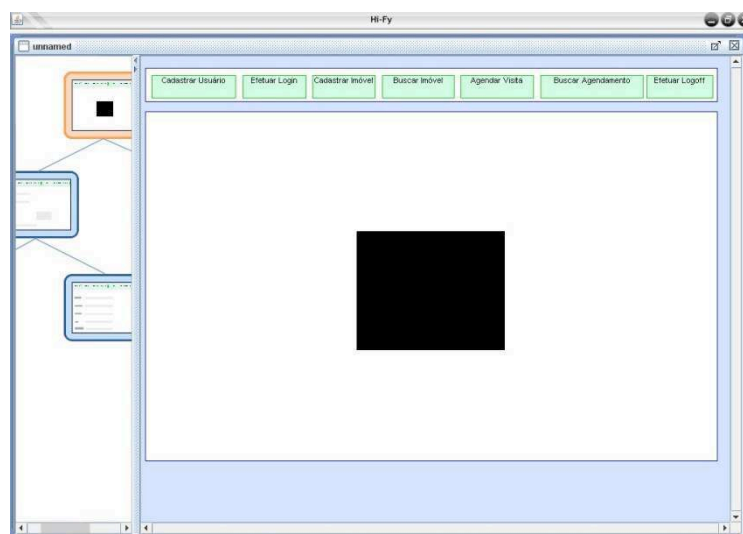
Quadro 8 – Exemplo de especificação de *User Story* e Testes de Aceitação para a ferramenta Hi-Fy.

US 04 – Abrir MI					
Essa tarefa consiste em realizar a operação de abrir um modelo de interação (MI) com informações providas do módulo SMILE.					
TA4.1	Verificar se o arquivo é do tipo FastInterfaceXML				
TA4.2	Verificar se o XML é um arquivo contendo o MI com informações providas do SMILE, ou seja, se contém informações de coordenadas diferente de zero, e atributos dos tipos dos objetos.				
Atividade	Descrição	Responsável	Estimativa de tempo (horas)	Tempo real (horas)	Status
Fazer ação de abrir		Gabriela	4,0	1,7	●
Refatorar arquitetura na camada de persistência		Gabriela	4,0	2,4	●
Validar arquivo XML		Gabriela	3,0	1,3	●
Carregar o arquivo XML em memória		Gabriela	4,0	3,0	●

3.4.7. Protótipo da interface

O protótipo do Hi-Fy foi confeccionado com base no modelo da tarefa especificado na etapa anterior. Como a ferramenta SMILE estava ainda em fase de desenvolvimento quando do projeto de Hi-Fy, tanto os protótipos de baixa, quanto os de média e alta fidelidade foram feitos de maneira não automatizada pela projetista. A Figura 13 apresenta a evolução dos diferentes protótipos produzidos para a ferramenta Hi-Fy.





(c)

Figura 13 – Visão de protótipos de diferentes fidelidades da ferramenta Hi-Fy.(a) Protótipo de Baixa Fidelidade. (b) Protótipo de Média Fidelidade. (c) Protótipo de Alta Fidelidade

3.4.8. Projeto Arquitetural de Hi-Fy

O projeto arquitetural de Hi-Fy segue o padrão MVC (*Model-View-Controller*), o qual sugere a divisão do código do sistema em camadas pré-definidas, quais sejam: Dados, Negócio, Controle e de Visão.

A Figura 14 apresenta o esquema da arquitetura do módulo Hi-Fy.

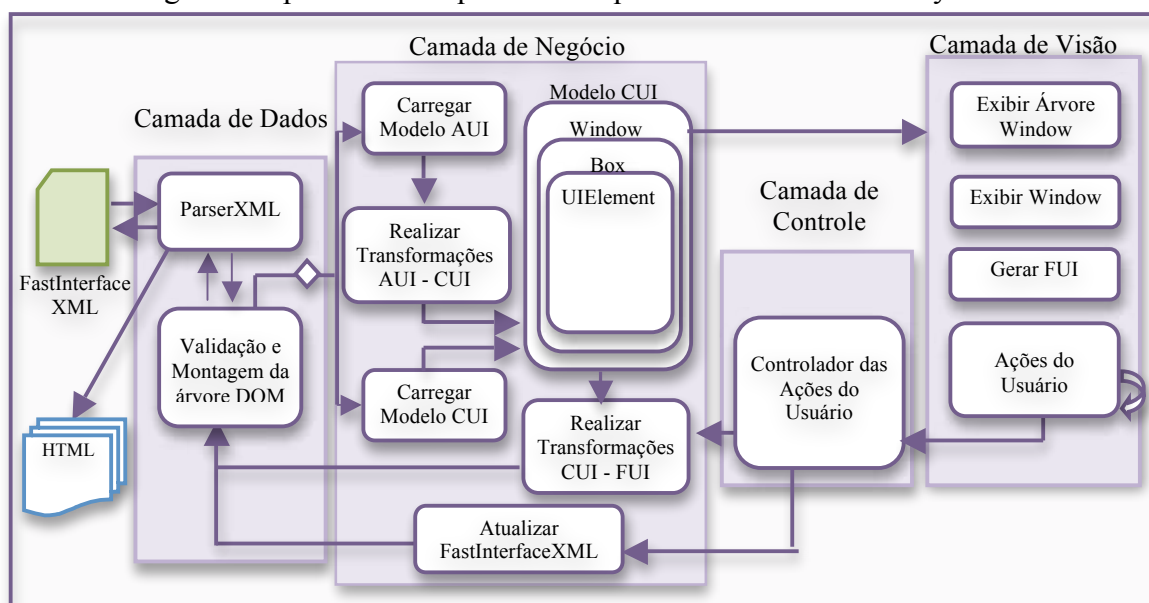


Figura 14 – Arquitetura do Módulo Hi-Fy

A camada de negócio com ênfase no modelo CUI pode ser visto na Figura 15.

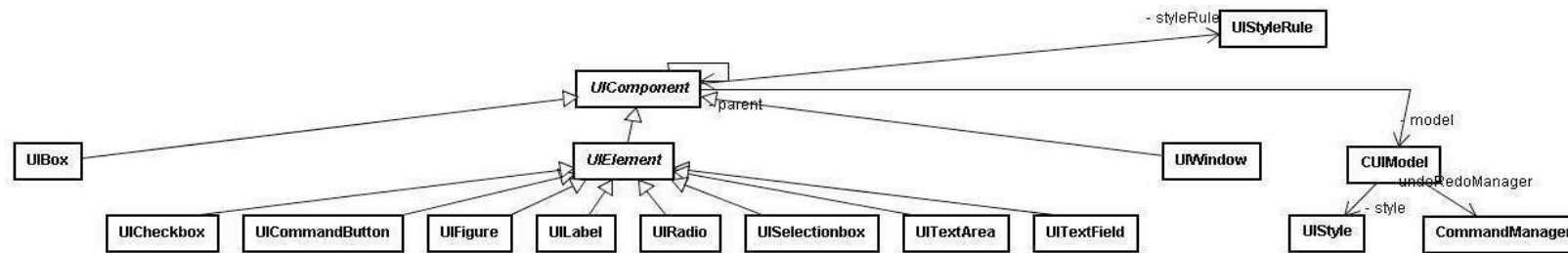


Figura 15 - Diagrama de Classes da Camada de Negócio do Hi-Fy com ênfase no modelo CUI

3.5. Ferramenta Hi-Fy

As subseções a seguir contêm as principais funcionalidades e modos de visualização da ferramenta Hi-Fy.

3.5.1. Modos de Visualização

De modo a favorecer a edição dos diferentes protótipos, a ferramenta Hi-Fy disponibiliza diferentes modos de visualização: (i) **Árvore de Protótipos**; (ii) **Protótipo**; e, (iii) **Split** (Árvore de Telas e Protótipo simultaneamente). Nas Figuras 16, 17 e 18, apresentam-se os modos de visualização supra-mencionados.

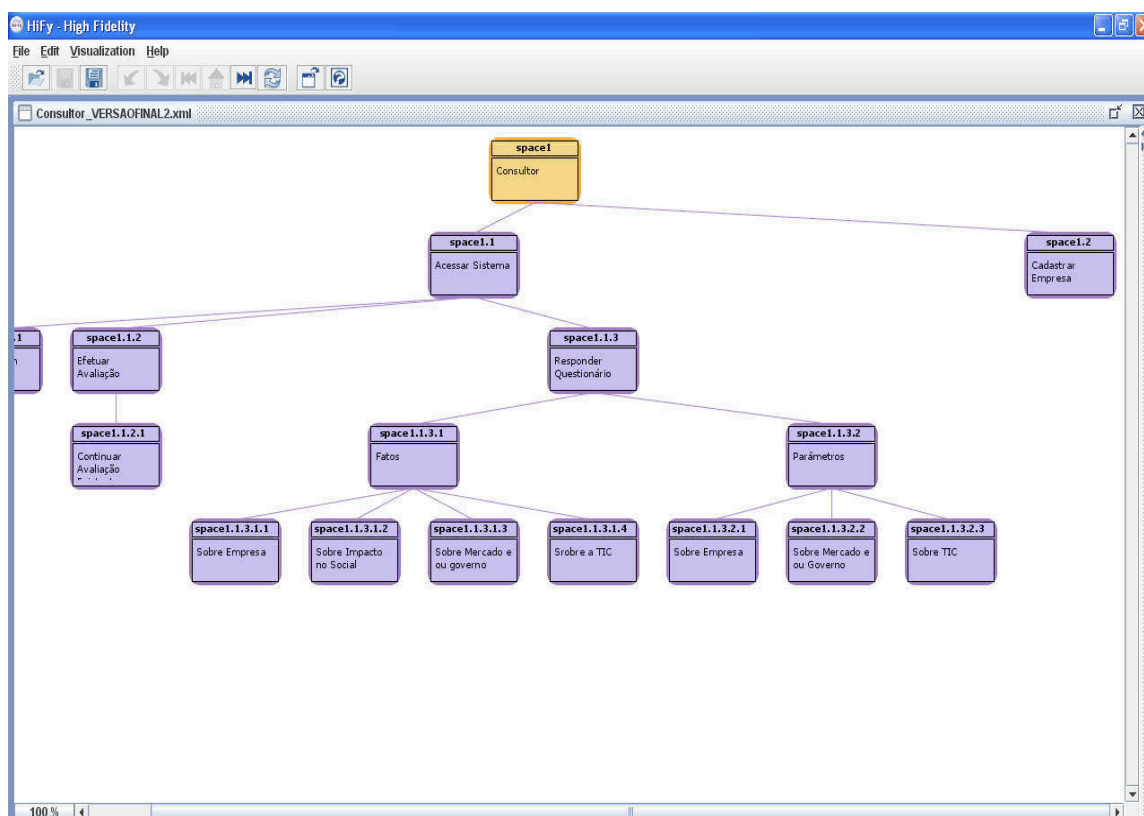


Figura 16 – Modo de Visualização Árvore de Protótipo

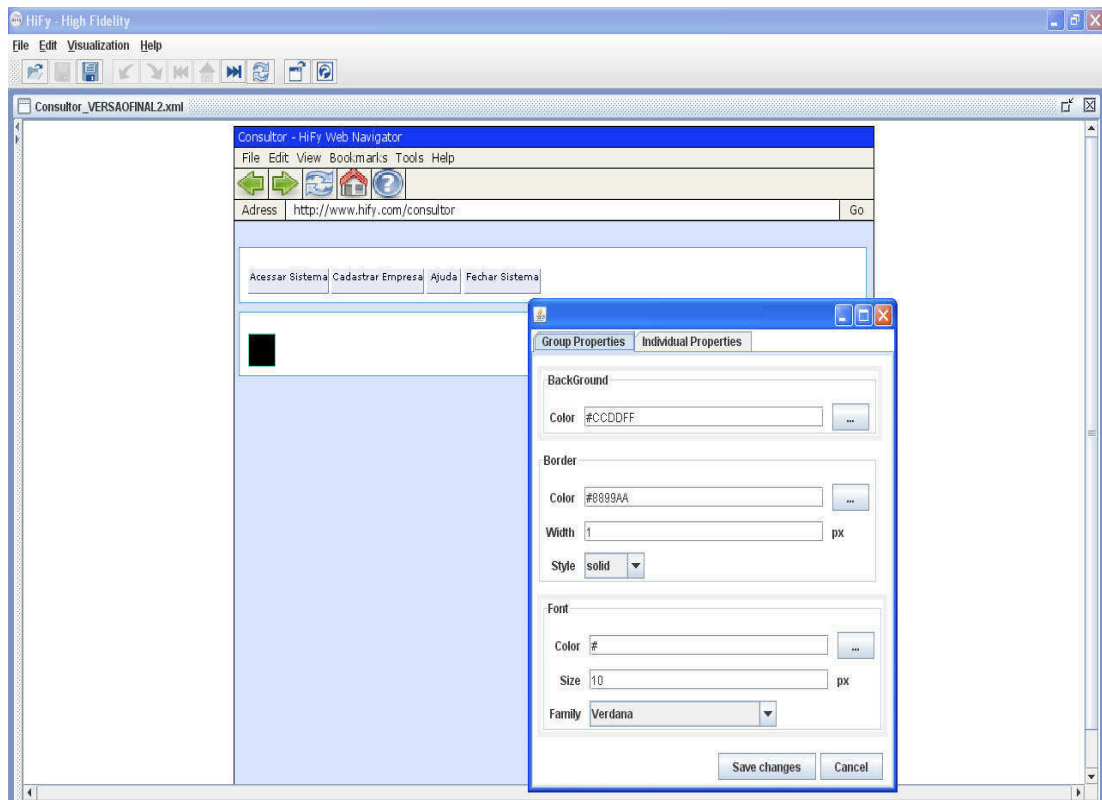


Figura 17 – Modo de Visualização Protótipo

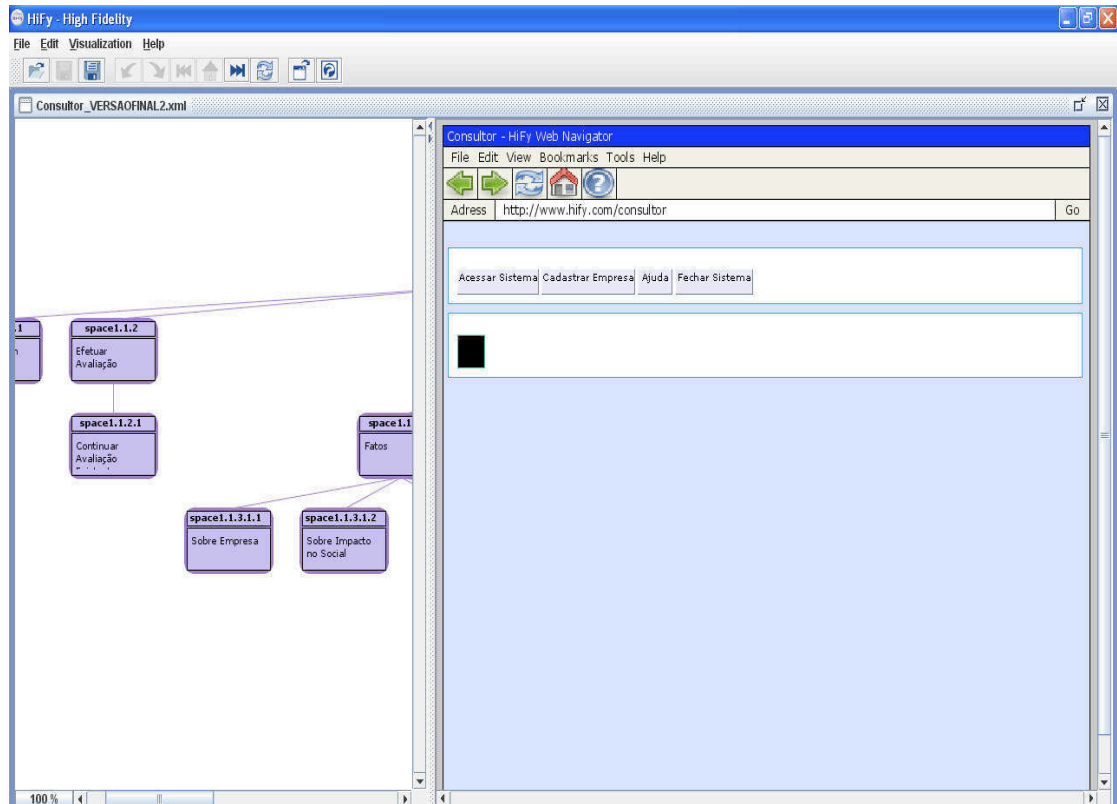


Figura 18 – Modo de Visualização Split

3.5.2. Edições

A função de edição envolve a manipulação direta nos estilos dos elementos interativos dos protótipos de alta fidelidade (*Window, Box, UIElement*). Assim, o sistema deve estar com o foco de visualização no protótipo. Na Figura 19, apresenta-se a edição de um protótipo de alta fidelidade com o uso do Hi-Fy.

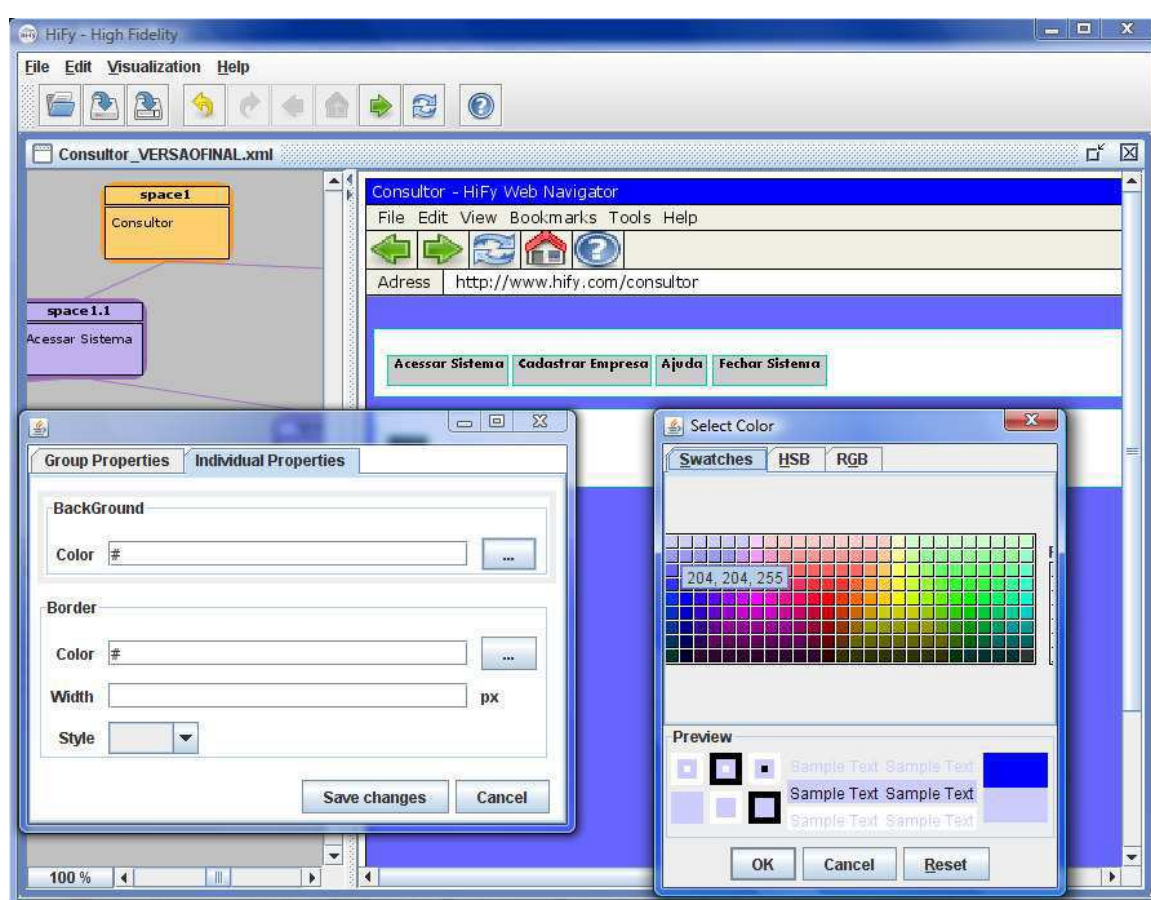


Figura 19 – Edição de Protótipo de Alta Fidelidade em Hi-Fy

3.6. Conclusão

Neste capítulo, foi apresentado o propósito da construção da ferramenta Hi-Fy. Como descrito, parte significativa do projeto de Hi-Fy, que envolveu a apresentação da metodologia de desenvolvimento utilizada e os artefatos produzidos, foi também utilizado no projeto de

construção do ambiente FastInterface. As principais funcionalidades e telas de Hi-Fy foram igualmente apresentadas.

No próximo capítulo, será descrito o projeto de implementação do ambiente de desenvolvimento de interface com usuário FastInterface, incorporando a ferramenta Hi-Fy enquanto módulo gerador de protótipos de alta fidelidade e gerador da interface final.

Capítulo 4

Projeto e Implementação de FastInterface

Este capítulo tem como objetivo apresentar o ambiente de desenvolvimento do usuário, FastInterface, que segue a abordagem baseada em modelos e múltipla prototipagem evolutiva, refletindo os métodos e formalismos propostos pela metodologia MEDITE. Para isto, FastInterface integrará quatro ferramentas de suporte à MEDITE: iTAOS, MAPA, SMILE e Hi-Fy. Aspectos sobre o projeto e implementação do ambiente serão apresentados, bem como, aspectos de integração dos módulos. As principais funcionalidades de FastInterface serão demonstradas através da exibição de algumas de suas telas.

4.1. Propósito de FastInterface

FastInterface é um ambiente de desenvolvimento de interface com o usuário que tem como pressuposto o suporte ao desenvolvimento baseado em modelos e múltipla prototipagem. Para que atinja tal objetivo, FastInterface deve: (i) dar suporte à análise e modelagem da tarefa; (ii) implementar o mapeamento entre os elementos do modelo da tarefa (MT) e os elementos do modelo da interação (interface abstrata - AUI); (iii) permitir a edição e simulação do modelo da interface abstrata a partir da edição e simulação de protótipos de média fidelidade; (iv) implementar o mapeamento entre os elementos do modelo AUI e os elementos do modelo de interface concreta (CUI); (v) permitir edição e simulação do modelo de interface concreta a partir da edição e simulação de protótipos de alta fidelidade; (vi) dar suporte à geração do código da interface final; e, por fim, (vii) oferecer manutenção da consistência entre os modelos envolvidos na concepção da interface com usuário em tempo real de projeto.

FastInterface, além de atender a estas características, objetiva eliminar os dois grandes problemas encontrados atualmente durante o uso de abordagens de desenvolvimento de interfaces baseadas em modelos, que são: (i) o uso de modelos não está associado a uma representação visual, isto faz com que não seja possível vislumbrar como os modelos afetam na apresentação visual da interface final em fases mais avançadas do projeto; e, (ii) o uso exclusivo de modelos declarativos nas fases iniciais e intermediárias do processo implica necessidade de se aplicar técnicas de avaliação da interface final em cima desse tipo de modelo, fazendo com que a avaliação seja adiada para os estágios finais do processo de desenvolvimento da interface.

Assim, o uso do ambiente FastInterface deverá favorecer a adoção da abordagem de desenvolvimento baseada em modelos, em especial o uso efetivo de MEDITE, pois os projetistas passarão a ter todo o processo de concepção da interface com usuário suportado por um único ambiente que abrange todas as ferramentas de suporte às suas fases, e usufruindo das vantagens do uso evolutivo de múltiplos protótipos.

4.2. Projeto e Implementação do Ambiente FastInterface

Como descrito anteriormente, o ambiente FastInterface foi desenvolvido seguindo a metodologia YP em consonância com MEDITE. Alguns artefatos sugeridos por YP e produzidos no projeto de FastInterface foram apresentados no capítulo anterior, a saber: Requisitos Não-Funcionais; Perfil do Usuário; Objetivos de Usabilidade; Modelo da Tarefa; e, *User Stories* e Testes de Aceitação. O restante dos artefatos que são inerentes ao ambiente FastInterface serão apresentados nas próximas subseções.

4.2.1. Requisitos Funcionais

As principais funcionalidades de FastInterface são: (i) dar suporte à análise e modelagem da tarefa através da incorporação e disponibilização do módulo iTAOS; (ii) implementar, através da incorporação do módulo MAPA, o mapeamento entre os elementos do modelo da tarefa (MT) e os elementos do modelo da interação (interface abstrata - AUI); (iii) permitir a edição e simulação do modelo da interface abstrata a partir da edição e simulação de protótipos de média fidelidade, através da incorporação do módulo SMILE; (iv) dar suporte à geração e

edição de protótipos de alta fidelidade da interface e (v) à geração do código da interface, através da incorporação do módulo Hi-Fy; (vi) oferecer manutenção da consistência entre os modelos envolvidos na concepção da interface com usuário em tempo real de projeto; e, por fim, (vii) possibilitar a concepção de múltiplos projetos simultaneamente.

4.2.2. Protótipo da interface

A criação do protótipo de FastInterface englobou a criação de um *split pane* o qual divide a área de manipulação de projetos e a área de edição de modelos. A área de manipulação de projetos apresenta todos os projetos carregados. Por sua vez, a área de edição de modelos contempla os módulos envolvidos no processo de concepção. Os módulos são acessados com o auxílio de abas, que dividem a edição dos diferentes modelos. Na Figura 20, apresenta-se o protótipo de alta fidelidade produzido para FastInterface.

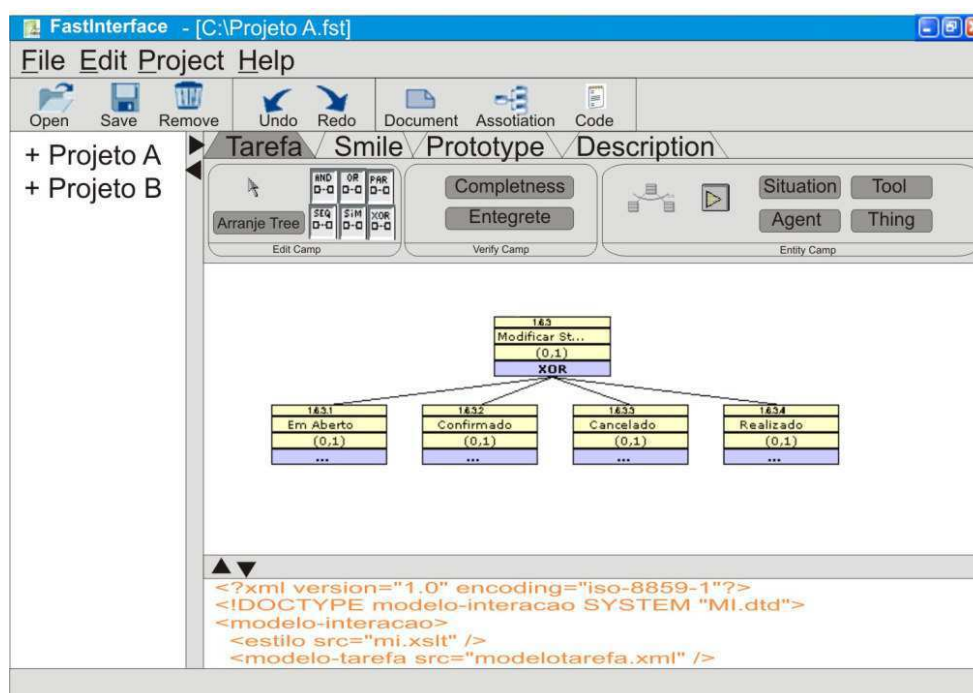


Figura 20 – Protótipo de alta fidelidade de FastInterface

4.2.3. Projeto arquitetural

Como descrito anteriormente, a arquitetura de FastInterface segue o padrão MVC. As camadas que foram definidas em seu projeto são: Camada de Dados (*data*), Camada de Negócio (*business*) e Camada de Apresentação (*gui*).

Como FastInterface trabalha com o conceito de Projeto, que engloba todos os modelos envolvidos na concepção da interface com usuário, sua Camada de Negócio integra e manipula todos os outros módulos ora apresentados. O esquema de integração dos módulos com as camadas do FastInterface é apresentado na Figura 21.

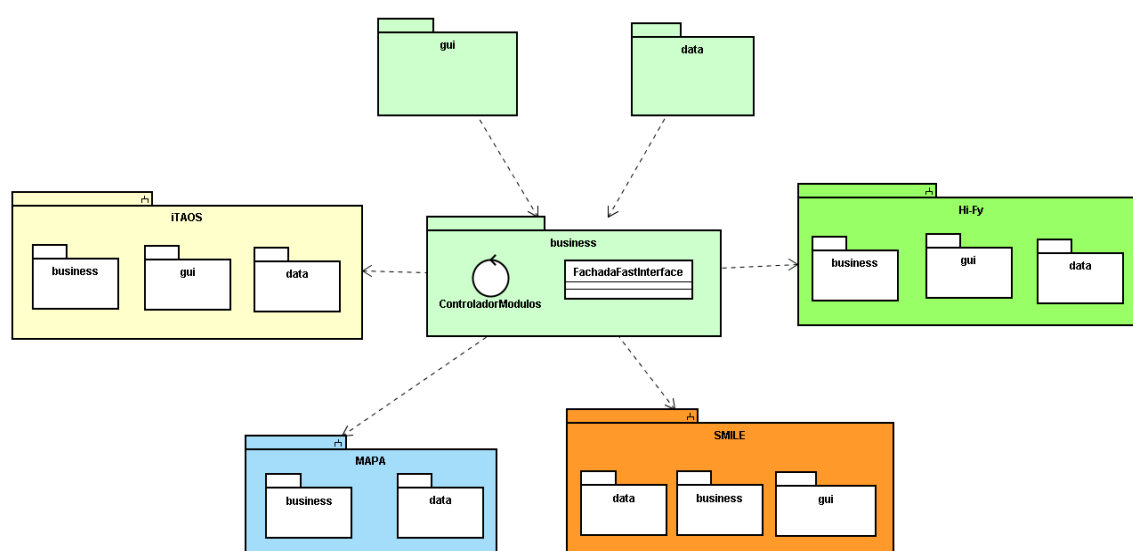


Figura 21 – Interação entre camadas e módulos de FastInterface

Camada de Apresentação

A camada de apresentação do FastInterface foi feita com o auxílio do *framework* gráfico JHotDraw uma vez ele facilita a criação da tela principal com seus menus e barra de ferramentas e também pela necessidade de integração da camada gráfica das ferramentas iTAOS, SMILE e Hi-Fy que usavam o *framework* nas suas camadas gráficas.

As relações de composição, herança e implementação de interfaces entre as classes e interfaces de JHotDraw e FastInterface são detalhadas na Figura 22. Abaixo, uma descrição dos principais componentes de aplicação e de projeto implementados em FastInterface.

- FASTApplication: basicamente composta pelo menu principal, barra de ferramentas e *split pane*. Constitui a janela principal da aplicação.
- FASTApplicationModel: é o modelo da aplicação que determina, dinamicamente, quais os tipos de projetos ela cria e suporta, além de inicializar os menus e barra de ferramentas.

- FASTProject: é o tipo de projeto da aplicação (projeto referente a JHD, não relacionado à classe do modelo Projeto).

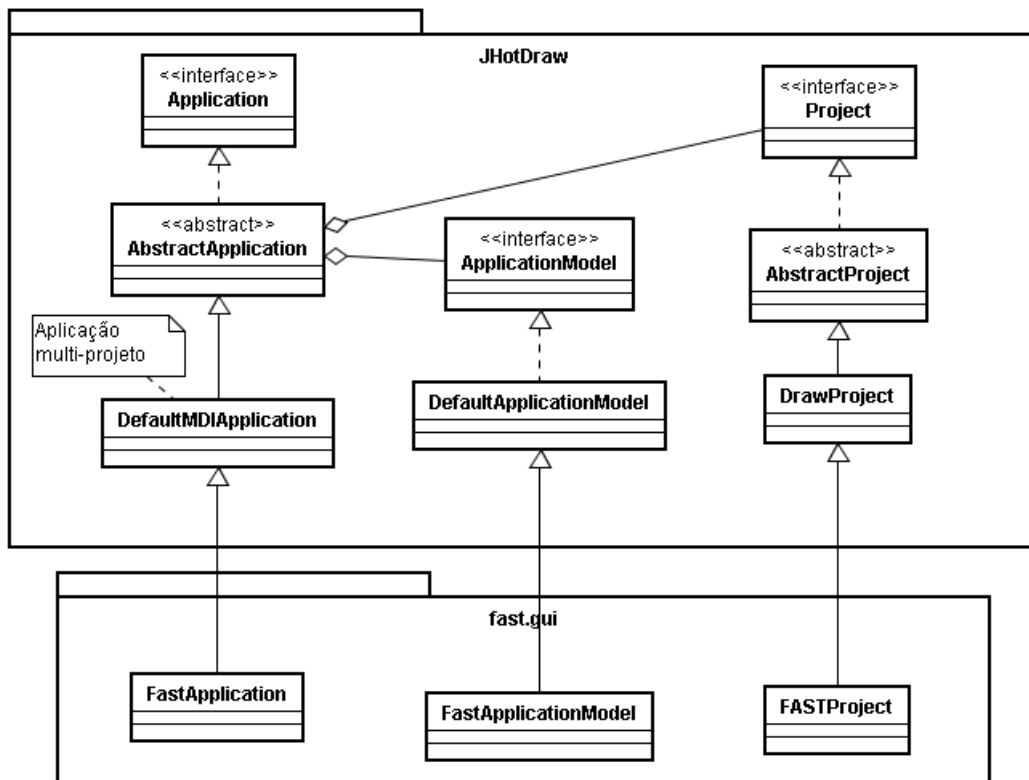


Figura 22 – Relação entre JHD e FastInterface

Além do *framework* JHotDraw, a implementação da Camada Gráfica foi feita utilizando a API de componentes gráficos Java Swing. O código desta camada ficou no pacote *fast.gui* e subpacotes. Os componentes Java Swing foram usados para criar a árvore de projetos e os painéis de abas.

Camada de Negócio

Como já foi mencionado anteriormente, a Camada de Regras de Negócio, ou Camada de Controle, é responsável pelo processamento dos dados. Toda comunicação entre Apresentação e Modelo deve passar pela Camada de Controle para que esta faça o tratamento necessário e verifique a consistência e coerência entre os dados.

Em FastInterface, foi criada uma classe denominada FachadaFastInterface que faz toda a manipulação na coleção de projetos. Sempre que uma classe necessita criar um projeto,

salvá-lo ou fazer alguma alteração nele, ou ainda, trabalhar com toda a coleção, deve fazer suas requisições a FachadaFastInterface. Esta classe foi inserida no pacote *fast.business*.

Da mesma forma que havia a necessidade de uma classe que intermediasse a comunicação com o modelo, era necessário também ter quem desempenhasse o mesmo trabalho em relação os módulos externos do FastInterface. Foi criada, assim, a classe ControladorFastInterface, responsável por abstrair e facilitar a comunicação com os diferentes módulos. Esta classe está presente no pacote *fast.business.controlador*.

Na Figura 23, apresenta-se um diagrama conceitual de classes contendo as supracitadas e seus relacionamentos.

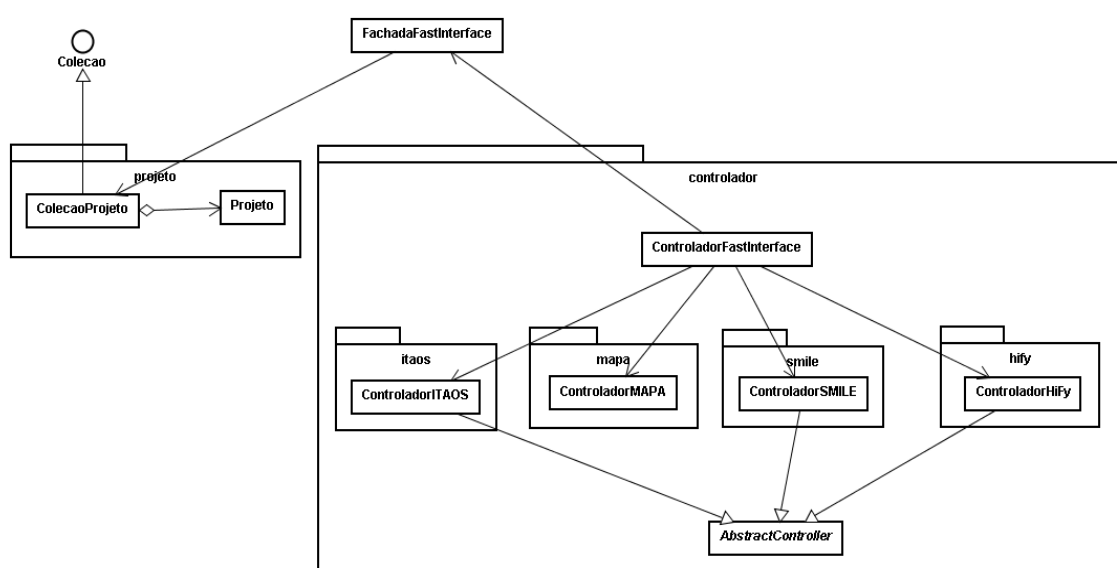


Figura 23 – Diagrama conceitual de classes do pacote *fast.business*

Camada de Dados

A implementação da Camada de Persistência de FastInterface fez uso da API JDOM e é constituída por:

- gerenciador de entrada/saída com formatação da saída, implementado na classe InputOutputManager;
- validador do FastInterfaceXML (descrito adiante), classe XMLValidator;
- objeto de acesso à dados do projeto que utiliza o gerenciador e o validador, classe ProjetoDAO;

Na Figura 24, é apresentado um diagrama de classes com os relacionamentos entre as classes da API JDOM e FastInterface.

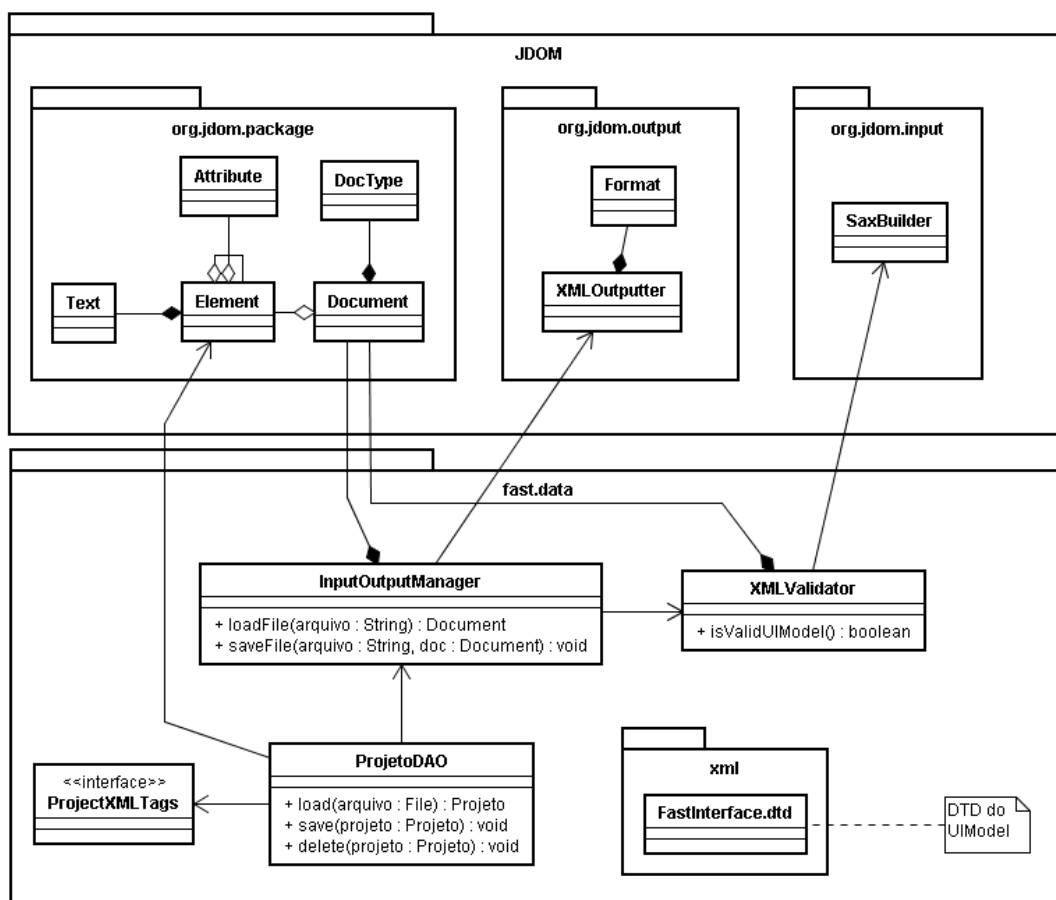


Figura 24 – Relação entre JDOM e Camada de Persistência de *FastInterface*

4.3. Aspectos de Integração

O *FastInterface* possui segmentos gerenciais que permitem a manipulação dos diferentes módulos sem ferir o encapsulamento e visibilidade das principais atividades de sua execução. O contexto de uso do ambiente leva em consideração a criação de projetos, que englobam os diversos modelos envolvidos; apresentação gráfica dos módulos, que provê recursos de manipulação nos modelos; e, por fim, a mudança de abas, que remete ao salvamento e aplicação de regras de transformação e manutenção da consistência entre os modelos. Dentre os segmentos, pode-se destacar:

- O Controlador de Módulos (CM): centraliza as chamadas às rotinas de manipulação dos modelos envolvidos no desenvolvimento da interface pelos diferentes módulos dado um determinado projeto;

- O Controlador de Abas (CA): segmento que gerencia a garantia do *layout* padrão dos módulos, feitos pelo *FactoryPanel* das Abas, onde através do contato com CM será possível realizar a habilitação de cada aba com sua respectiva *toolbar* e área de manipulação, *Project* de cada módulo;
- O Controlador de Árvore de Projeto é responsável pela construção do *Project Explore (GUI)* e conectar-se com os controladores dos módulos.
- A Fachada GUI: responsável por oferecer as principais funções de interação que são primordiais para manipulação dos modelos no FastInterface.
- O Controlador de Projetos (CP): módulo que agrega as regras de negócio de gerenciamento dos projetos criados e abertos;

4.3.1. Controle da Interface Gráfica

O controle da interface gráfica do FastInterface com relação aos aspectos de integração envolve o uso de alguns módulos gerenciais do ambiente, entre os quais: (i) Controlador de Módulos, (ii) Controlador de Abas e (iii) Controlador de Árvore de Projetos; (iv) FachadaGUI.

O **Controlador de Módulos** engloba o conjunto de controladores de módulos da aplicação e o *ControladorFastInterface*, que os acessa. Ao se acionar a criação de um novo projeto, o *ControladorFastInterface* guarda a referência do objeto Projeto recém construído e cria os *Projects (JHotDraw)* para cada módulo acessado, fazendo uma ligação entre o Projeto e o *Project (JHotDraw)* dos módulos. A partir de então, todas as vezes que se fizer uma troca de aba, o **Controlador de Abas** verifica qual aba foi acessada e faz uma requisição ao controlador correspondente ao modelo acessado.

O **Controlador de Árvore de Projetos** dá acesso aos projetos carregados pela aplicação. Desta forma, quando o usuário pretende acessar algum modelo de algum projeto, o **Controlador de Árvore de Projetos**, que manipula o *Project Explorer (GUI)*, faz acesso ao **Controlador de Abas**. Este, conseqüentemente, acessa o **Controlador de Módulos**. Todos os passos resultam na abertura ou ativação de uma aba com o modelo disponível para edição. Na Figura 25, apresenta-se o esquema do Controlador de Módulos.

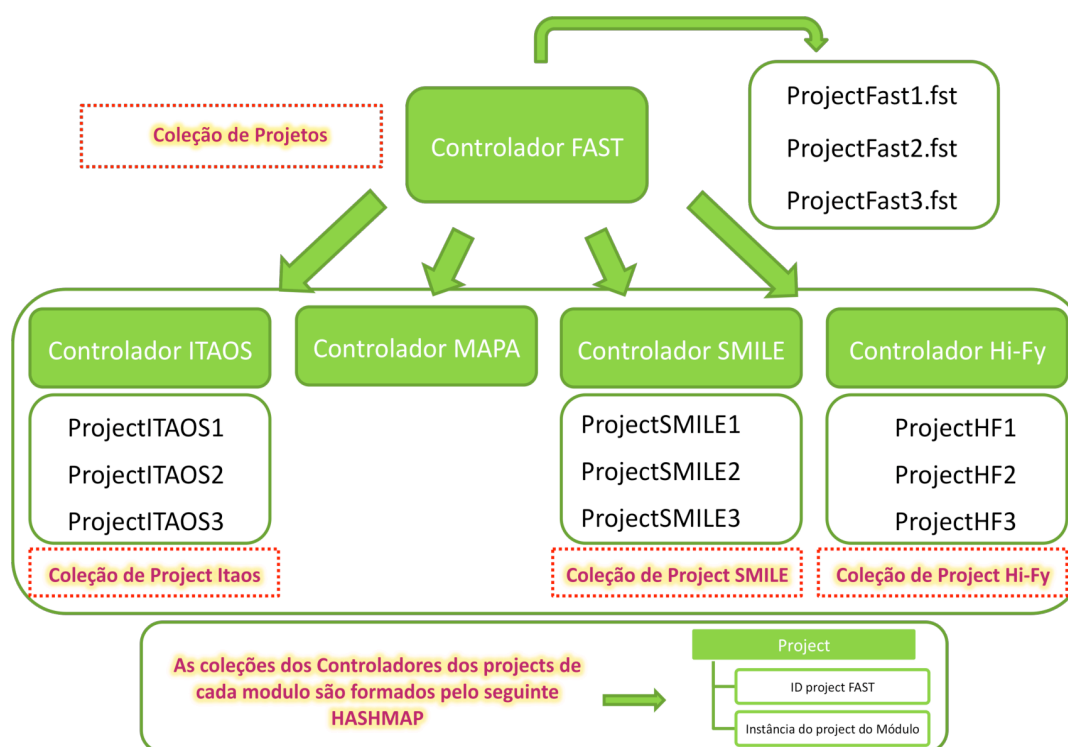


Figura 25– Estrutura do Controlador de Módulos

Ao se realizar a troca de abas, é realizado acionamento das ações de salvamento e manutenção da consistência. Estas ações estão disponíveis em uma fachada de integração a qual deve estar presente em todos os módulos que possuem interface gráfica.

Em relação à fachada de integração, esta deve implementar um conjunto de métodos definidos na documentação de integração, entre eles estão:

- `registerActions()`: registra as ações, uma vez que o método `init()` de `Application` não pode ser chamado;
- `getToolBar()`: retorna um objeto do tipo `JToolBar`, que é a barra de ferramentas do módulo adaptada para ser inserida no `FastInterface`;
- `startGUI()`: inicializa a Camada Gráfica do módulo;
- `openProject(File arquivo)`: retorna um objeto do tipo `Project`, que é a representação gráfica do modelo de trabalho do módulo passado no arquivo;
- `saveProject(Project project)`: salva o modelo representado pelo `Project` passado como parâmetro;

Estas manipulações ficam a cargo do módulo gerencial, denominado **FachadaGUI**. As operações de manutenção da consistência são disponibilizadas pelos módulos que possuem esta competência.

4.3.2. Persistência e Transformação entre os Modelos

Os modelos ficam armazenados em um repositório que utiliza o formato XML e podem ser manipulados por diferentes ferramentas durante o estágio de geração da interface. Desta forma, foi definida uma linguagem de descrição do modelo de interface com usuário o qual possui informações sobre o projeto, modelo da tarefa, modelo de interação (AUI) e modelo CUI. Esta linguagem foi estruturada com base nos atributos inerentes dos formalismos utilizados em MEDITE e é um padrão que foi denominado **FastInterfaceXML**.

Segundo o padrão FastInterfaceXML, cada arquivo deve conter um elemento pai *uimodel*, cujos filhos são:

- *project*, contendo informações do projeto;
- *task-model*: contendo o modelo da tarefa;
- *interaction-model*: contendo o modelo da interação;
- *cui-model*: contendo o modelo da interface concreta do usuário;

A Figura 26 ilustra como ficou a divisão do arquivo e a responsabilidade de cada módulos.

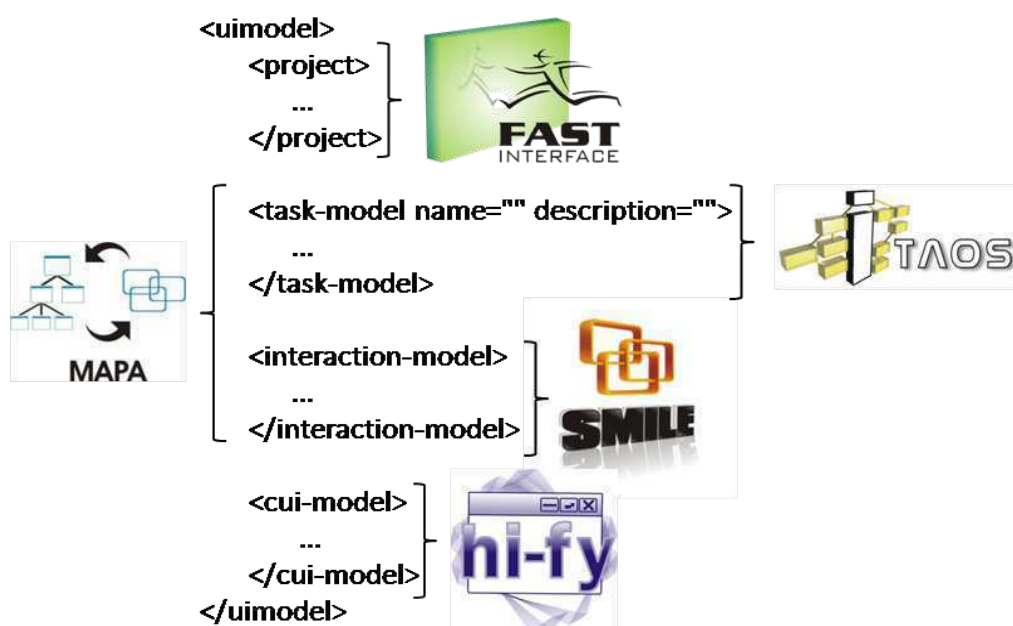


Figura 26 – Organização dos arquivos do tipo FastInterfaceXML

4.3.3. Mecanismo de Manutenção da Consistência entre os Modelos

Nas próximas sub-seções, será apresentado o mecanismo de manutenção da consistência entre os modelos utilizados pelo ambiente FastInterface.

4.3.3.1. Consistência entre o Modelo da Tarefa e Modelo de Interface Abstrata com o Usuário

O mecanismo de geração do modelo de interface abstrata do usuário a partir do modelo da tarefa que foi apresentado no capítulo 2, é informalmente denominado “*mecanismo de ida*” e remete à consistência entre o modelo da tarefa e o modelo da interface abstrata. O “*mecanismo de volta*”, consistência entre o modelo de interface abstrata com o modelo da tarefa, é realizado por uma série de passos que é apresentada no Quadro 9.

Quadro 9 – Seqüência de ações de manutenção da consistência AUI - MT

Módulo Responsável	Passo	Atividade
SMILE	1	Captura das ações realizadas nos protótipos de média fidelidade
	2	Persistência da pilha operações de volta no arquivo FastInterfaceXML
MAPA	3	Leitura da pilha de operações realizadas
	4	Inclusão e remoção de tarefas e ações do modelo da tarefa previamente modelado baseadas nas operações estabelecidas
	5	Remoção da pilha de operações do arquivo XML e salvamento do novo modelo da tarefa

O modelo da interação (*AUI model*) é composto por informações sobre a apresentação, navegação e operações realizadas. Esta última informação diz respeito a pilha de operações descrita no Quadro 9.

As operações realizadas no modelo de interação são persistidas de acordo com a especificação que engloba operações pré-definidas de inclusão e remoção de elementos interativos. A especificação das operações de manutenção da consistência AUI – MT, com exemplo de seu uso, é apresentada no Quadro 10.

Quadro 10 - Operações de Manutenção da Consistência AUI – MT

Elemento de Interação	Operação	Tag XML		
Objeto de Interação	Inserir	<operation name="INSERT_IO" elementName="Unnamed OI" fatherspaceID="NEWspace1.2" viewID="NEWview1.2.1" brotherID="NEWio1.2.1.1" ocorrencia="(0,1)" side="right" classification="interaction" />		
	Remover	<operation name="REMOVE_IO" elementID="NEWio1.2.1.2" viewID="NEWview1.2.1" />		
Visão	Inserir	<operation name="INSERT_VIEW" elementName="Unnamed View" elementID="view1.1.5" fatherspaceID="space1.1" brotherID="view1.1.4" ocorrencia="(0,1)" side="right" classification="interaction" defaultIoName="Unnamed OI"/>		
	Remover	<operation name="REMOVE_VIEW" elementID="view1.2.2" fatherspaceID="space1.1" />		
Espaço	insertir	em espaço inicial	<operation name="INSERT_INTERACTIONSSPACE_SON_INITIALSPACE" elementName="New Space" elementID="space1.2" ocorrencia="(0,1)" brotherID="space1.1" side="right" defaultViewName="Unnamed View" defaultIoName="Unnamed OI" />	
		em espaço de interação	<operation name="INSERT_INTERACTIONSSPACE_SON_INTERACTIONSSPACE" elementName="Unnamed Space" elementID="space1.1.3" fatherspaceID="space1.1" ocorrencia="(0,1)" brotherID="space1.1.2" side="right" defaultViewName="Unnamed View" defaultIoName="Unnamed OI" />	
		em espaço de direcionamento	<operation name="INSERT_INTERACTIONSSPACE_SON_DIRECTIONSPACE" elementName="Unnamed Space" elementID="space1.3.2" fatherspaceID="space1.3" ocorrencia="(0,1)" brotherID="space1.3.1" side="right" defaultViewName="Unnamed View" defaultIoName="Unnamed OI" />	
	dir	filho do inicial	<operation name="INSERT_DIRECTIONSPACE_SON_INITIALSPACE" elementName="Unnamed Space" elementID="space1.1" defaultViewName="Unnamed View" defaultIoName="Unnamed OI" />	
		filho de espaço de direcionamento	<operation name="INSERT_DIRECTIONSPACE_SON_DIRECTIONSPACE" elementName="Unnamed Space" elementID="space1.1.1" fatherspaceID="space1.1" defaultViewName="Unnamed View" defaultIoName="Unnamed OI" />	
	remover	int	filho do espaço inicial	<operation name="REMOVE_INTERACTIONSSPACE_SON_INITIALSPACE" elementID="space1.2" />
			filho de espaço de interação	<operation name="REMOVE_INTERACTIONSSPACE_SON_INTERACTIONSSPACE" elementID="space1.2" />
			filho de espaço de direcionamento	<operation name="REMOVE_INTERACTIONSSPACE_SON_DIRECTIONSPACE" elementID="space1.1.2.2"/>
		dir	filho de espaço inicial	<operation name="REMOVE_DIRECTIONSPACE_SON_INITIALSPACE" elementID="space1.1"/>
			filho de espaço de direcionamento	<operation name="REMOVE_DIRECTIONSPACE_SON_DIRECTIONSPACE" elementID="space1.1"/>

4.3.3.2. Consistência entre o Modelo de Interface Abstrata do Usuário e o Modelo de Interface Concreta do Usuário

O mecanismo de consistência entre o modelo de interface abstrata e o modelo da interface concreta só ocorre em um sentido: AUI – CUI. Até então, as operações realizadas no nível CUI não necessitam de um mapeamento prévio no nível AUI uma vez que só são permitidas edições de estilo e fonte dos elementos interativos. A função de geração do modelo de CUI a partir do modelo AUI descrita no capítulo 2 já remete à coerência entre estes modelos.

4.4. FastInterface

As sub-seções a seguir têm por objetivo apresentar as principais funcionalidades do ambiente FastInterface.

4.4.1. Múltiplos Projetos

Com FastInterface é possível trabalhar com múltiplos projetos simultaneamente. Deste modo, estando abertos ou não, os projetos carregados são apresentados na área de trabalho em uma árvore. Os nós dos projetos abertos têm folhas que, ao serem ativadas, abrem a aba do módulo correspondente ou a aba de descrição do projeto, como mostrado na Figura 27 .

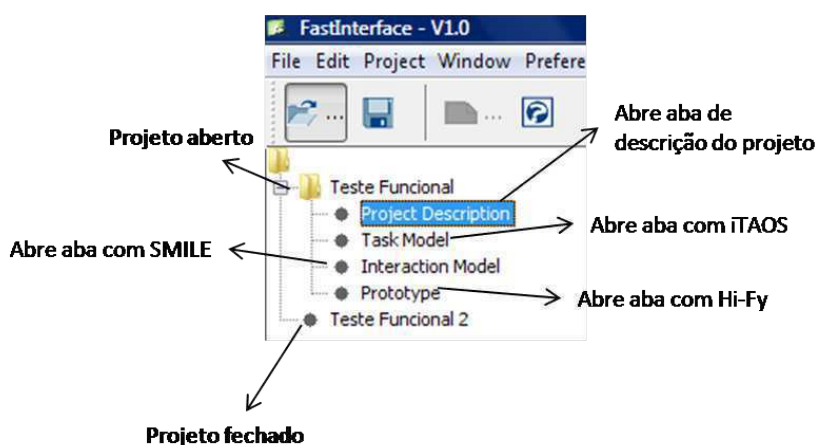


Figura 27 – Árvore de projetos em *FastInterface*

4.4.2. Edições em Modelos

Os modelos envolvidos na concepção de interface com usuário segundo MEDITE e disponibilizados pelo ambiente FastInterface podem ser vistos de maneira gráfica pelos módulos responsáveis, tais como: Modelo da Tarefa, cuja parte gráfica é disponibilizada pelo módulo iTAOS; Modelo da Interação, cuja parte gráfica é disponibilizada pelo módulo SMILE; Modelo da Interface Concreta, cuja parte gráfica é disponibilizada pelo módulo Hi-Fy.

Todas as manipulações realizadas pelos usuários nos modelos ficam a cargo dos módulos que disponibilizam uma *toolbar* (com suas principais funcionalidades de edição) e uma área de edição – *Project* – (onde acontecem propriamente ditas as ações de edição em cima dos modelos). Na Figura 28, apresentam-se os componentes disponibilizados pelos módulos ao ambiente com relação à edição gráfica.

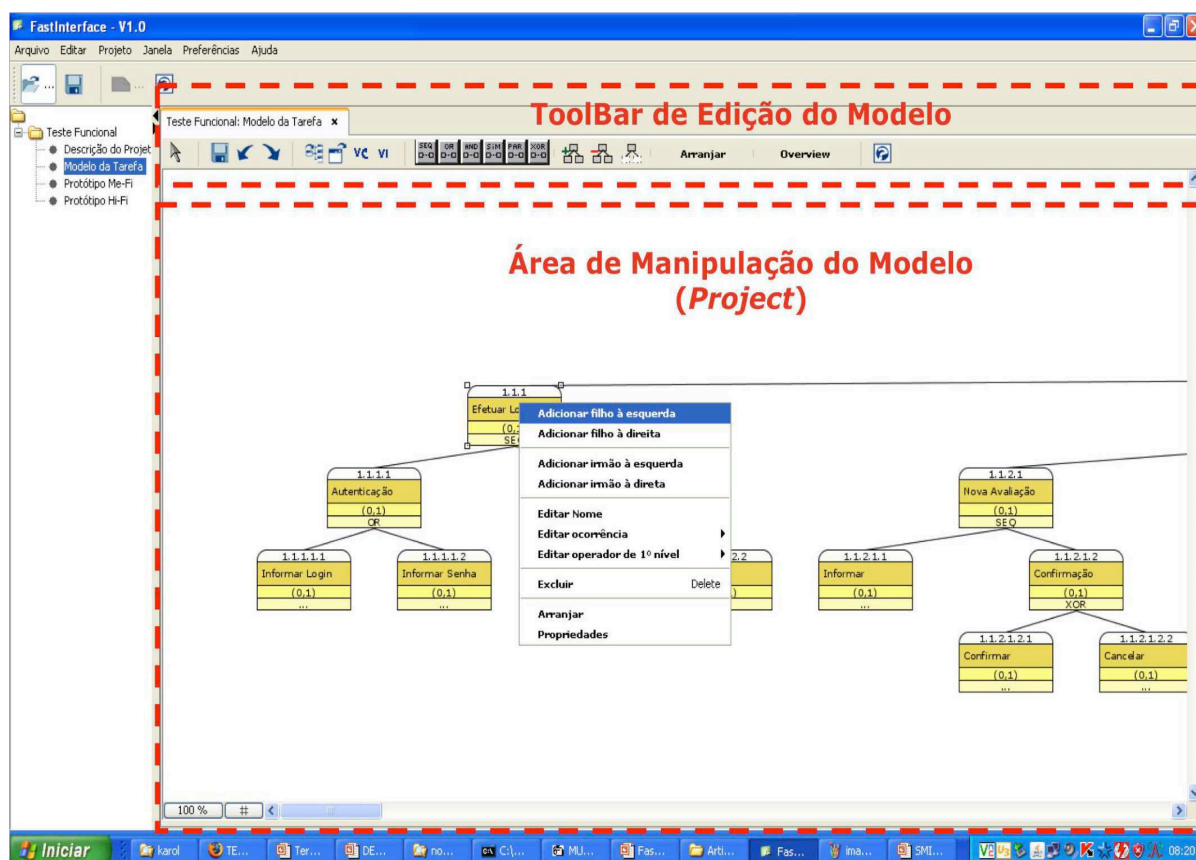


Figura 28 – Área de Edição de Modelos no FastInterface

4.4.3. Manutenção da Consistência entre os Modelos

A manutenção da consistência dentro do ambiente FastInterface acontece na troca de abas, momento em que é acionado o salvamento e operações de transformação e coerência nos modelos. Na Figura 29, evidencia-se a seqüência de passos de geração e transformação entre os modelos sob a perspectiva da troca de abas: (i) Modelagem da Tarefa; (ii) Transformação MT-AUI e AUI-MT (troca de abas); (iii) Manipulação no modelo AUI apresentado sob a forma de protótipo de média fidelidade; (iv) Manipulação no modelo CUI apresentado sob a forma de protótipo de alta fidelidade; e. (v) Geração da FUI.

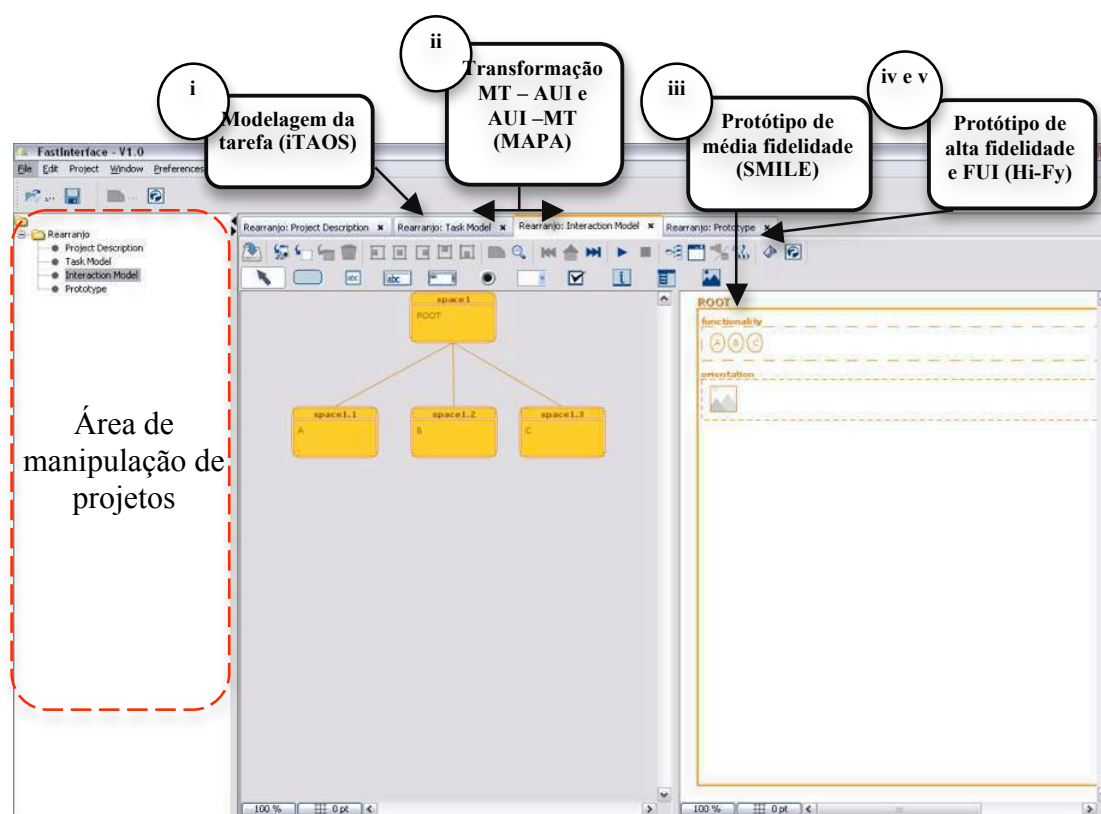


Figura 29 - Ambiente *FastInterface* e as abas dos módulos

4.4.4. Modos de Visualização

Os projetos podem ser vistos sob 2 perspectivas diferentes: (i) no modo de janela de abas externa; e, (ii) no modo de abas internas à janela principal. No primeiro caso, todos os modelos pertencentes a um projeto ficam agrupados em uma única janela, neste caso, a evidência é no projeto. No segundo caso, os modelos são abertos internamente ao ambiente, neste caso, a evidência é nas abas.

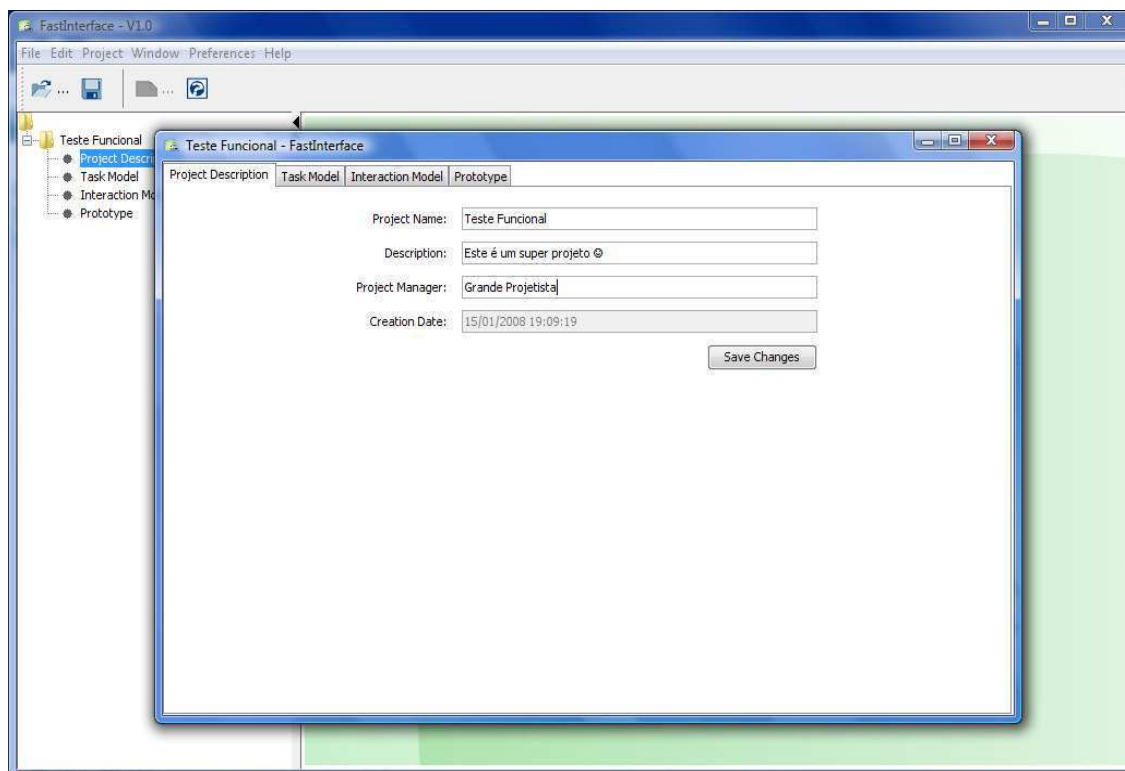


Figura 30 – Projeto aberto no modo de janela de abas externa

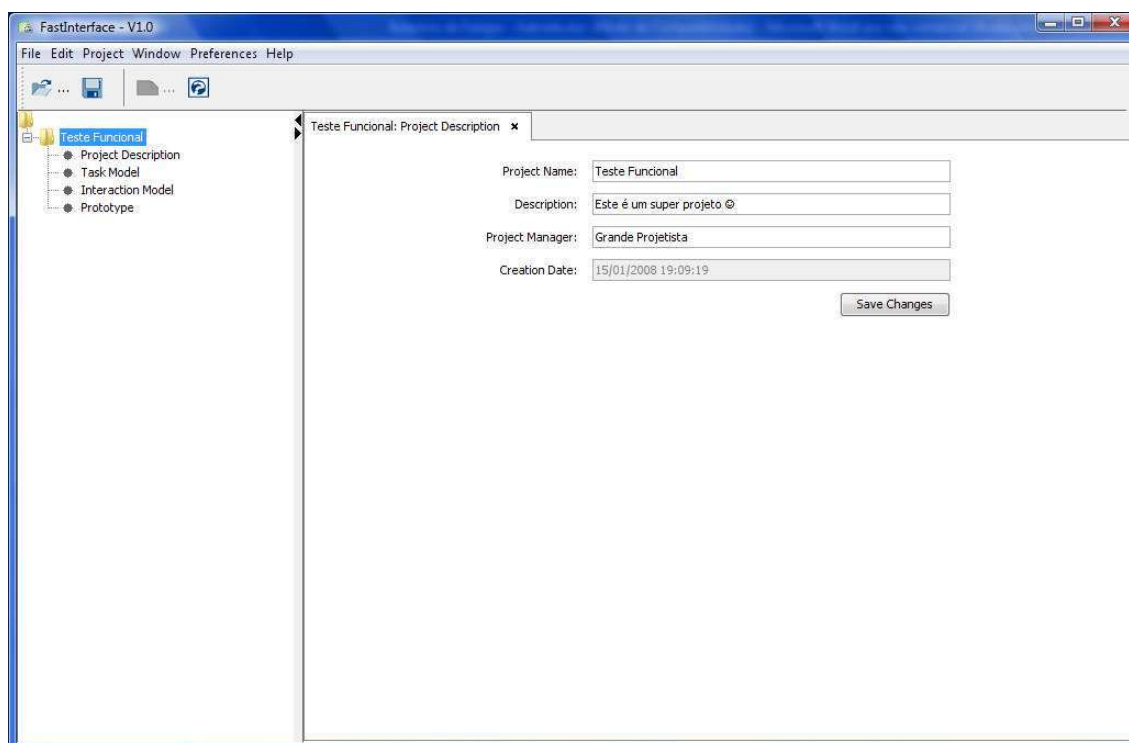


Figura 31 – Projeto aberto no modo de abas internas à janela principal

4.5. Conclusões

Neste capítulo foi apresentado o propósito da construção do ambiente FastInterface. Como visto, boa parte dos artefatos inerentes do projeto de FastInterface foi apresentado no capítulo que descrevia a ferramenta Hi-Fy, uma vez que ambos foram desenvolvidos dentro de um mesmo contexto.

As principais particularidades relacionadas ao projeto de FastInterface estão diretamente ligadas ao seu projeto arquitetural e aspectos de integração dos módulos e modelos provenientes da metodologia MEDITE. Para esse caso, foram demonstrados os controladores gráficos, aspectos de persistência e transformações entre os diferentes modelos e os mecanismos de manutenção da consistência existentes. Por fim, as principais funcionalidades de FastInterface foram apresentadas através da ilustração de algumas telas do sistema.

A fim de validar o ambiente FastInterface enquanto ferramenta de apoio ao processo de desenvolvimento de interface com usuário baseado em modelos dentro de um processo evolutivo que considera as diferentes técnicas de prototipagem, será apresentado no Capítulo 5 um estudo experimental de seu uso.

Capítulo 5

Estudo Experimental sobre o Uso do Ambiente FastInterface

A fim de evidenciar os ganhos de uso e testar a aplicação do ambiente FastInterface, neste capítulo descrevem-se sucintamente as técnicas de avaliação utilizadas para validá-lo e os resultados obtidos.

5.1. Perspectivas de avaliação

O planejamento avaliatório de FastInterface foi realizado sob três perspectivas: (i) Inspeção de Conformidade, (ii) Observação do Uso do Produto e (iii) Sondagem da Satisfação Subjetiva do Usuário. As diferentes perspectivas de avaliação propiciaram a detecção de problemas e resultados sob vários aspectos, tornando a análise mais completa. O planejamento avaliatório ora mencionado foi baseado na abordagem de avaliação multidimensional concebida por Queiroz (2001).

A **inspeção de conformidade** a um padrão é uma modalidade sistematizada de inspeção da usabilidade, dotada de regras pré-definidas, de forma a propiciar um grau de confiança adequado. A inspeção de um produto tem por objetivo avaliar se os elementos interativos (caixa de diálogo, menus, estrutura de navegação) estão de acordo com princípios de usabilidade pré-estabelecidos. Neste contexto, insere-se o conceito de usabilidade da ISO (*Internacional Organization for Standardization*), como a medida na qual um produto pode ser usado por usuários específicos para atingir metas específicas com eficácia, eficiência e satisfação em um contexto específico de uso (ISO 1998). Dado que a interação de

FastInterface dá-se através de menus, manipulação direta e formulários, a inspeção do ambiente foi feita com relação ao Padrão ISO 9241, com ênfase nas partes 14 – *Diálogos via menus* (ISO, 1997) , 16 – *Diálogo via manipulação direta* (ISO, 1999) e 17 – *Diálogo via Formulários* (ISO, 1998). A metodologia utilizada para a inspeção de conformidade e os resultados obtidos estão descritos na seção 5.2.

A **observação do uso do produto** tem por objetivo mensurar a performance do usuário ao interagir com o produto. Neste caso, examina-se se a interface com usuário satisfaz ou não às necessidades do usuário (Aguiar, 2007). No decorrer do processo avaliatório, é possível identificar falhas de usabilidade e dificuldades encontradas pelos usuários, tanto a partir de indicadores quantitativos quanto de qualitativos. Para FastInterface, realizou-se um **estudo de caso** de modo a permitir a avaliação do ambiente através da técnica de observação direta. A descrição e análise do estudo de caso são apresentadas na seção 5.3.

A **sondagem da satisfação subjetiva** consiste em analisar as respostas dadas por usuários recrutados para um estudo de caso a questões sobre perfil do usuário e à satisfação do usuário ao utilizar um produto. Neste caso, utilizou-se dois tipos de questionários: *USer (User Sketcher)* e *USE (User Satisfaction Enquirer)* (De Oliveira *et. al.*, 2005). O resultado da sondagem da satisfação subjetiva dos usuários com o produto e delineamento do perfil do usuário está descrito na seção 5.4.

5.2. Inspeção de conformidade ao Padrão ISO 9241

O Padrão ISO 9241 define a conformidade de um produto através de duas análises: (i) aplicação de *checklist*, o qual consiste em um conjunto de itens a ser verificado para o produto; (ii) taxa de adoção do itens previstos no *checklist*. O resultado final da avaliação deve conter os itens considerados, as recomendações aplicáveis e as recomendações adotadas.

Para a inspeção de conformidade do ambiente FastInterface, foram consideradas as partes 14, 16 e 17 do Padrão ISO 9241, descritas a seguir:

- **Parte 14 - Diálogos via menus:** contempla um conjunto de recomendações para a estrutura dos menus, a navegação dentro desta estrutura, a seleção e execução de opções de menu;
- **Parte 16 - Diálogos via manipulação direta:** contempla um conjunto de recomendações para o projeto ergonômico de diálogos via manipulação direta, incluindo a manipulação de objetos e o projeto de metáforas, objetos e atributos.
- **Parte 17 – Diálogo via formulários:** contempla um conjunto de recomendações a respeito da estrutura dos formulários, das entradas, do *feed-back* e da navegação pelos campos.

5.2.1. Resultado da Inspeção de Conformidade de FastInterface ao Padrão ISO 9241-14: Diálogos via menus

A inspeção de conformidade de FastInterface à Parte 14 do padrão ISO 9241, referente às recomendações para Diálogo via Menus, revelou um conjunto de **11 falhas**. No Quadro 11, apresentam-se as falhas identificadas e as recomendações correspondentes.

Quadro 11 – Falhas de Conformidade de FastInterface relacionadas ao Padrão ISO 9241-14

Subseção	ID	Descrição	Sugestão
5.2.1	01	O padrão especifica a inserção de um divisor nos agrupamentos lógicos das opções de menu. Desta forma, no menu <i>Arquivo</i> , observa-se a inexistência de uma divisão entre as opções <i>Novo</i> , <i>Abrir</i> e <i>Fechar</i> das opções <i>Salvar</i> , <i>Salvar como</i> e <i>Exportar</i> . A opção <i>Sair</i> também deve ser posicionada em um grupo distinto das demais opções do menu <i>Arquivo</i> .	Conforme é utilizado pela grande maioria das ferramentas <i>desktop</i> conhecidas, além das opções disponibilizadas no menu <i>Arquivo</i> , é pertinente disponibilizar também uma opção que permita o usuário <i>Fechar</i> o projeto aberto.
6.1.1 b)	02	Os títulos utilizados nas opções de menu devem ser compostos por termos relacionáveis.	Recomenda-se que a opção de menu <i>Janela</i> apresente as seguintes opções: <i>Abas Externas</i> e <i>Abas Internas</i> .
7.1.1	03	Deve-se fazer uso de métodos/dispositivos de entrada de dados alternativos para seleção das opções de menu.	Recomenda-se a definição de designadores alfabéticos para cada uma das opções de menu. Apenas as opções <i>Editar</i> e <i>Preferências</i> apresentam designadores, respectivamente as letras E e P.

7.1.3	04	Vide Falha 03 (Subsecção 7.1.1.)	
7.2.1	05	Designadores alfabéticos devem ser utilizados para as opções de menu. Vide Falha 03 (Subsecção 7.1.1.)	
7.2.4	06	Designadores alfabéticos devem ser utilizados para as opções de menu. Vide Falha 03 (Subsecção 7.1.1.)	
7.2.5	07	Designadores alfabéticos devem ser utilizados para as opções de menu. Vide Falha 03 (Subsecção 7.1.1.)	
8.1.7	08	Vide Falha 02 (Subsecção 6.1.1. b))	
8.1.11	09	Vide Falha 03 (Subsecção 7.1.1.)	
8.3.4	10	Vide Falha 02 (Subsecção 6.1.1. b))	
8.3.10	11	O padrão especifica que se uma opção de menu conduz a outro diálogo, então, deve ser fornecida uma indicação visual apropriada. Usualmente, tem sido utilizado as reticências (...) para indicar que uma determinada opção conduz a outro diálogo.	As opções de menu Novo, Abrir, Salvar como e Exportar devem apresentar indicação visual (...).

A partir de uma breve análise do quadro acima é possível perceber a presença de **falhas recorrentes** (falhas 04, 05, 06, 07, 08, 09 e 10), ou seja, a correção de uma falha implica no ajuste de outra falha. Essa característica facilita a correção de um conjunto de falhas sem que seja necessário empenhar tempo demasiado nesta tarefa. Avaliando o impacto dessas correções, pode-se considerar que todas as falhas apresentadas são de baixo impacto e fáceis de serem corrigidas.

5.2.2. Resultado da Inspeção de Conformidade de FastInterface ao Padrão ISO 9241-16: Diálogos via manipulação direta

A inspeção de conformidade de FastInterface à Parte 16 do Padrão ISO 9241, referente às recomendações para *Diálogo via Manipulação Direta*, revelou um conjunto de **05 falhas**. O **Quadro 12** apresenta as falhas detectadas.

Assim como para as falhas apresentadas na seção anterior, as falhas detectadas nesta seção são de baixo impacto e não comprometem a interação do usuário com o sistema. A

recomendação associada à falha 05 está relacionada à falha 04, podendo este ser considerado um caso de falha recorrente.

Quadro 12 – Falhas de Conformidade de FastInterface relacionadas ao Padrão ISO 9241-16

Subseção	ID	Descrição
5.2.3	01	Uma vez que o botão de ajuda disponível na barra de ferramentas do FastInterface não executa nenhuma ação, recomenda-se que esta opção seja excluída da barra de ferramentas ou que seja exibida com indicação visual de indisponibilidade.
5.3.4	02	Recomenda-se que seja fornecido <i>feedback</i> contínuo e imediato durante a manipulação direta de objetos/ícones. No entanto, percebe-se que nenhum retorno é fornecido ao usuário, de maneira contínua e imediata, ao clicar duas vezes sobre um objeto do <i>Protótipo Hi-Fi</i> .
6.2.9	03	Recomenda-se que sejam fornecidas teclas de atalho sempre que adequado ao nível de experiência dos usuários. Portanto, recomenda-se que sejam fornecidos mecanismos de atalhos para as principais funções/ícones disponíveis na barra de ferramentas da janela <i>Protótipo Hi-Fi</i> , tais como: <i>Previous Window</i> , <i>Initial Window</i> , <i>Next Window</i> e <i>Refresh</i> .
8.1.2	04	Recomenda-se que as barras de rolagem movam-se automaticamente de acordo com a movimentação do conteúdo na janela. Por exemplo, ao mover um <i>space</i> com a seta para direita do teclado, quando tal <i>space</i> ultrapassar os limites visíveis da janela, a barra de rolagem deve mover-se automaticamente de maneira a acompanhar a movimentação do objeto.
	05	Recomenda-se que as barras de rolagem movam-se automaticamente de acordo com a seleção de objetos na janela. Por exemplo, ao selecionar um grupo de objetos em uma janela do <i>Protótipo Hi-Fi</i> , quando esta seleção extrapolar os limites visíveis da janela, a barra de rolagem deve mover-se automaticamente de maneira a acompanhar a seleção dos objetos.

5.2.3. Resultado da inspeção de conformidade de FastInterface ao Padrão ISO 9241-17: Diálogos via Formulários

A inspeção de conformidade de FastInterface à Parte 17 do Padrão ISO 9241, referente às recomendações para *Diálogo via Formulários*, revelou um conjunto de **06 falhas**. O **Quadro 13** apresenta as falhas detectadas.

As falhas mais críticas se referem a problemas relacionados a internacionalização da ferramenta. O fato de se trabalhar com vários módulos integrados fez com que houvesse conflito de nomenclatura. As demais falhas são tidas como de menor impacto na interação do usuário com o sistema.

Quadro 13 – Falhas de Conformidade de FastInterface relacionadas ao Padrão ISO 9241-17

Subseção	ID	Descrição
5.1.1	01	O formulário para criação de um novo projeto é exibido com o título do formulário (caixa de diálogo) em Inglês, apesar de o idioma selecionado ser o Português. Esta falha é recorrente em outras janelas da ferramenta avaliada.
5.2.7	02	Recomenda-se que o alinhamento do formulário da caixa de diálogo <i>Novo Projeto</i> seja da seguinte maneira: rótulos alinhados à direita e campos de entrada de dados alinhados à esquerda, conforme pode ser visto na aba <i>Descrição do Projeto</i> .
5.3.1	03	Ao tentar criar um projeto cujo nome do projeto era composto por 630 caracteres, o sistema apresentou a seguinte mensagem “ <i>Project file could not be saved</i> ”. Recomenda-se que, se há um limite para a entrada de dados, este limite deve ser explícito para o usuário, como por exemplo, delimitar a quantidade máxima de caracteres que podem ser digitados. Além desta falha, percebe-se, mais uma vez, que apesar da ferramenta estar com o idioma selecionado Português, a mensagem foi exibida em Inglês.
5.3.2	04	Recomenda-se que os campos obrigatórios de formulário sejam facilmente distinguíveis daqueles campos opcionais. No entanto, percebe-se que no formulário para criação de um novo projeto não há nenhuma indicação para distinguir os campos obrigatórios (<i>Nome do Projeto</i> e <i>Caminho do Projeto</i>) daqueles opcionais (<i>Projetista Responsável</i> e <i>Descrição</i>).
6.1.3	05	Recomenda-se que sejam utilizados valores <i>default</i> onde for apropriado para a tarefa. Portanto, recomenda-se que o campo <i>Caminho do Projeto</i> apresente como valor default o último caminho selecionado pelo usuário ao criar um projeto.
6.4.2 a)	06	Recomenda-se que, se a validação de um formulário encontrar erros, o cursor deve ser posicionado no campo com o erro. Este comportamento não é observado na ferramenta quando há erros nos campos <i>Nome do Projeto</i> ou <i>Caminho do Projeto</i> no formulário de <i>Novo Projeto</i> .

5.2.4. Taxas de adoção

Dada a Tabela 1, verifica-se que FastInterface conseguiu taxas de adoção ao Padrão ISO 9241 extremamente positivas, principalmente no que diz respeito à Parte 16, **88,88%**. As demais taxas, da parte 14 e 17, ficaram acima de **81%**. Além do mais, boa parte das falhas relacionadas à parte 14 são **falhas recorrentes**.

Tabela 1– Taxas de Adoção

Padrão	#P	#S	TA(%)
ISO 9241-14	47	58	81,03
ISO 9241-16	40	45	88,88
ISO 9241-17	26	32	81,25

Legenda: P – Recomendações adotadas pelo produto
 S – Recomendações aplicáveis ao produto avaliado
 TA – Taxa de Adoção

$$TA = \frac{P}{S} \times 100\%$$

5.3. Estudo de caso

O estudo de caso teve por objetivo investigar a eficiência do uso do ambiente FastInterface enquanto ferramenta de apoio ao desenvolvimento de interface com usuário baseado em modelos e múltipla prototipagem, refletindo os processos e formalismos propostos em MEDITE. A avaliação de FastInterface tem como referência o ambiente TERESA, que tem o mesmo propósito de FastInterface no sentido de oferecer suporte ferramental ao desenvolvimento baseado em modelos. A escolha de TERESA se deu pela sua facilidade de *download* e instalação, tal como, sua grande utilização no meio acadêmico e até mesmo industrial. TERESA segue a abordagem baseada em modelos tradicional, ou seja, prevê as transformações entre os modelos e ao final do processo disponibiliza um protótipo de alta fidelidade, que pode ser visualizado, editado e simulado.

Ambos os ambientes, FastInterface e TERESA, tem como propósito: (i) dar suporte à análise e modelagem da tarefa; (ii) realizar geração do modelo de interface abstrata do usuário (apresentação + diálogo) a partir do modelo da tarefa; (iii) visualizar e editar o modelo de interface abstrata do usuário; (iv) realizar geração do modelo de interface concreta do usuário a partir do modelo da interface abstrata; (v) visualizar e editar o modelo de interface concreta do usuário; e, (vi) realizar geração do código da interface final.

Desta forma, a realização deste estudo de caso pretende evidenciar o resultado positivo ou negativo das seguintes hipóteses:

- **Hipótese 1:** O ambiente FastInterface é mais fácil de aprender e usar que TERESA;
- **Hipótese 2:** O ambiente FastInterface oferece menor resistência ao uso e edição de modelos nas diferentes fases do processo de desenvolvimento do que o ambiente TERESA;

5.3.1. Preparação do estudo de caso

A preparação do estudo de caso foi realizada em cinco etapas que serão descritas nas subseções a seguir.

5.3.1.1. Primeira Etapa – Levantamento de Requisitos para Sistema Consultor

A primeira etapa para a preparação da avaliação, utilizando a técnica de observação do uso das ambientes, consistiu na identificação de requisitos funcionais para um software (hipotético) cujo protótipo e código da interface com usuário deveria ser desenvolvido durante a realização do estudo de caso. O sistema adotado para compor o estudo de caso foi uma aplicação que visa auxiliar a tomada de decisão sobre adoção de Tecnologias da Informação e Comunicação (TIC) nas Empresas, sistema *Consultor*. Suas principais funcionalidades são apresentadas no Quadro 14.

Quadro 14 – Requisitos funcionais do sistema Consultor

Requisitos	Descrição
Efetuar <i>Login</i> no Sistema	Para que o usuário possa acessar o sistema, é necessário primeiramente que ele efetue <i>login</i> no aplicação. Ao efetuar <i>login</i> , o usuário deve informar <i>login</i> , senha e confirmar a ação.
Cadastrar Empresas	Caso o usuário não possua <i>login</i> , deve cadastrar-se. O cadastro da empresa consiste em informar os dados da empresa, a respeito de: CNPJ, Razão Social, Logradouro, Complemento, Bairro, Cidade, Estado, <i>E-mail</i> , URL e <i>Login</i> e Senha desejados.
Realizar Nova Avaliação de TIC ou Continuar uma Avaliação Existente	O usuário logado pode iniciar uma nova avaliação ou continuar uma já existente. O cadastro de uma avaliação consiste na informação do nome da TIC a ser avaliada seguida de uma confirmação. No caso de continuação, o usuário escolhe uma avaliação que deve estar disponível em uma lista de avaliações iniciadas.

Responder Questionários sobre Fatos e Parâmetros relacionados a TIC, Empresa, Mercado/Governo e Custos	<p>Escolhida a avaliação da TIC, o usuário responderá questionários sobre Fatos (valores fixos) e Parâmetros (valores variados).</p> <p>Questionários sobre fatos estão divididos em: Fatos sobre a Empresa, sobre a TIC, sobre o Mercado/Governo e o impacto da TIC na Empresa.</p> <p>Questionários sobre parâmetros estão divididos em: Parâmetros sobre a empresa, a TIC e o Mercado/Governo.</p>
Gerar Relatórios com resultados sobre o Risco, Estratégia, ROI a respeito da TIC avaliada	<p>Após ter respondido todos os questionários, o usuário fica apto a gerar relatórios sobre a tecnologia.</p> <p>Os relatórios disponíveis são: Relatório sobre a Estratégia, Relatório sobre o Risco, Relatório sobre o Custo e Relatório sobre Retorno do Investimento (ROI).</p>

5.3.1.2. Segunda Etapa – Análise e Modelagem da Tarefa para Sistema Consultor

Tendo em vista o processo definido em *Cameleon*, a fase inicial é *Tarefas e Conceitos*. Neste caso, a segunda etapa preparatória para o estudo de caso consistiu na análise e modelagem da tarefa para o sistema *Consultor*. A modelagem da tarefa foi realizada seguindo os formalismos TAOS e CTT, com o auxílio das ferramentas computacionais *FastInterface* e *TERESA*, resultando em dois arquivos XML's, contendo a descrição do modelo da tarefa segundo os dois formalismos.

5.3.1.3. Terceira Etapa – Preparação de Material de Nivelamento

A terceira etapa consistiu na elaboração de: (i) material didático para nivelar o conhecimento dos usuários de teste sobre conceitos relacionados à modelagem da tarefa (formalismos TAOS e CTT), sobre transformações entre modelos e sobre as definições de protótipos de baixa, média e alta fidelidade (APÊNDICE C); (ii) uma apresentação sobre o produto a ser desenvolvido, ou seja, requisitos funcionais do sistema *Consultor* (APÊNDICE D); e, (iii) um tutorial de uso para cada um dos ambientes, *TERESA* e *FastInterface* (APÊNDICE E).

5.3.1.4. Quarta Etapa – Planejamento da Avaliação

A quarta etapa consistiu no planejamento da avaliação, ou seja, na definição: (i) das metas de teste; (ii) dos objetivos gerais e específicos; (iii) dos indicadores quantitativos e qualitativos; (iv) da natureza da avaliação; (v) da dimensão do universo amostral; (vi) da categoria dos usuários de teste; (vii) do número de avaliadores; e, (viii) da duração de cada sessão de teste. O Quadro 15 apresenta o plano de avaliação para o estudo de caso.

Quadro 15 - Plano de avaliação

Planejamento de teste – Observação Direta	
Meta	Investigar a navegabilidade, a clareza da estruturação das funcionalidades, a adequação da terminologia e simbologia utilizada.
Objetivos Gerais	Avaliação de aspectos do processo interativo usuário-produto
Objetivos Específicos	<ul style="list-style-type: none"> ▪ Observação da facilidade de uso da aplicação ▪ Observação da facilidade de execução das tarefas ▪ Mensuração do tempo de conclusão das tarefas ▪ Mensuração do número de consultas à ajuda
Indicadores Quantitativos	<ul style="list-style-type: none"> ▪ Tempo da sessão de avaliação ▪ <i>Status</i> da tarefa (concluída, não concluída) ▪ Número de consultas à ajuda (ou tutorial) ▪ Número de atividades refutadas (concluídas apenas parcialmente)
Indicadores Qualitativos	<ul style="list-style-type: none"> ▪ Facilidade de uso ▪ Facilidade de aprendizado ▪ Facilidade de conclusão da tarefa ▪ Satisfação com resultado da tarefa
Natureza da Avaliação	Somativa objetiva/ subjetiva e qualitativa/ quantitativa
Dimensão do Universo Amostral	12 (doze)
Categoria de Usuários de Teste	Analistas de Tecnologia da Informação da Unidade de Desenvolvimento da DATAPREV Paraíba
Número de Avaliadores	2 (dois)
Duração da Sessão de Teste	90 minutos

5.3.1.5. Quinta Etapa – Identificação e Recrutamento de Usuários de Teste

A quinta etapa consistiu na identificação e recrutamento dos 12 usuários para as sessões de avaliação. O recrutamento deu-se a partir do envio de um e-mail de solicitação de participação para analistas de tecnologia da informação da Unidade de Desenvolvimento da DATAPREV (Empresa de Tecnologia da Informação da Previdência Social) da Paraíba (APÊNDICE G).

5.3.2. Estratégia de realização do estudo de caso

Os usuários de teste, no total de 12 pessoas, trabalharam em duplas (equipes) escolhidas aleatoriamente dentro de um espaço amostral de 114 analistas de tecnologia da informação da DATAPREV. O objetivo das equipes era desenvolver o código da interface final (em HTML) para o sistema *Consultor*.

As duplas foram organizadas em 3 grupos: (i) Grupo 1 (2 equipe) utilizou apenas o ambiente TERESA; (ii) Grupo 2 (2 equipes) utilizou apenas o ambiente FastInterface; (iii) Grupo 3 (2 equipes) utilizou ambos ambientes. O uso de cada ambiente foi avaliado através de sessões de teste de usabilidade. Assim, como o Grupo 3 utilizou ambos ambientes, participou de 2 sessões de teste de usabilidade.

Para o desenvolvimento da interface final do usuário, as oito sessões de avaliação tiveram como base a descrição dos requisitos funcionais da ferramenta *Consultor*, um arquivo contendo o modelo da tarefa incompleto nos formalismos utilizados por ambos os ambientes, um tutorial de uso do ambiente a ser utilizado, além do material de nivelamento. A sessão de teste foi guiada por um roteiro de teste de usabilidade (APÊNDICE F) o qual fornecia sistematicamente os passos de execução das tarefas relativas à geração da interface final do usuário. Cada roteiro de teste disponibilizado dispunha de um total de oito atividades relacionadas à edição do modelo da tarefa, geração e edição da AUI, geração e edição da CUI e geração da FUI.

O início da sessão se deu com uma breve explicação (por parte do avaliador) acerca do objetivo que se desejava atingir com a realização do estudo de caso e apresentação dos materiais disponíveis na sessão, em especial, o roteiro de teste de usabilidade. Em seguida, os usuários iniciavam o desenvolvimento da interface com usuário para o sistema *Consultor*. A

finalização da sessão de teste, independente do ambiente (FastInterface ou TERESA), se dava quando os usuários já se davam por satisfeitos com o resultado obtido após edição nos protótipos de alta fidelidade e geravam a sua interface final. Em seguida, eram aplicados dois tipos de questionários: (i) questionário de levantamento do perfil do usuário (*USer* – APÊNDICE H); e, (ii) questionário da sondagem da satisfação subjetiva do usuário (*USE* – APÊNDICE I). Todos os questionários foram respondidos individualmente e não possuíam identificação pessoal.

Foram analisados aspectos objetivos e subjetivos acerca do uso dos ambientes no processo de concepção de interface durante a sessão de teste. Os aspectos objetivos considerados foram: (i) tempo da sessão de avaliação (desenvolvimento da interface com usuário); (ii) *status* da tarefa (interface gerada ou não); (iii) número de consultas ao tutorial e ajuda fornecida pelo ambiente em uso; e, (iv) número de atividades refutadas (executadas parcialmente). Os aspectos subjetivos analisados tinham relação com a facilidade de uso de ambos ambientes. Neste sentido, dados foram coletados a partir de comentários informais feitos pelos usuários ao utilizarem os ambientes e pelas respostas obtidas com a aplicação do questionário *USE*.

5.3.3. Análise dos resultados obtidos

A análise dos resultados obtidos com a realização do estudo de caso, através da coleta de dados durante a observação direta da sessão de teste de usabilidade do produto, teve como pressuposto os indicadores quantitativos e qualitativos apresentados previamente no Quadro 15. Esses indicadores se referem diretamente aos aspectos objetivos e subjetivos apresentados na sessão anterior.

5.3.3.1. Análise dos indicadores objetivos

Os dados coletados durante a realização do estudo de caso para cada uma das equipes descritas levaram em consideração os indicadores quantitativos ora citados e estão apresentados na Tabela 2.

Tabela 2– Mensuração do desempenho a partir dos indicadores quantitativos

Grupo	Equipe	Tempo (min)	Status	Consultas à Ajuda/Tutorial	No. de Atividades Refutadas
FastInterface	FastInterface_1	52	C	0	0
	FastInterface_2	39	C	3	0
	FastInterface_Teresa_1	63	C	1	1
	FastInterface_Teresa_2	62	C	0	0
TERESA	Teresa_1	80	C	5	4
	Teresa_2	49	C	1	2
	FastInterface_Teresa_1	51	C	5	3
	FastInterface_Teresa_2	42	C	2	3

Legenda: C – Concluída; N/C – Não Concluída

Fazendo um comparativo entre os dados relacionados aos indicadores quantitativos coletados durante a execução do estudo de caso com o uso dos ambientes FastInterface e TERESA, percebe-se que:

- i. O tempo médio para conclusão da sessão de avaliação dos ambientes FastInterface (54 min) e TERESA (55,4 min) é bastante similar;
- ii. De um total de 8 atividades a serem realizadas de acordo com o roteiro de teste disponibilizado, todas (4/4) as equipes de TERESA refutaram algum tipo de atividade, enquanto que apenas (1/4) equipe refutou em FastInterface;
- iii. Entre as equipes que refutaram algum tipo de atividade, em TERESA (3/4) equipes refutaram pelo menos (3/8) atividades (índice bastante elevado) e (1/3) equipe refutou 1 atividade, em FastInterface (1/4) equipe refutou (1/8) atividade;
- iv. Para o uso de TERESA, todas (4/4) as equipes necessitaram consultar a ajuda/tutorial de uso, enquanto que para o uso de FastInterface (2/4) equipes fizeram uso desse tipo de recurso;
- v. A equipe que mais utilizou ajuda/tutorial (3 consultas) no FastInterface, foi a que registrou menor tempo para conclusão da sessão e não refutou nenhuma atividade, enquanto que as equipes que mais utilizaram a ajuda/tutorial (5 consultas), no

TERESA, foram as que mais demoram para concluir suas sessões e apresentaram os maiores índices de atividades refutadas (4/8 e 3/8, respectivamente).

Ao se detalhar as atividades realizadas na execução do estudo de caso utilizando os ambientes FastInterface e TERESA, obtêm-se os resultados apresentados na Quadro 16.

Quadro 16 – Resultado da Execução das Atividades Previstas no Roteiro de Teste

Grupo	Equipe	Editar MT	Gerar AUI	Editar AUI	Consistência AUI-MT	Editar CUI	Gerar FUI
FastInterface	FastInterface_1	C	C	C	C	C	C
	FastInterface_2	C	C	C	C	C	C
	FastInterface_Teresa_1	C	C	C	N/C	C	C
	FastInterface_Teresa_2	C	C	C	C	C	C
TERESA	Teresa_1	N/C	N/C	N/C	N/C	C	C
	Teresa_2	C	C	C	N/C	N/C	C
	FastInterface_Teresa_1	N/C	C	N/C	N/C	C	C
	FastInterface_Teresa_2	C	C	N/C	N/C	N/C	C

Legenda: C – Concluída; N/C – Não Concluída

Analisando os dados coletados, observa-se que: (i) embora FastInterface e TERESA tenham apresentado um tempo médio de finalização da sessão de avaliação bastante similar, o número de tarefas refutadas em TERESA é bastante elevado; (ii) do total de equipes que realizaram o estudo de caso para FastInterface apenas uma equipe deixou de concluir todas as atividades, enquanto que com o uso de TERESA todas as equipes deixaram de concluir pelo menos duas atividades; (iii) o número de consultas à ajuda/tutorial durante o uso de TERESA (treze registros) diverge consideravelmente com o número obtido com o uso de FastInterface (quatro registros).

Os indicadores quantitativos relacionados ao número de consultas à ajuda/tutorial que foram coletados mostram evidências para a confirmação da **Hipótese 1**. O grande número de ocorrências (treze) relativos à ajuda unido a não conclusão de atividades ao se utilizar o

ambiente TERESA indicam a dificuldade do uso da ferramenta frente ao FastInterface. Esta análise, ainda que prematura, aponta que o ambiente FastInterface é mais fácil de usar do que o ambiente TERESA.

Observando-se ainda o número de atividades refutadas pelos usuários, percebe-se que: (i) das equipes que utilizaram TERESA, (2/4) equipes não conseguiram ou simplesmente desistiram editar o modelo da tarefa, enquanto que nenhuma (0/4) equipe que utilizou FastInterface deixou de editar esse modelo; (ii) (3/4) equipes que utilizaram TERESA não conseguiram ou desistiram de editar a AUI, enquanto que em FastInterface todas as equipes editaram esse modelo; (iii) (2/4) equipes que utilizaram TERESA, não conseguiram ou desistiram de editar o modelo no nível CUI, enquanto que com FastInterface nenhuma (0/4) deixou de realizar essa atividade. A diferença significativa em termos de número de atividades de edição de modelos refutadas em FastInterface (0/12), face TERESA (7/12) reforça a **Hipótese 2**. Esse resultado indica que é verdadeira a afirmação que o ambiente FastInterface oferece menor resistência ao uso e edição de modelos nas diferentes fases do processo de desenvolvimento do que o ambiente TERESA.

5.3.3.2. Análise dos indicadores subjetivos

Os dados relacionados aos indicadores qualitativos foram coletados através de comentários feitos pelas equipes durante as sessões de avaliação. Tais comentários revelam algumas das principais dificuldades encontradas pelos usuários ao utilizarem os ambientes TERESA e FastInterface e refletem de alguma forma o que foi apresentado com relação aos indicadores objetivos.

Dentre as equipes que realizaram as atividades do roteiro com TERESA (2/4) apresentaram bastante dificuldade em realizar a edição no modelo da tarefa e refutaram essa atividade. Os fatores mais evidentes para esse fato estão relacionados com o preenchimento de atributos de descrição da tarefa, cuja falta impede a geração da AUI. Não estava claro para os usuários quais informações deveriam ser obrigatórias. O *Help* da ferramenta não disponibilizava esse tipo de informação. Da mesma forma, o sistema apresentava falhas ao se utilizar a barra de rolagens. Nesse momento algumas tarefas sumiam e só apareciam novamente ao se minimizar e maximizar a ferramenta. Alguns comentários da equipe Teresa_1 reforçam essa afirmação, como podem ser vistos abaixo:

”O sistema está bugado, as tarefas somem. Que negócio doido.(Quando do uso das barras de rolagem)”;

”A tarefa não tem relacionamento temporal”;

“Claro, a gente não consegue colocar“;

Após um tempo estimado de vinte minutos essa equipe desistiu da edição do modelo da tarefa e passou para outra atividade. Da mesma forma, os comentários da equipe FastInterface_Teresa_1 evidenciam as dificuldades na edição do modelo da tarefa:

”Porque eu tenho que informar novamente a plataforma? (Referindo-se à necessidade de informar a plataforma em dois lugares distintos do descritor da tarefa. A não inclusão da informação impedia a geração da AUI)”;

”Fecha tudo e abre novamente, tá cheio de bugs”;

“Eu não acredito não, a ferramenta apagou as tarefas! (Fato ocorrido ao fazer acionamento do Help do ambiente)“;

“O usuário fica revoltado fazendo isso“.

Após a ferramenta ter corrompido o arquivo com o modelo da tarefa uma vez, e depois de várias tentativas frustradas de edição, a equipe desistiu da conclusão da atividade de edição do modelo.

Um outro fator de grande insatisfação com o uso do TERESA diz respeito à lentidão da ferramenta quando da geração do modelo AUI. As (4/4) equipes demonstraram insatisfação quanto ao desempenho do ambiente, como pode ser observado em alguns comentários:

”A gente não pode esperar o dia inteiro não, que coisa lenta (Equipe Teresa_1)”;

”O desempenho da ferramenta é horrível (Equipe FastInterface_Teresa_2)”;

Dentre estas equipes, a equipe Teresa_1 não conseguiu concluir a geração e refutou essa atividade. Desta forma, foi disponibilizado um arquivo XML com o modelo AUI previamente gerado.

Da mesma forma, com relação à edição da AUI, a ferramenta apresentou-se bastante lenta na abertura do arquivo XML com informações do modelo em questão. O comentário abaixo resume a afirmação:

“A ferramenta parece ter travado, não dá indicação de nada (Equipe FastInterface_Teresa_1)”.

Ao ter o modelo AUI carregado em seu editor, apenas (1/4) equipe, a Teresa_2, apresentou maior interesse em editar os elementos do modelo gerado. Porém, depois de criarem uma nova tela para a tarefa de *login*, a ferramenta não dava opção de copiar ou arrastar o agrupamento previamente gerado. Assim, desistiram de fazer a edição que tinha imaginado e deixaram a tela de lado. Eis o comentário feito com relação a esta operação:

“Esse agrupamento não era pra estar nesta tela. (Com relação à tela de login)”

“Nã, essa ferramenta é muito complicada. Muito estranha!(Ao tentar editar nova tela no modelo AUI)”.

Por fim, (2/4) equipes que utilizaram o TERESA não conseguiram concluir a atividade de edição do modelo CUI. Esse fato ocorreu devido à falta de confirmação das operações de edição, a qual não conseguiu ser detectada pelos usuários. Eis alguns comentários acerca da edição da CUI:

“Não consigo ver o que estou editando (Pelo fato de se editar o estilo em cima de descritores)”;

“Oxe! É preciso dar Apply (Quando se descobriu que a edição só poderia ser visualizada quando o botão Apply tivesse sido acionado)”;

“A edição não serve pra o grupo todo (Ao tentar aplicar a edição para todas as telas)”.

Para os usuários de FastInterface, as maiores insatisfações se referem aos aspectos estéticos de edição do ambiente, pois a grande maioria não apresentou problemas com relação à geração e transformação entre os modelos. Tomando com base a edição do modelo da tarefa, todas as equipes conseguiram concluir a atividade. Apenas uma equipe relatou a dificuldade de edição devido ao tamanho reduzido da figura que representa visualmente a tarefa, fazendo com que seu nome não seja apresentado completamente.

Com relação a edição da AUI, que no caso de FastInterface é visualizado sob a forma de um protótipo de média fidelidade, uma equipe apresentou dificuldades para remover uma figura com o botão *delete*. Ao acionar, a foco da aplicação estava em cima do projeto corrente fazendo com o usuário apagasse o projeto todo. Mesmo com a mensagem alertando sobre a remoção do projeto, o usuário acabou executando a tarefa.

A edição da CUI, que no caso de FastInterface é vista e editada sob a forma de um protótipo de alta fidelidade, gerou algumas sugestões de melhoria que foram feitas ao longo das sessões de avaliação. Dentre elas, podemos destacar:

“Senti falta de um ajuste fino nesse protótipo de alta fidelidade(Com relação à falta de opção de edição de localização dos objetos em Hi-Fy)“;

“Era para ter um preview (Quando se estava editando o estilo na caixa de edição)“;

“Esse agrupamento de opções de edição não facilita a compreensão, deveria ser mais evidente“;

“Deveria ter um botão para ‘resetar’ os estilos individuais“;

“Essa distância entre os objetos deveria ser melhor equalizada (Quando da geração da localização dos protótipos de alta fidelidade baseada nos de média)“.

Para os itens levantados como causa de insatisfação em FastInterface, estes serão avaliados e incorporados em uma próxima versão da ferramenta.

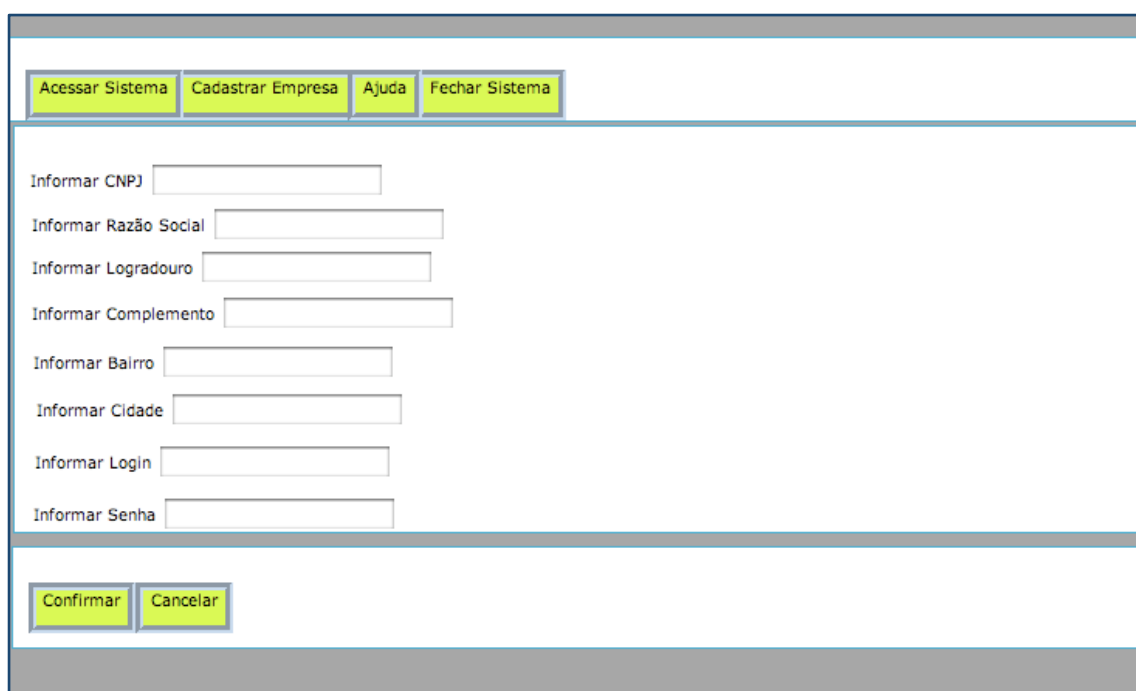
A Figura 32 apresenta a interface final gerada pelo ambiente TERESA para as tarefas Cadastrar Empresa e Efetuar *Login*, já as Figuras 33 e 34 apresentam as interfaces finais geradas para as mesmas tarefas utilizando o ambiente FastInterface, após a sessão de avaliação. Percebe-se que as regras de transformação entre os modelos inerentes da ferramenta TERESA agrupam as duas tarefas ora citadas em uma única tela.

Title

Informar_Login1	<input type="text"/>
Informar_Senha1	<input type="text"/>
<input type="button" value="Ok"/>	
Informar_CNPJ1	<input type="text"/>
Informar_Razao_Social1	<input type="text"/>
Informar_Logradouro1	<input type="text"/>
Informar_Complemento1	<input type="text"/>
Informar_Bairro1	<input type="text"/>
Informar_Cidade1	<input type="text"/>
Informar_Estado1	<input type="text"/>
Informar_Login_Desejado1	<input type="text"/>
Informar_Senha_Desejada1	<input type="text"/>
<input type="button" value="Ok"/>	

[Ajuda1](#)

Figura 32 – Interface Final Gerada pelo Ambiente TERESA para Tarefa Cadastrar Empresa e Efetuar *Login*



Acessar Sistema Cadastrar Empresa Ajuda Fechar Sistema

Informar CNPJ

Informar Razão Social

Informar Logradouro

Informar Complemento

Informar Bairro

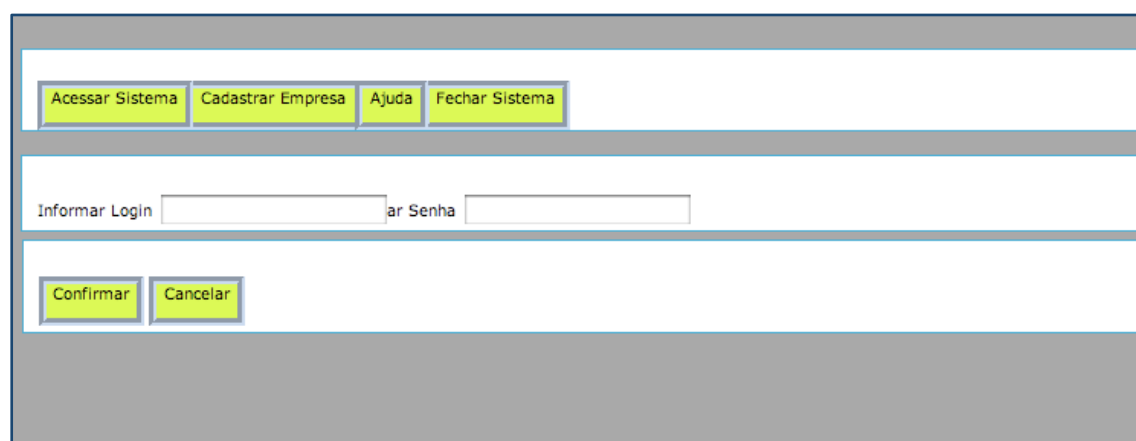
Informar Cidade

Informar Login

Informar Senha

Confirmar Cancelar

Figura 33 – Interface Final Gerada pelo Ambiente FastInterface para a Tarefa Cadastrar Empresa



Acessar Sistema Cadastrar Empresa Ajuda Fechar Sistema

Informar Login ar Senha

Confirmar Cancelar

Figura 34 - Interface Final Gerada pelo Ambiente FastInterface para a Tarefa Efetuar *Login*

5.4. Sondagem da satisfação subjetiva do usuário

Nesta seção serão apresentados os resultados relativos à análise dos questionários *USer* e *USE*, os quais permitem delinear o perfil dos usuários participantes da sessão de avaliação do estudo de caso, bem como sondar o grau de satisfação de cada um deles ao utilizarem os ambientes FastInterface e TERESA.

5.4.1. Análise do questionário USer– Perfil do Usuário

O estudo de caso envolveu a participação de 12 voluntários que exercem o cargo de analista de tecnologia da informação na DATAPREV. Os participantes têm entre 23 e 35 anos de idade (Média = 29 anos), incluindo 3 pessoas do sexo feminino e 9 pessoas do sexo masculino. Os participantes foram identificados e recrutados de acordo com as atividades que estavam desempenhando nos projetos em andamento na DATAPREV.

Todos eles trabalham diretamente com o desenvolvimento de sistemas, e em sua grande maioria (10/12) com a produção de protótipos. Porém, nenhum (0/12) usuário afirmou ter conhecimento prévio acerca da modelagem da tarefa. A Tabela 3 sumariza as características dos usuários.

Tabela 3 – Dados sumarizados a respeito das características dos usuários de teste

No	Idade	Sexo		Grau de Instrução			Experiência	
		F	M	Graduado	Mestrando	Mestre	Prototipagem	Modelagem da Tarefa
12	29	3	9	8	2	2	10	0

5.4.2. Análise do questionário USE

O questionário *USE* foi composto por um total de 30 questões para os usuários do ambiente FastInterface e 28 para o TERESA. Esses dois itens a menos no questionário de TERESA se relacionam com a questão da representação visual no nível AUI, a qual o ambiente não apresenta.

As questões abordam aspectos sobre: (i) usabilidade (facilidade de aprendizado e uso); (ii) funcionalidades do sistema; (iii) situações de erro e travamento; (iv) ajuda; (v) satisfação de uso; e, por fim (vi) contexto de uso do ambiente.

Os dados coletados através das questões relacionadas à facilidade de uso dos ambientes estão sumarizados na Tabela 4.

Tabela 4–Dados sumarizados com relação à facilidade de uso dos ambientes FastInterface e TERESA

Questão	FastInterface					TERESA				
	MF	F	NF/ND	D	MD	MF	F	NF/ND	D	MD
O uso do ambiente de desenvolvimento na construção de protótipos		8						2	5	1
O uso da ferramenta na edição de protótipos é	1	6	1					1	6	1
A transformação e edição entre os modelos em diferentes etapas do projeto utilizando o ambiente é	5	3					1		7	
A visualização e compreensão dos diferentes modelos envolvidos no processo de concepção é	1	5	2					1	6	1
O uso das funcionalidades mais comuns do sistema é	2	6						6	1	1
A navegação através das diferentes opções de menu, janelas de diálogo e barra de ícones do ambiente é	3	4	1				1	4	3	
A compreensão da estruturação dos menus e barras de ícones é	2	5	1				3	3	2	
A recuperação de situações de erro ao utilizar o ambiente é		2	4	1	1			3	4	1
A recuperação de situações de travamento ao utilizar o ambiente é		1	5					1	6	1
A compreensão das mensagens de erro apresentadas pelo ambiente é		3	4			1	2	1	2	2
A compreensão das informações de interesse existentes na ajuda da ferramenta é		3	3					4	2	1

Legenda: MF – Muito Fácil; F – Fácil; NF/ND – Nem Fácil Nem Difícil; D – Difícil; MD – Muito Difícil

Percebe-se, observando a Tabela 4, que, no tocante à **funcionalidade** (primeira parte da tabela), FastInterface apresentou resultados bastante satisfatórios em face dos resultados apresentados pelo TERESA. Dentre os principais resultados, pode-se destacar que: (i) (8/8) usuários classificaram que a construção de protótipos com FastInterface é fácil, enquanto que com TERESA (6/8) indicaram algum tipo de dificuldade; (ii) com o uso de FastInterface, (7/8) usuários classificaram a edição de protótipos fácil ou muito fácil, enquanto que (7/8) usuários de TERESA indicaram algum tipo de dificuldade; (iii) (6/8) usuários de FastInterface indicaram facilidade com relação à visualização e compreensão dos diferentes modelos envolvidos no processo de concepção, enquanto que (7/8) usuários de TERESA apresentaram dificuldades; e (iv) (8/8) usuários de FastInterface classificaram que o acesso às principais

funcionalidades do sistema era fácil ou muito fácil, enquanto que (2/8) usuários de TERESA apresentaram algum tipo de dificuldade e (6/8) não apresentaram nem facilidade nem dificuldade.

Com relação à **usabilidade**, (7/8) usuários indicaram que o ambiente FastInterface é fácil ou muito fácil a respeito da estruturação de aspectos funcionais, enquanto que com TERESA, em sua maioria, os usuários julgaram nem fácil nem difícil.

Os aspectos pertinentes às **situações de erro e travamento** demonstraram que a grande maioria dos usuários de TERESA indicaram algum tipo de dificuldade, enquanto que a maioria dos usuários de FastInterface classificaram esses aspectos como nem fáceis nem difíceis. Porém, esse indicador relacionado a FastInterface parece ter correspondência ao fato dos usuários do ambiente não terem tido nenhuma ocorrência de travamento durante as sessões de teste. Quanto ao mecanismo de **ajuda**, (3/7) usuários de TERESA demonstraram dificuldades em seu uso e (4/7) julgaram nem fácil nem difícil, enquanto que dos usuários de FastInterface (3/3) indicaram facilidade de uso e (3/3) julgaram nem fácil nem difícil, o restante dos usuários não opinou.

Os dados levantados através de questões sobre a concordância de afirmações relacionadas ao uso dos ambientes FastInterface e TERESA estão agrupados no tocante ao contexto de uso da ferramenta, à satisfação, à usabilidade e à ajuda, respectivamente. Esses dados estão sumarizados na **Tabela 5**.

Tabela 5 – Dados sumarizados com relação à concordância de afirmações sobre o uso dos ambientes FastInterface e TERESA

Questão	FastInterface					TERESA				
	CT	C	NC/ND	D	DT	CT	C	NC/ND	D	DT
O uso de modelo da tarefa auxilia na compreensão e correteude dos requisitos do usuário	5	2	1			2	6			
Acredito que o modelo da tarefa seja um ótimo ponto de partida para a construção de interfaces do usuário	4	3	1			2	5	1		
Acredito que a abordagem baseada em modelos favorece na redução de esforço na construção de interfaces do usuário	5	2	1			1	5	2		
O uso de representação visual numa etapa intermediária do processo auxilia na redução da disparidade entre o protótipo final e	3	5				-	-	-	-	-

O uso de representação visual numa etapa intermediária do processo auxilia na redução da disparidade entre o protótipo final e a interface com usuário que se deseja	3	5				-	-	-	-	-
O uso de diferentes técnicas de prototipagem em diferentes estágios do processo de concepção de interfaces do usuário auxilia o desenvolvimento	2	3	2		1	-	-	-	-	-
As respostas da ferramenta às minhas ações são muito lentas			2	5	1	2	5	1		
Acho que a ferramenta atende plenamente às minhas necessidades		3	4	1				3	4	1
Sinto-me receoso em alguns momentos ao usar a ferramenta		1		4	3	1	5	2		
Em alguns momentos sinto-me frustrado com o modo como a ferramenta executa tarefas de meu interesse		2		5	1	2	5	1		
Sinto-me satisfeito com o resultado que obtive ao final da sessão	2	6					1	2	2	3
Acredito que diante do protótipo de alta fidelidade exibido pela ferramenta, ao final da sessão, o usuário pode contribuir com sugestões de melhoria para o sistema Consultor	3	5					6	1	1	
Acredito que um usuário do sistema <i>Consultor</i> é capaz de interagir com a ferramenta sem problemas	2	3	2	1			1	2	5	
De um modo geral, sinto-me satisfeito ao usar a ferramenta	4	3	1				1	1	5	1
Passarei a usar a ferramenta quando necessitar projetar interfaces do usuário		5	3					1	5	2
Consigo executar as tarefas de modo direto ao usar a ferramenta		3	3	2				3	4	1
Perco muito tempo tentando aprender a usar a ferramenta			1	5	2	2	6			
Consigo fazer exatamente o que desejo com os recursos oferecidos pela ferramenta		3	3	2			1	2	4	1
A ferramenta apresenta tantas funcionalidades que as vezes acho que nunca aprenderei ou terei				2	6	1	3	3		1

Legenda: CT – Concordo Totalmente; C – Concordo; NC/ND – Nem Concordo Nem Discordo; D – Discordo; DT – Discordo Totalmente

Dentre os dados apresentados na **Tabela 5**, os que merecem mais destaque são:

- i. Oito (8/8) usuários de FastInterface admitiram (parcial e totalmente) que o uso de uma **representação visual numa etapa intermediária do processo** auxilia na redução da disparidade entre o protótipo final e a interface com usuário que se deseja;
- ii. Cinco (5/8) usuários de FastInterface concordaram (parcial e totalmente) que o **uso de diferentes técnicas de prototipagem em diferentes estágios do processo** de concepção de interfaces do usuário auxilia o desenvolvimento;
- iii. Sete (7/8) usuários de TERESA indicaram **frustração** quanto ao uso do ambiente, enquanto que seis (6/8) não expressaram este tipo de sentimento com relação ao uso do ambiente FastInterface;
- iv. Sete (7/8) usuários de FastInterface concordaram (parcial e totalmente) que, de um modo geral, sentiram-se **satisfeitos** ao utilizarem o ambiente, enquanto que seis (6/8) não apresentaram o mesmo tipo de sentimento com relação a TERESA;
- v. Sete (7/8) usuários de TERESA não revelaram interesse **em utilizar o ambiente para o desenvolvimento de interfaces**, enquanto que cinco (5/8) usuários de FastInterface revelaram interesse.

5.5. Conclusões

O uso da abordagem multidimensional na avaliação do ambiente FastInterface foi bastante positivo uma vez que puderam ser abordados diversos aspectos sob diferentes perspectivas, dando, assim, uma maior abrangência ao processo avaliatório.

A avaliação de FastInterface resultou no levantamento de falhas obtidas através da (i) inspeção de conformidade às Partes 14, 16 e 17 do Padrão ISO 9241; (ii) análise dos indicadores quantitativos e qualitativos coletados durante a execução de um estudo de caso; e, (iii) sondagem da satisfação subjetiva do usuário.

A **inspeção de conformidade** realizada para FastInterface resultou em um total de **22 falhas**, entre as quais 11 falhas referem-se à Parte 14 (Diálogo via Menus), 05 falhas à Parte 16 (Diálogo via Manipulação Direta) e 06 falhas à Parte 17 (Diálogo via Formulário). Mesmo com o número de falhas, as **taxas de adoção** são consideradas extremamente satisfatórias:

81,03% para Diálogo via Menu; **88,88%** para Diálogo via Manipulação Direta; e, **81,25%** para Diálogo via Formulário.

A **análise dos indicadores quantitativos e qualitativos**, bem como os resultados provenientes da **sondagem da satisfação subjetiva do usuário (USE)** confirmam as **Hipóteses 1 e 2** levantadas para o estudo de caso.

Capítulo 6

Conclusão e trabalhos futuros

Este capítulo apresenta as conclusões e considerações finais da dissertação, da mesma forma, indica proposições para trabalhos futuros que poderão contribuir significativamente para complementar o trabalho ora descrito.

6.1. Contextualização geral da dissertação

O objetivo geral deste trabalho foi favorecer a adoção da abordagem de desenvolvimento baseada em modelos fornecendo uma proposta que aliasse múltipla prototipagem evolutiva com os diferentes níveis de abstração propostos no *framework* de referência *Cameleon*.

A princípio, fez-se necessário identificar o estado da arte das técnicas de concepção de interface baseadas em modelos e abordagens que utilizassem protótipos de diferentes níveis de fidelidade de modo a obter conhecimento acerca das melhores práticas de projeto. Sendo assim, foi feita uma revisão bibliográfica e levantamento e uso das principais ferramentas de suporte ao projeto baseado em modelos.

A fim de avaliar os ambientes identificados no estado da arte, foi estudado e analisado o *framework* referência unificado para desenvolvimento de interfaces multi-plataforma, *Cameleon*, visando tê-lo como referência conceitual para identificar as lacunas nas técnicas de concepção elencadas. Desta maneira, detectou-se que antes de qualquer modelagem, os projetistas costumam fazer um protótipo de baixa fidelidade a fim de entender melhor o domínio do problema da aplicação que se deseja conceber. A partir de então, observou-se as

técnicas e formalismos utilizados no nível de abstração *Tarefas e Conceitos*, *Interface Abstrata do Usuário*, *Interface Concreta do Usuário* e *Interface Final do Usuário*, traçando um panorama sobre o que existe atualmente em termos de desenvolvimento baseado em modelos.

A partir das lacunas identificadas no levantamento do estado da arte, foi feita uma proposta de *framework* que preenche algumas delas, o *framework* FastInterface. Esse *framework* visa mapear os processos previstos em *Cameleon* com as diferentes técnicas de prototipagem existente a fim de oferecer representações visuais ao projetista e ao usuário ao longo de todo processo de concepção. Desta forma, apresentou-se que, por suas características, protótipos de média fidelidade se adequam ao processo de *Interface Abstrata do Usuário*, enquanto que protótipos de alta fidelidade se adequam ao processo de *Interface Concreta do Usuário*. Por fim, a *Interface Final do Usuário*, deve refletir, em alguma linguagem compilada ou interpretada, o que foi previsto em termos de protótipo de alta fidelidade.

Por seus atributos, a metodologia MEDITE foi estudada e adequada ao *framework* FastInterface de modo a avaliar os ganhos de se ter múltiplos protótipos dentro de um processo evolutivo com diferentes níveis de abstração. Desta forma, foi necessária a elaboração de uma técnica de geração automática de protótipos de alta fidelidade baseado nos protótipos de média fidelidade. Para este caso, foi disponibilizando uma aparato ferramental, o sistema Hi-Fy, que além de dar suporte a esse processo, tem como objetivo gerar a interface final do usuário a partir do protótipo de alta fidelidade produzido.

Tendo MEDITE como referência, era necessário integrar seus processos através de um ambiente de concepção de interface que englobasse todas as etapas do desenvolvimento refletindo o *framework* proposto. Desta maneira, projetou-se e implementou-se um MB-IDE, denominado FastInterface, que tinha como um de seus objetivos contemplar todas as ferramentas de suporte a MEDITE como seus módulos.

A partir da obtenção do MB-IDE FastInterface, foi possível fazer uma validação da abordagem e do ambiente de desenvolvimento de interfaces por meio de um processo de avaliação multidimensional que incluiu: (i) inspeção de conformidade ao Padrão ISO 9241; (ii) estudo de caso comparativo com testes de usabilidade realizados por usuários, frente a outra abordagem (desenvolvimento baseado em modelos tradicional); por fim, (iii) sondagem da satisfação subjetiva do usuário.

Uma página de acompanhamento do projeto FastInterface está disponível com informações sobre os modelos e técnicas utilizados em MEDITE, suas ferramentas de suporte, as melhorias implementadas e o *link* de *download* tanto do ambiente como de seus módulos. O acesso à ferramenta é livre e inclui o código fonte (<http://www.dsc.ufcg.edu.br/~figroup>).

6.2. Conclusão

De acordo com as idéias iniciais que nortearam a elaboração e execução deste trabalho é possível afirmar que os resultados obtidos com a finalização do mesmo foram satisfatórios, dado que:

- (i) de acordo com as informações coletadas a partir da aplicação dos questionários de satisfação subjetiva dos usuários que participaram do estudo de caso (para ambas os ambientes – FastInterface e TERESA), o uso de uma abordagem baseada em modelos reduz o tempo e custo no desenvolvimento de interface com usuário;
- (ii) de forma semelhante, o uso de informações baseadas nos modelos da tarefa favoreceu ao entendimento e levantamento dos requisitos do usuário e se revelou satisfatório pela maioria dos usuários recrutados para o estudo de caso;
- (iii) o uso da abordagem apresentada pelo *framework FastInterface* favoreceu a redução da resistência de uso de abordagens baseadas em modelos uma vez que propiciou a representação visual da interface durante os diferentes estágios do processo de concepção, favorecendo, assim, o entendimento acerca dos modelos que se está utilizando;
- (iv) a facilidade de uso de FastInterface, em comparação ao ambiente TERESA, mostrou-se evidente durante a realização do estudo de caso a partir da coleta dos indicadores quantitativos e qualitativos, com os resultados obtidos com a aplicação do questionário pós-teste; e,
- (v) a partir da visualização da interface gerada em um momento intermediário do processo de concepção, foi observado que a antecipação da avaliação por parte dos projetistas e usuários da ferramenta favoreceu a redução do esforço de

edição, fazendo com que essa seja menos onerosa do que em estágios mais avançados do processo.

No entanto, embora a avaliação geral de FastInterface tenha sido bastante positiva, foram encontradas falhas decorrentes da inspeção de conformidade às Partes 14, 16 e 17 do Padrão ISO 9241. Embora as falhas encontradas não comprometem de forma significativa o processo de interação do usuário com o produto, objetiva-se corrigi-las a fim de que o processo de interação seja otimizado e as taxas de adoção e o índice de satisfação subjetiva do usuário possam ser melhorados.

6.3. Proposições para trabalhos futuros

Considerando o tempo restrito para concretização deste trabalho de dissertação, existem ainda algumas proposições de trabalho futuros, quais sejam:

- (i) eliminar as falhas encontradas em FastInterface com a inspeção de conformidade e com a realização do estudo de caso para melhorar o processo interativo do usuário;
- (ii) ampliar o escopo do estudo de caso (em número de usuário e de indicadores quantitativos e qualitativos) a fim de realizar um processamento estatístico mais rígido em cima dos dados coletados;
- (iii) fazer um teste de usabilidade relacionado à interface gerada ao final do processo de desenvolvimento com o uso do ambiente FastInterface;
- (iv) inserir uma base de regras ergonômicas, padrões de projeto e heurísticas à geração dos protótipos de alta fidelidade;
- (v) inserir o ambiente FastInterface enquanto ferramenta de apoio a um processo de desenvolvimento de software;

Referências

- (Abrahão, Iborra e Vanderdonkt, 2007) Abrahão, S., Iborra, E., Vanderdonkt, J.: Usability Evaluation of User Interfaces Generated with a Model-Driven Architecture Tool. Chapter 2. In: Law, E., Hvannberg, E., Cockton, G. (eds.) *Maturing Usability: Quality in Software, Interaction and Value*. HCI Series, Springer, Berlin (2007)
- (Aguiar, 2007) Aguiar, Y. P. C. *SMILE – Uma Ferramenta para Geração Automática, Edição e Simulação de Protótipos de Interface com usuário*. Dissertação (Mestrado em Informática) – Ciência da Computação, Universidade Federal de Campina Grande, Campina Grande PB, 2007.
- (Balzert *et. al.*, 1996) Balzert, H., Hofmann, F., Kruschinski, V., Niemann, C. The JANUS Application Development Environment - Generating More than the User Interface. *CADUI 1996*: 183-208, 2006.
- (Brown, 1997) Brown J.: Exploring Human-Computer Interaction and Software Engineering Methodologies for the Creation of Interactive Software. *SIGCHI Bulletin* 29,1 (1997) 32–35 .
- (Calvary, Coutaz e Thevenin, 2003) Calvary, G., Coutaz, J., Thevenin, *et. al.*. A Unifying Reference Framework for Multi-Target User Interfaces. In *Interacting with Computer*, p. 289–308, 2003.
- (Card, Moran e Newell, 1983) Card, S.K., Moran, T.P., Newell, A. *The psychology of human-computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates. 1983.

- (Ceri, Fraternali e Bongio, 2000) Ceri, S. Fraternali, P. Bongio, A.: "Web Modeling Language (WebML): a Modeling Language for Designing Web Sites". *WWW9 Conference*, Amsterdam, May 2000.
- (Constantine e Lockwood, 1999) Constantine, L. L., Lockwood, L. A. D. *Software for use*. Addison-Wesley. 1999.
- (Cordeiro, 2003) Pedro B. Cordeiro. *Projeto e implementação do módulo tame da ferramenta itaos para análise e modelagem da tarefa*. Dissertação de Mestrado, Universidade Federal de Campina Grande, Departamento de Ciência da Computação, Campina Grande, 2003.
- (Coyette, 2007) Coyette, A.. *A Methodological Framework for Multi-Fidelity Sketching of User Interfaces*. Thèse de Docteur, Université catholique de Louvain, Autumn, 2007.
- (Coyette, Kieffer e Vanderdonckt, 2007) Coyette, A., Kieffer, S., Vanderdonckt, J.: Multi-Fidelity Prototyping of User Interfaces. In: *Proc. of Interact'2007*, Springer-Verlag (2007) 149–162
- (Coyette, Vanderdonckt e Limbourg, 2006) Coyette, A., Vanderdonckt, J., e Limbourg, Q. SketchiXML: An Informal Design Tool for User Interface Early Prototyping. In *Proc. of RISE'2006 Workshop on Rapid User Interface Prototyping Infrastructures Applied to Control Systems RUIPICAS'2006*, Geneva, 2006.
- (CTTE, 2002) *The ConcurTaskTrees Environment*. Disponível em: <http://giove.cnuce.cnr.it:8443/CTTE/servlet/DownloadCTTE> Acesso em: Julho de 2007.
- (EasyAccept, 2005) *EasyAccept Tool*. Disponível em <http://easyaccept.org>. Acesso em: Junho de 2008.)
- (Ferré, 2001) Ferré, Xavier. Incorporating Usability into an Object Oriented Development Process. *Workshop on Usability throughout the entire systems development lifecycle (INTERACT)*, Tokyo, Japan, 2001.

- (Gamma *et. al.*, 2000) Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. *Padrões de Projeto - Soluções Reutilizáveis de Software Orientado a Objeto*. Porto Alegre: Bookman, 2000. ISBN: 8573076100
- (Garcia *et. al.*, 2004) Garcia, F. P., Lima, A. H. G., Ferreira D. S., Leite Junior, F. L., Rocha, G. R. C., Mendes, G. W. D., Pontes, R. F., Rocha, V. K. H., Dantas, V. F. (2004) “easYProcess: Um Processo de desenvolvimento para Uso no Ambiente Acadêmico”. XII WEI - Workshop de Educação em Computação, *XXIV Congresso da Sociedade Brasileira de Computação*.
- (Gomez, 2004) Gomez, J. *Model-Driven Web Development with VisualWADE*. Web Engineering, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2004.
- (Griffiths *et. al.*, 2001) Griffiths, T; McKirdy, J; Paton, N. W. ; Gray, P. D.; Kennedy, J.; Cooper, R.; Goble, C. A.; West, A. and Smyth, M.. Teallach: A model-based user interface development environment for object databases. P. 31-68. *Interacting with Computers* 14, 1, 2001.
- (ISO, 1997) ISO – International Organization for Standardization. *ISO 9241 –Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) –Part 14: Menu Dialogues*. Geneva, 1997.
- (ISO, 1998) ISO – International Organization for Standardization. *ISO 9241 –Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) – Part 17: Form filling dialogues*. Geneva, 1999.
- (ISO, 1999) ISO – International Organization for Standardization. *ISO 9241 –Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) – Part 16: Direct manipulation dialogues*. Geneva, 1999.
- (JAVA, 2007) *Java Platform, Standard Edition 6 Release*. Disponível em: <http://java.sun.com/javase/6/>. Acesso em: Junho de 2007.
- (JavaHelp System, 2007) *JavaHelp*. Disponível em: <http://java.sun.com/products/javahelp/>. Acesso: Abril de 2007.
- (JDOM, 2004) Disponível em: <http://www.jdom.org/>. Acesso em: Dezembro de 2006.

- (JHotDraw, 2004) *Framework JHotDraw*. Disponível em: <http://www.randelshofer.ch/oop/jhotdraw/>. Acesso em: Outubro de 2006
- (JUnit, 2001) *Framework JUnit*. Disponível em: <http://www.junit.org/index.htm>. Acesso em: Outubro de 2006.
- (Kieras e Polson, 1985) Kieras, D.E.; e Polson, P.G. (1985). An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies*, 22, 365-394.
- (Libório *et. al.*, 2005) Andréia Libório, Elizabeth Furtado, Ismael Rocha, and Vasco Furtado. *Geração de interfaces através de sistemas baseados em conhecimento: uma abordagem centrada em explicações de modelos de resolução de problemas*. *CONFERENCIA LATINO-AMERICANA DE INTERACCION HUMANO-COMPUTADORA - CLIHC*, 2005.
- (Limbourg e Vanderdonckt, 2004) Limbourg, Q., Vanderdonckt, J. Addressing the Mapping Problem in User Interface Design with UsiXML. In *TAMODIA 2004*, p. 155-163, 2004.
- (Medeiros, Kafure e Lula Jr.,2000) Medeiros, H.; Kafure, I.; Lula Jr., B. TAOS: a task-and-action oriented framework for user's task analysis in the context of human-computer interfaces design. In: *XX International Conference of the Chilean Computer Science Society (SCCC'2000)*, Santiago de Chile, Chile, 2000.
- (Medeiros, Lula Jr. e Cordeiro, 2002) Medeiros, F.P.A.; Lula Jr., B.; Cordeiro, P. B.. A Graphical Tool to Support Task Description Using TAOS Formalism for UI Design. In *Proc. of the 7thERCIMWorkshop*, p. 45- 51, 2002.
- (Medeiros, 2003) Francisco P. A. Medeiros. *Projeto e implementação do módulo taos-graph da ferramenta itaos para análise e modelagem da tarefa*. Master's thesis, Universidade Federal de Campina Grande, Departamento de Ciência da Computação, Campina Grande, 2003.
- (Model-Driven Architecture Guide, 2003) The Model-Driven Architecture - Guide Version 1.0.1. *OMG Document:omg/2003-06-01*.

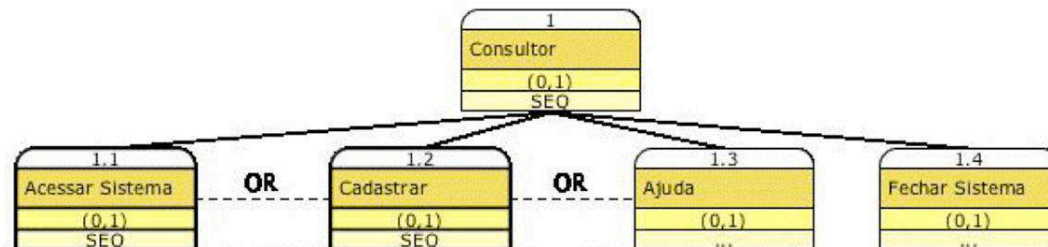
- (Molina, 2004) Molina, P. User interface generation with OlivaNova model execution system. *Intelligent User Interfaces 2004*: 358-359
- (Montero e López-Jaquero, 2006) Montero, F.; López-Jaquero, V., *IdealXML: An Interaction Design Tool-A Task-Based Approach to User Interfaces Design*, *Proc. of 6th Int. Conf. on Computer-Aided Design of User Interfaces CADUI'2006* (Bucharest, 6-8 June 2006), Chapter 20, Springer-Verlag, Berlin, 2006, pp. 245-252.
- (Mori, Paternò e Santoro, 2004) Mori, G.; Paternò, F; Santoro, C. Design and development of multidevice user interfaces through multiple logical descriptions. *IEEE Transactions on Software Engineering*, 30(8), 2004.
- (Myers, Hudson e Pausch, 2000) Myers, B.; Hudson, S. E. and Pausch, R. Past, present, and future of user interface software tools. *ACM Transaction. Comput.-Hum. Interact*, p.3-28, 2000.
- (Nielsen, 1993) Nielsen, J. *Usability Engineering*. Academic Press, Londres, 1993.
- (Pederiva et. al., 2007) Pederiva, I., Vanderdonckt, J., España, S., Panach, I., Pastor, O., The Beautification Process in Model-Driven Engineering of User Interfaces, *Proc. of 11th IFIP TC 13 Int. Conf. on Human-Computer Interaction INTERACT'2007* (Rio de Janeiro, September 10-14, 2007), Lecture Notes in Computer Science, Vol. 4662, Springer-Verlag, Berlin, 2007, pp. 409-422.
- (Queiroz, 2001) Queiroz, J. E. R. *Abordagem híbrida para a avaliação da usabilidade de interfaces com o usuário*. 2001. 410 p. Tese (Doutorado em Engenharia Elétrica) – Pós-graduação em Engenharia Elétrica, Universidade Federal da Paraíba, Campina Grande, 2001.
- (Rodrigues, Lula Jr. e Suárez, 2005) Rodrigues, C. E. C. L, Lula Jr., B, Suárez, P. R., Using a script model to preserve the consistency within an UI design environment. *Proceedings of the 4th international workshop on Task models and diagrams*, 2005
- (Rodrigues, 2005) Rodrigues, C. E. C. L. *Meditate+: utilizando o processo de roteirização para a obtenção do modelo de interação Editor-estendido*. Dissertação (Mestrado em

Informática) – Ciência da Computação, Universidade Federal de Campina Grande, Campina Grande PB, Junho de 2005.

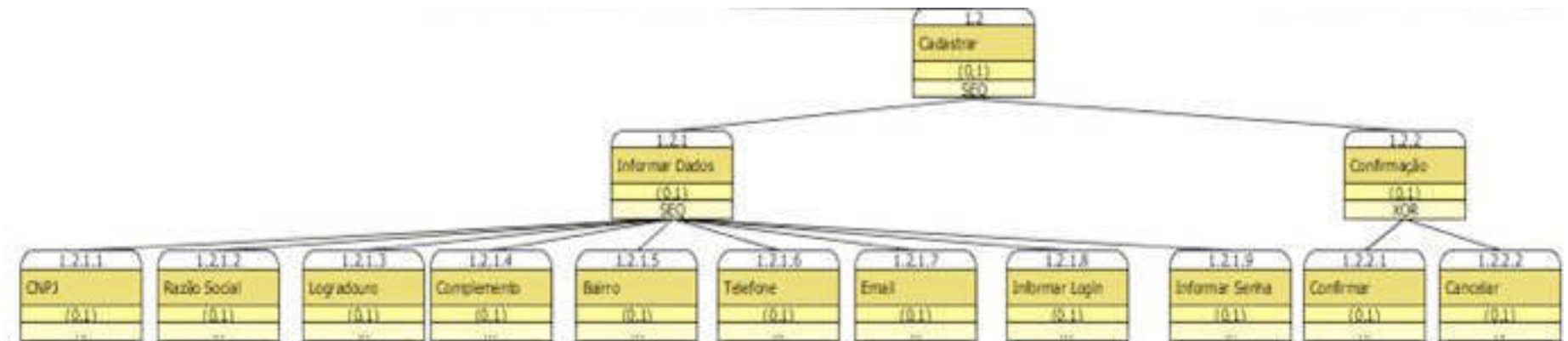
- (Schulungbaum e Elwert, 1996) E. Schlungbaum and T. Elwert. Automatic user interface generation from declarative models. In *CADUI*, 1996.
- (Suárez, Lula Jr. e Barros, 2004) Suárez, P. R., Lula, B., Barros, M. A. Applying knowledge management in UI design process. *Proceedings of the 3rd annual conference on Task models and diagrams, ACM International Conference Proceeding Series*; Vol. 86, 2004.
- (Stary, 2000) Stary, C. Contextual prototyping of user interface. In *DIS, Brooklyn, New York*, 2000.
- (Szekely, 1996) Szekely P. Retrospective and challenge for Model Based Development. *CADUI'96*, Namur, 1996. p. xxi-xliv
- (Vanderdonckt e Berquim, 1999) Vanderdonckt, J., Berquin, P.: Towards a Very Large Model-Based Approach for User Interface Development. In: Paton, N.W., Griffiths, T. (eds.): *Proc. of 1st IEEE Int. Workshop on User Interfaces to Data Intensive Systems UIDIS'99 (Edinburgh, September 5-6, 1999)*. *IEEE Computer Society Press*, Los Alamitos (1999) 76–85.

APÊNDICE A – Modelo da Tarefa com uso do Formalismo TAOS para o Sistema *Consultor*

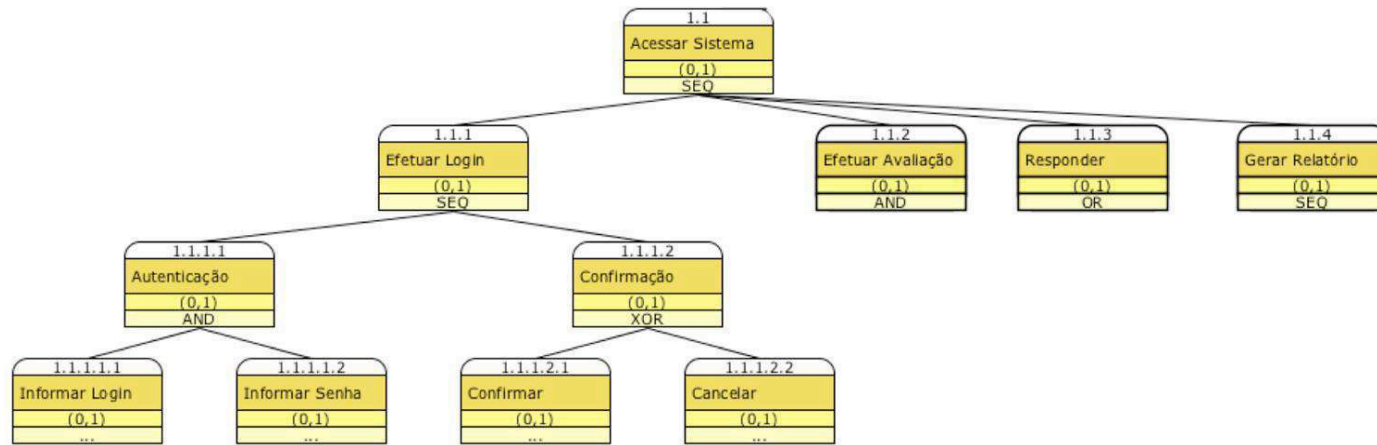
Tarefa Inicial 1 – Consultor



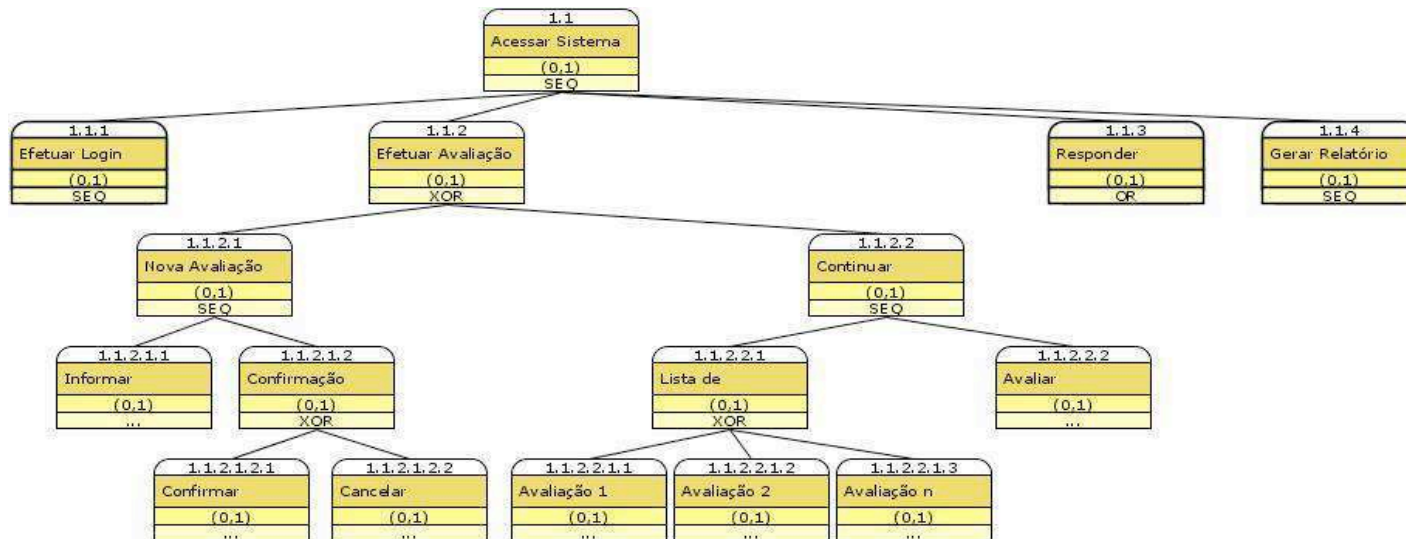
Tarefa 1.2 – Cadastrar Empresa



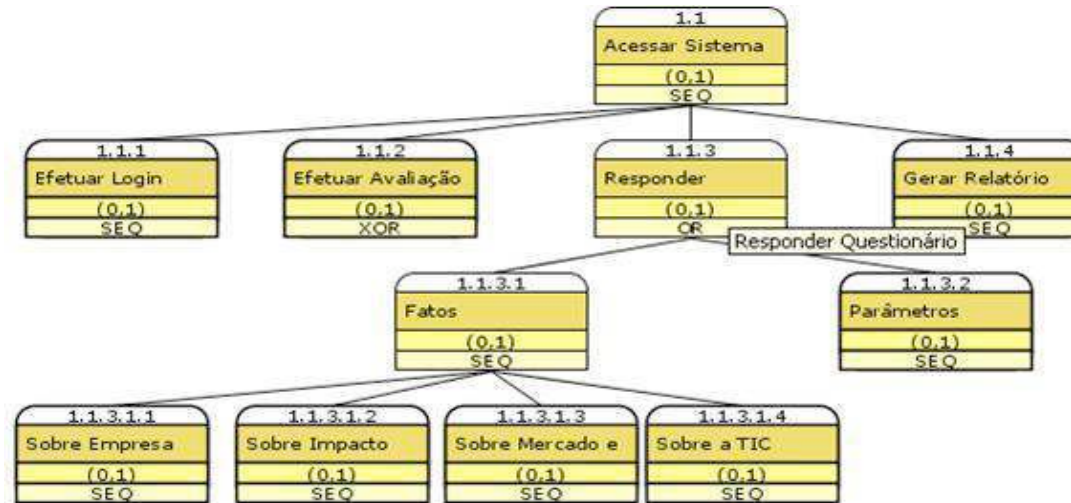
Tarefa 1.1.1 – Efetuar *Login*



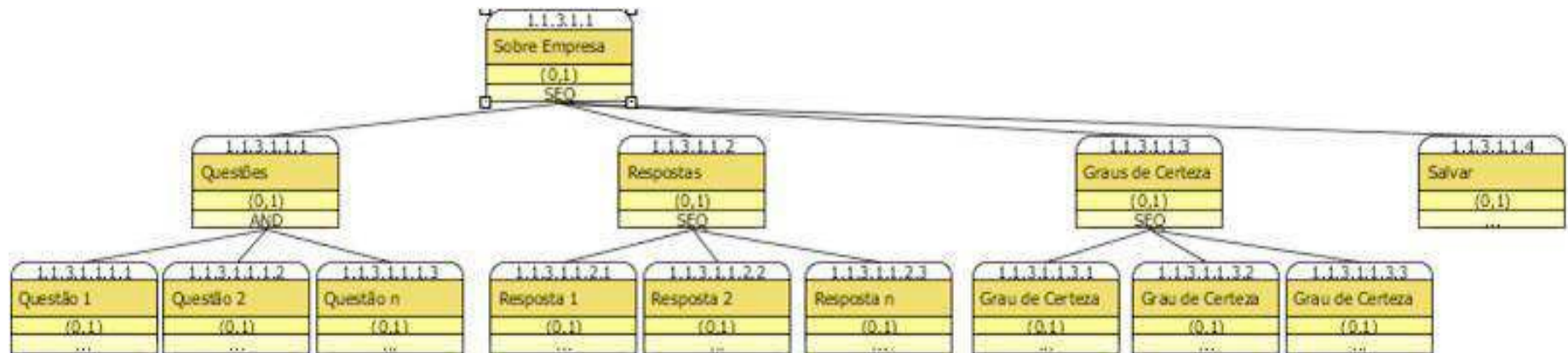
Tarefa 1.1.2 – Efetuar Avaliação



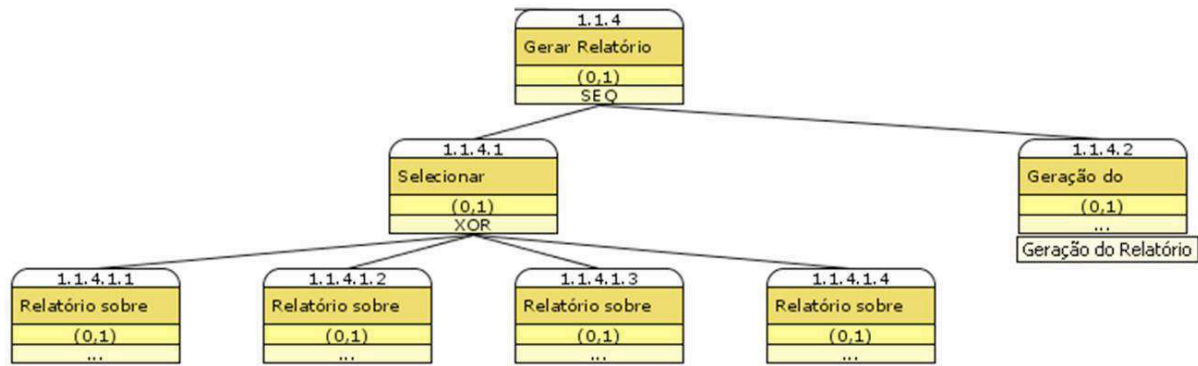
Tarefa 1.1.3 – Responder Questionário



Tarefa 1.1.3.1.1– Responder Questionário de Fatos sobre a Empresa



Tarefa 1.1.4 – Gerar Relatório




APÊNDICE C – Material de Nivelamento



Estudo de Caso
Material de Nivelamento

Karolyne Maria Alves de Oliveira
karolyne@dsc.ufcg.edu.br


www.dsc.ufcg.edu.br/~figroup



Conteúdo

- Introdução
- Desenvolvimento de Interface do Usuário (IU) Baseado em Modelos
- Prototipagem da Interface do Usuário
- Ambientes de Desenvolvimento de Interface Baseado em Modelos (MB-IDEs)
- Tutorial do Uso das Ferramentas
 - FastInterface
 - TERESA

www.dsc.ufcg.edu.br/~figroup




Introdução

- A codificação da IU pode representar entre **50% e 70%** do total de Linhas de Código
- Em média **44%** do tempo total do desenvolvimento do software é gasto com a IU

Desenvolvimento de IU é um processo muito oneroso


www.dsc.ufcg.edu.br/~figroup



Introdução

- Cenário Atual
 - Demanda por sistemas cada vez mais complexos;
 - Em geral, artefatos de usabilidade não são utilizados de modo a melhorar a qualidade da IU → Desenvolvimento Informal;
 - Concepção de IU sem apoio metodológico e ferramental adequado → dificuldade de produção


www.dsc.ufcg.edu.br/~figroup



Introdução

- Os requisitos do usuário podem ser explorados sob dois aspectos:
 - Desenvolvimento à mão livre → Interface editada manualmente de maneira subjetiva
 - Desenvolvimento suportados por algoritmos de transformação → IU é automaticamente gerada, sendo assumida como usável para o usuário final

www.dsc.ufcg.edu.br/~figroup



Introdução

- Para redução do tempo e custo no desenvolvimento de IU → abordagem que incorpore o uso de modelos como meio de gerar de forma automatizada seu código (Engenharia Dirigida Por Modelos)

www.dsc.ufcg.edu.br/~figroup



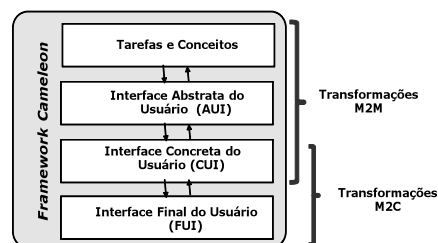
Engenharia Dirigida por Modelos

- Os requisitos do usuário são suportados por algoritmos de transformação
 - modelo-para-modelo (M2M)
 - modelo-para-código (M2C)

www.dsc.ufcg.edu.br/~figroup



Desenvolvimento de IU Baseado em Modelos Framework Cameleon



www.dsc.ufcg.edu.br/~figroup



Prototipagem de Interface do Usuário

- Gargalos de desenvolvimento de IU baseado em modelos:
 - O uso de Modelos não está associado a uma representação visual que permita vislumbrar a interface final
 - Técnicas de avaliação da interface final gerada é feita em cima de modelos declarativos

www.dsc.ufcg.edu.br/~figroup



Prototipagem de Interface do Usuário

- Solução: Técnicas de prototipagem em diferentes etapas do processo de *design*
 - Técnica Lo-fi (baixa fidelidade)
 - Técnica Me-fi (média fidelidade)
 - Técnica Hi-fi (alta fidelidade)

www.dsc.ufcg.edu.br/~figroup



Ambientes de Desenvolvimento Baseados em Modelos

	Conceitos e Modelo da Tarefa	Interface Abstrata	Interface Concreta	Interface Final
Mastermind	Modelo da Tarefa Hierárquico	Descritores	Protótipo Low-Fi	Gera código C++
TADEUS	TATAR	Script	Protótipo	-
Teallach	Modelo da Tarefa Hierárquico	Script	Protótipo Hi-Fi	Código Java Swing
TransformiXML	CCT	Esboços rudimentares	Protótipo Hi-Fi	XHTML
Teresa	CCT	Descritores Lógicos Múltiplos	Apresentação em Página Web (não editável)	XHTML
FastInterface	TAOS	Edição e Simulação em Protótipo Me-Fi	Edição e Simulação em Protótipo Hi-Fi	HTML

www.dsc.ufcg.edu.br/~figroup



Comparativo entre os MB-IDES

	Mastermind	TADEUS	Teallach	TransformiXML	Teresa	FastInterface
Múltiplos projetos						x
Mecanismo de consistência				x	x	x
Representação visual da AUI				x		x
Edição e simulação da AUI						x
Representação visual na CUI	x	x	x	x	x	x
Edição da CUI	x	x	x			x
Geração de código	x		x	x	x	x
Multiplataforma				x	x	x
Multi-fidelidade						x

www.dsc.ufcg.edu.br/~figroup



Estudo de Caso - Objetivo

- Gerar e editar protótipos de interface do usuário e código da interface final através da utilização do Modelo da Tarefa como artefato base.

www.dsc.ufcg.edu.br/~figroup



Modelo da Tarefa

- Descreve os objetivos que o usuário pretende alcançar, e as ações que ele tem que realizar para alcançar tais objetivos
- Representa de forma hierárquica os passos que o usuário deve seguir para atingir tal objetivo

www.dsc.ufcg.edu.br/~figroup



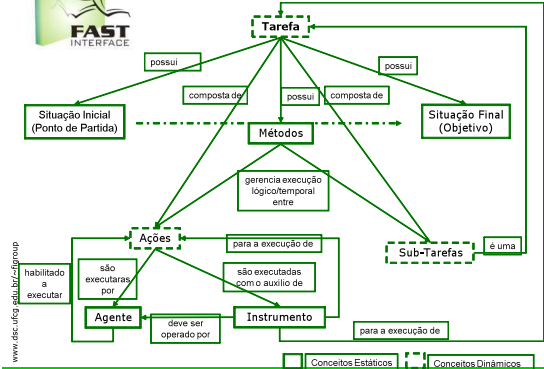
Modelo da Tarefa

- Os Formalismos adotados para o Modelo da Tarefa são TAOS e CTT
 - Task and Action Oriented System
 - Concur Task Tree

www.dsc.ufcg.edu.br/~figroup



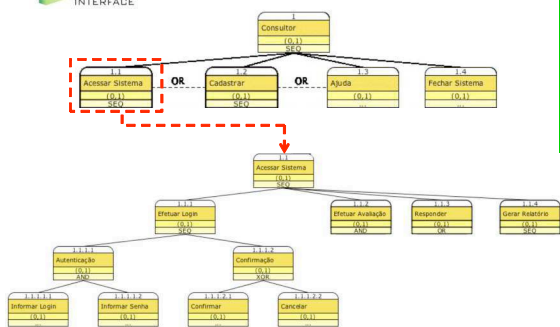
Modelo da Tarefa – Formalismo TAOS



www.dsc.ufcg.edu.br/~figroup



Modelo da Tarefa – TAOS Exemplo



www.dsc.ufcg.edu.br/~figroup



Modelo da Tarefa – Formalismo CTT

Alocação de Tarefas

- Agente ⇔ **Quem** realiza a tarefa
 - Tarefa do **Usuário**
 - Realização na totalidade pelo usuário
 - Tarefa de **Interação**
 - Interação do usuário com o sistema, iniciada pelo usuário
 - Tarefa do **Sistema**
 - Realização pelo sistema, sem a participação do usuário
 - Tarefa **Abstrata**
 - Tarefa mais complexa

www.dsc.ufcg.edu.br/~figroup



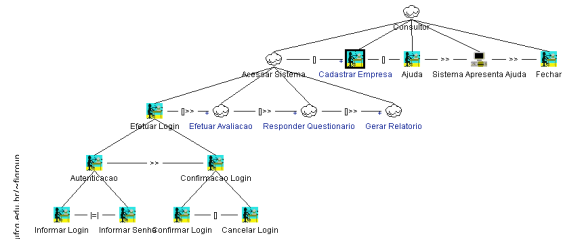
Modelo da Tarefa – Formalismo CTT

- **Objetos e Atributos das Tarefas**
 - Identificação dos objetos a serem manipulados na realização da tarefa
 - Objetos: internos ou perceptíveis
 - Informações sobre a tarefa
 - Tipo ⇨ Seleção, edição, monitoração
 - Plataforma ⇨ Desktop, PDA, celular
 - Frequência de uso ⇨ Baixa, média, alta
 - Pré e Pós-condições
 - Tempo estimado para a realização

www.dsc.ufmg.edu.br/~fgroup



Modelo da Tarefa – CTT Exemplo



www.dsc.ufmg.edu.br/~fgroup

APÊNDICE D – Requisitos Funcionais do Sistema *Consultor*

ESTUDO DE CASO

Ferramenta Consultor

Karolyne Maria Alves de Oliveira
Julho de 2008

Roteiro

- O Sistema
- Requisitos Funcionais
- Modelagem da Tarefa

Consultor

- Ferramenta de Apoio à Decisão sobre Adoção de Tecnologias da Informação e Comunicação (TIC) nas Empresas
- Usuário
 - Empresa que deseja avaliar novas tecnologias
- *Web/Standalone*

Requisitos Funcionais

- Cadastrar Empresa;
- Efetuar *Login*;
- Avaliar nova avaliação da TIC ou continuar uma já existente;
- Responder questionários sobre fatos e parâmetros relacionados a TIC, empresa, mercado/governo e custos;
- Gerar relatórios com resultados sobre o risco, estratégia, ROI a respeito da TIC avaliada.

Requisitos Funcionais

Cadastrar Empresa

- Para o cadastro devem ser informados
 - CNPJ
 - Razão Social
 - Logradouro
 - Complemento
 - Bairro
 - Cidade
 - Estado
 - E-mail
 - URL
 - Login e Senha desejados

Requisitos Funcionais

Efetuar *Login*

- O usuário do Consultor entra no sistema informando:
 - *Login*
 - Senha

Requisitos Funcionais
Efetuar Avaliação

- O usuário logado pode iniciar uma nova avaliação ou continuar uma já existente;
 - O cadastro de uma avaliação consiste na informação do nome da TIC a ser avaliada;
 - No caso de continuação, o usuário escolhe uma avaliação que deve estar disponível em uma lista de avaliações iniciadas;

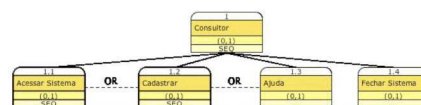
Requisitos Funcionais
Responder Questionários

- Escolhida a avaliação da TIC, o usuário responderá questionários sobre Fatos (valores fixos) e Parâmetros (valores variados);
 - Questionário sobre fatos está dividido em: Fatos sobre a Empresa, sobre a TIC, sobre o Mercado/Governo e o impacto da TIC na Empresa;
 - Questionário sobre parâmetros está dividido em: Parâmetros sobre a empresa, a TIC e o Mercado/Governo.

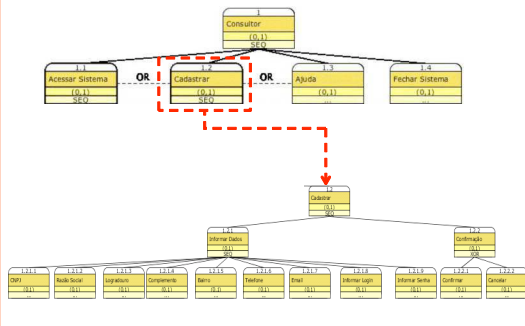
Requisitos Funcionais
Gerar Relatórios

- Após ter respondido todas os questionários, o usuário fica apto a gerar relatórios sobre a tecnologia
- Os relatórios disponíveis são:
 - Relatório sobre a Estratégia
 - Relatório sobre o Risco
 - Relatório sobre o Custo
 - Relatório sobre Retorno do Investimento (ROI)

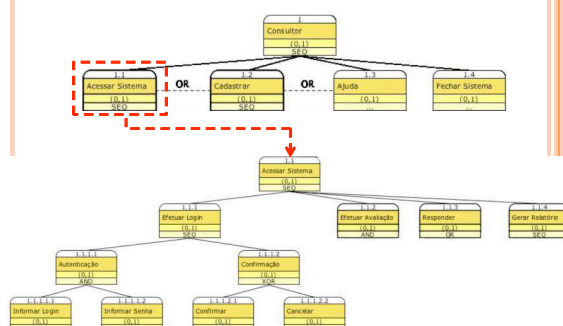
Modelo da Tarefa – Ferramenta Consultor



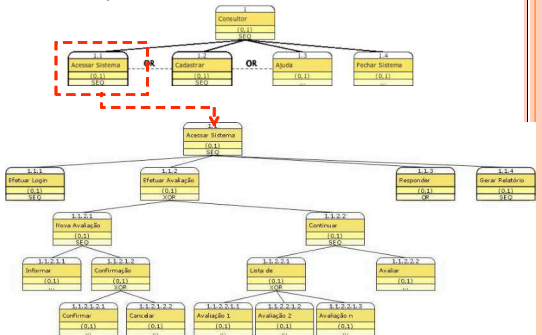
Modelo da Tarefa – Cadastrar Empresa



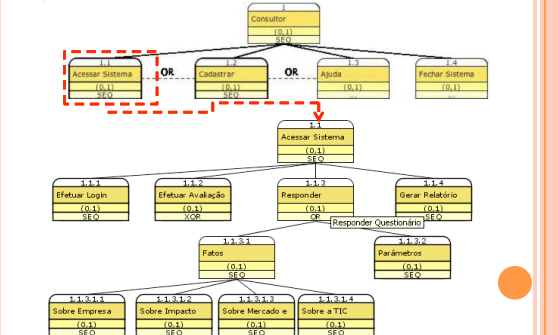
Modelo da Tarefa – Efetuar Login



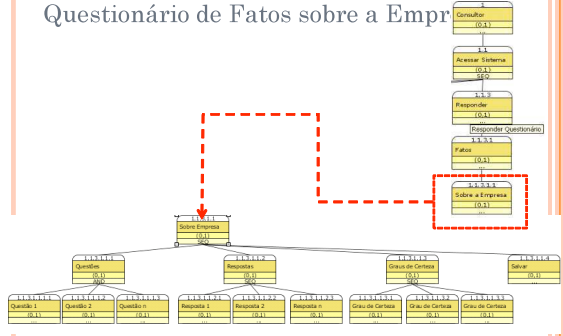
Modelo da Tarefa – Efetuar Avaliação



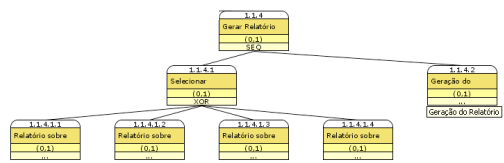
Modelo da Tarefa – Responder Questionário



Modelo da Tarefa – Responder Questionário de Fatos sobre a Empr




Modelo da Tarefa – Gerar Relatório



APÊNDICE E – Tutorial de Uso dos Ambientes FastInterface e TERESA

TUTORIAL de USO



FAST
INTERFACE

Karolyne Maria Alves de Oliveira
karolyne@dsc.ufcg.edu.br

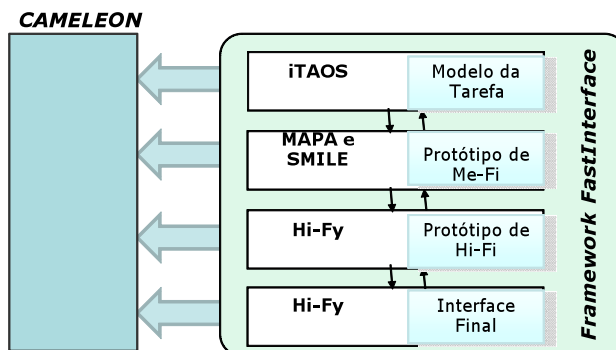


FastInterface

- *Ambiente de Desenvolvimento Baseado em Modelos e Múltipla Prototipagem Evolutiva*
- *Disponível em:*
www.dsc.ufcg.edu.br/~figroup/FAST



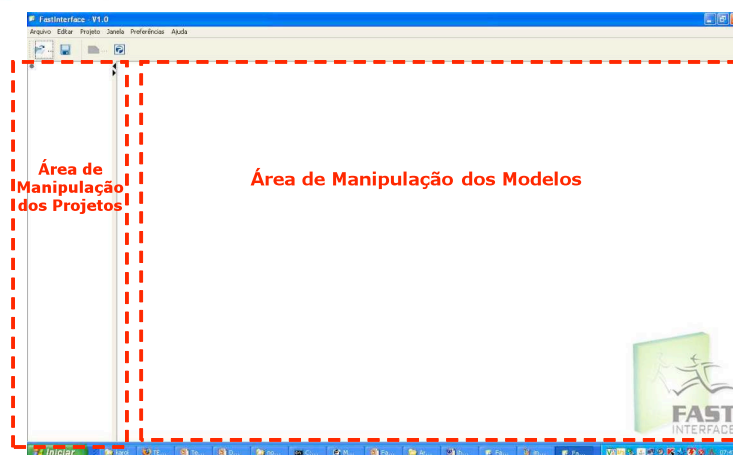
Framework FastInterface



www.dsc.ufcg.edu.br/~figroup



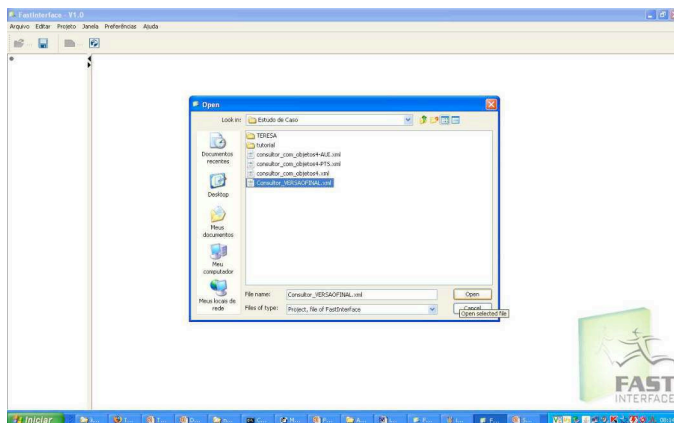
FastInterface – Tela Inicial



www.dsc.ufcg.edu.br/~figroup



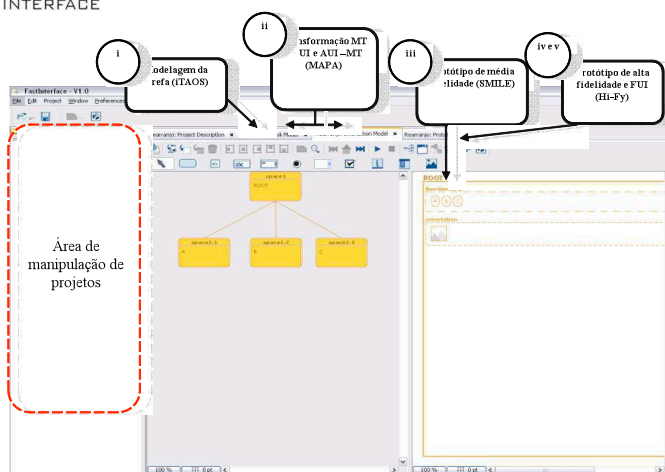
Abrindo XML do Projeto da Interface



www.dsc.ufcg.edu.br/~figroup



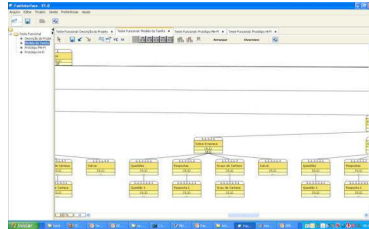
Ambiente FastInterface



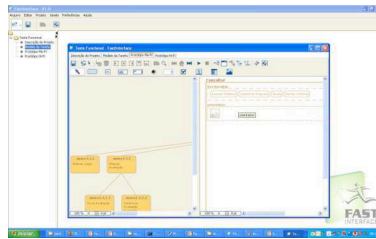
www.dsc.ufcg.edu.br/~figroup



Modos de Visualização



Modo Internal Tab

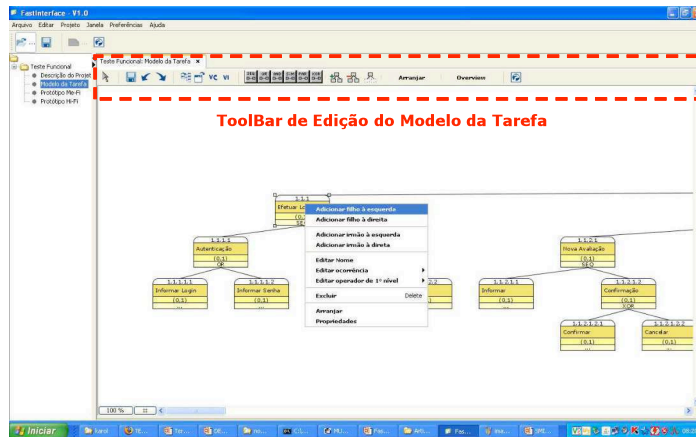


Modo Janela Externa

www.dsc.ufcg.edu.br/~figroup



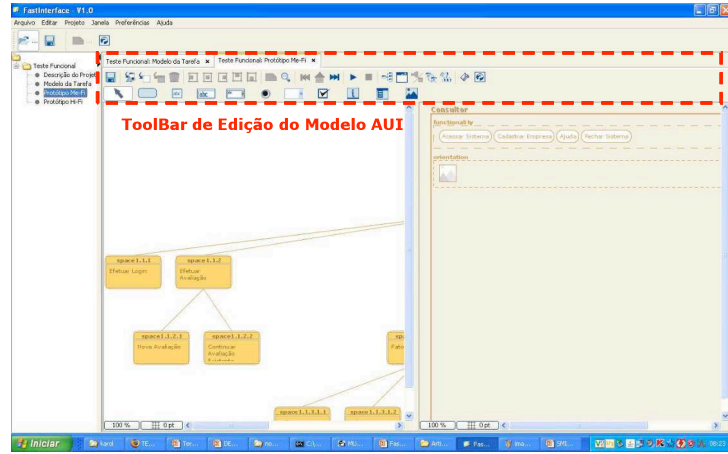
Edição do Modelo da Tarefa



www.dsc.ufcg.edu.br/~figroup



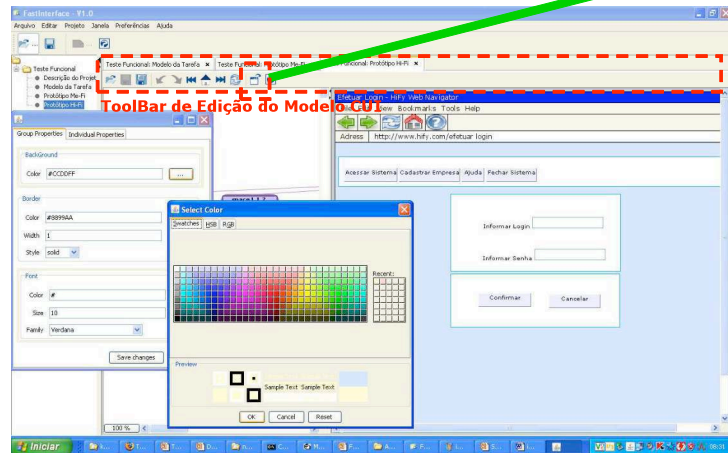
Edição do Protótipo de Média Fidelidade



www.dsc.ufcg.edu.br/~figroup



Edição do Protótipo de Alta Fidelidade



Exportar Código

www.dsc.ufcg.edu.br/~figroup



Principais Funções

- dar suporte à análise e modelagem da tarefa;
- Implementar o mapeamento entre os elementos do modelo da tarefa (MT) e os elementos do modelo da interação (interface abstrata - AUI);
- permitir a edição e simulação do modelo da interface abstrata a partir da edição e simulação de protótipos de média fidelidade;
- dar suporte à geração e edição de protótipos de alta fidelidade da interface;
- geração do código da interface final;
- oferecer manutenção da consistência entre os modelos envolvidos na concepção da interface do usuário em tempo real de projeto;
- possibilitar a concepção de múltiplos projetos simultaneamente .



TERESA

Tutorial de Uso

Karolyne Maria Alves de Oliveira



TERESA

- Uma Ambiente de Desenvolvimento de Interface do Usuário Baseado em Modelos com Múltiplos Descritores Lógicos
- <http://giove.isti.cnr.it/teresa>

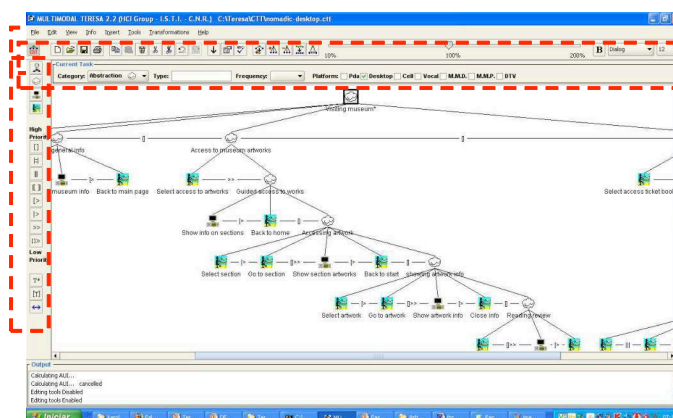
Ambiente Teresa – Tela Principal

Opções de Menu

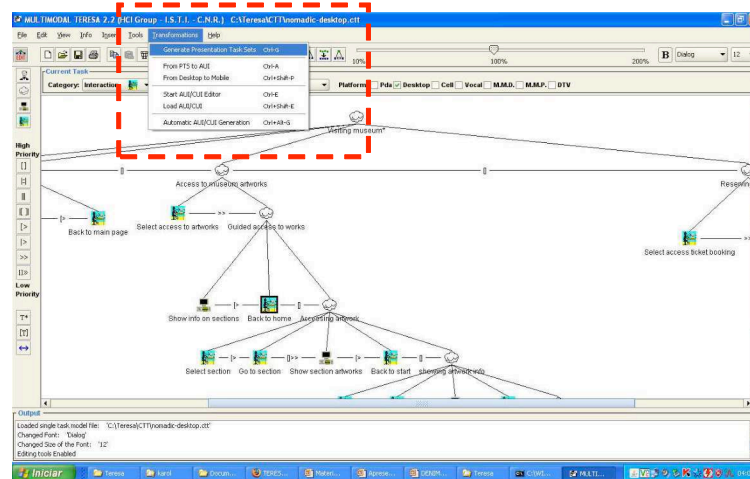


Ativar Ferramenta TERESA

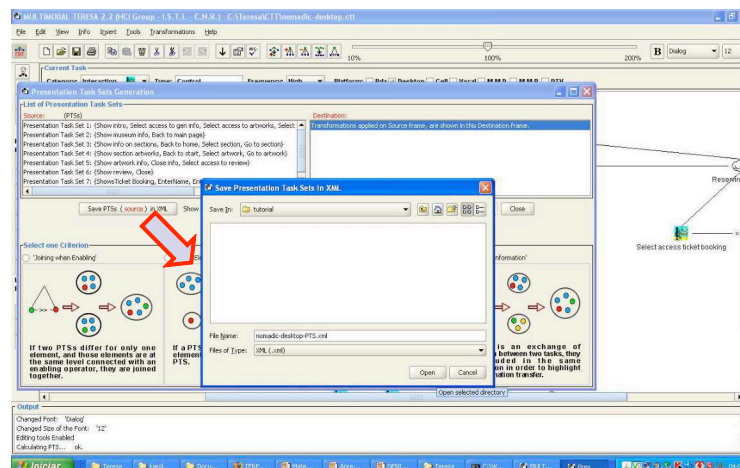
Barras de Manipulação do Modelo da Tarefa



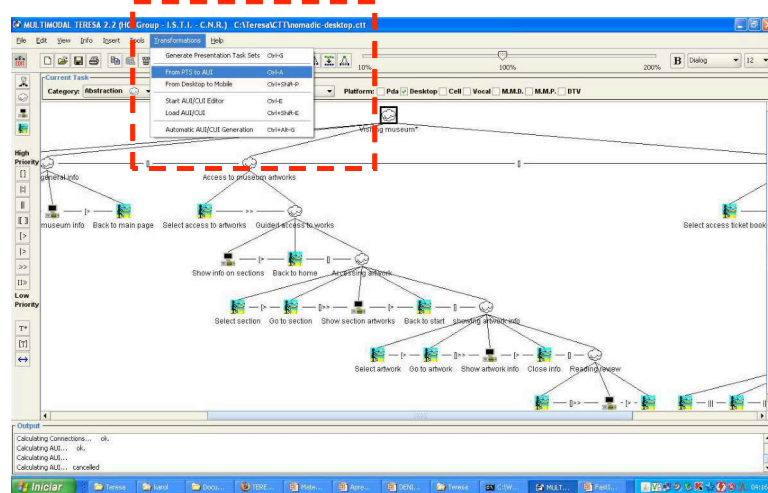
Gerar PTS a partir do Modelo da Tarefa



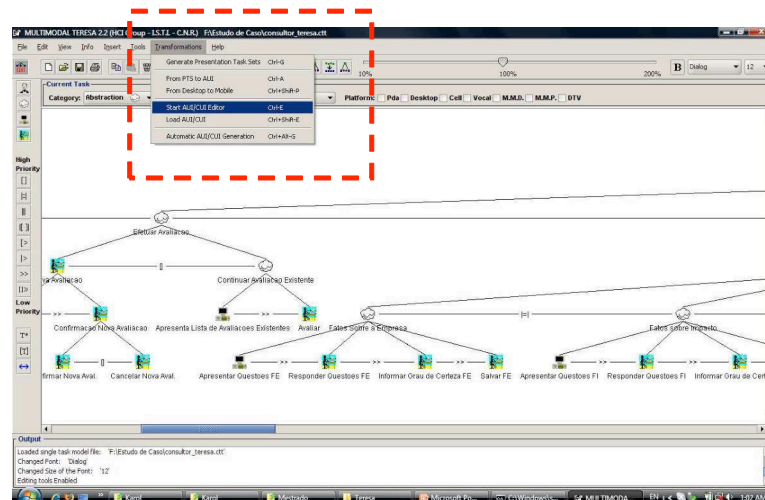
Salvar PTS



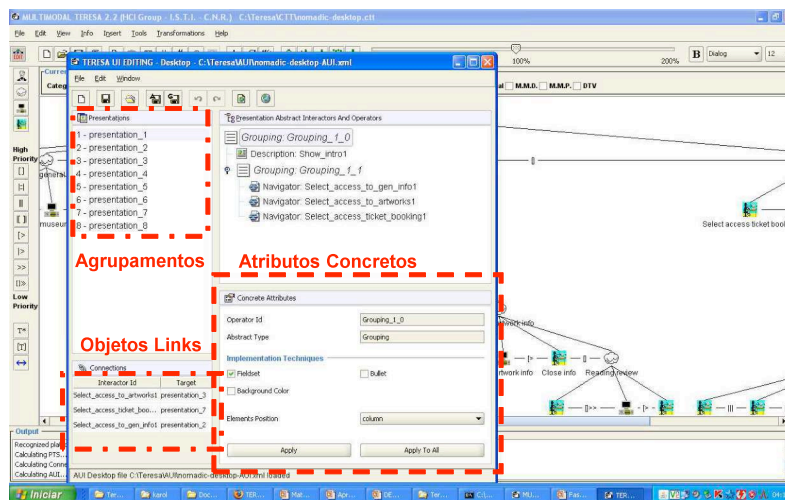
Geração da AUI (Através da PTS)



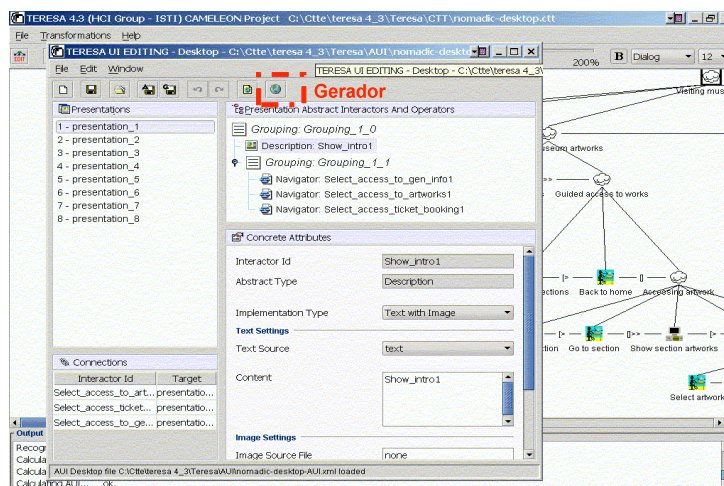
Modo de Edição AUI/CUI



Edição da AUI/CUI



Geração da Interface Final





Demonstração



Outras Informações Tutorial Avançado

- Como usar a ferramenta
 - <http://giove.isti.cnr.it/teresa/howToUse.html>
- Vídeo Demonstrativo
 - <http://giove.isti.cnr.it/teresa/videos.html>

**APÊNDICE F – Roteiros de Teste de Usabilidade para ambientes
FastInterface e TERESA**

**Roteiro de Teste de Usabilidade do
Uso do Ambiente FastInterface**

Karolyne Maria
Alves de Oliveira

Atividade 01: Abrir Projeto já existente

Roteiro: Nesta atividade você irá entrar no sistema *FastInterface* e abrir um arquivo, no formato *FastInterface XML*, previamente existente.

Instruções:

- Escolha a opção de Arquivo -> Abrir
- Escolha o caminho do arquivo – C:\Consultor.XML
- Informe o nome do projeto, do projetista e uma descrição para o projeto Consultor

Atividade 02: Abra a Aba do Modelo da Tarefa

Roteiro: Ao abrir um projeto, o sistema habilita a opção de edição nos modelos existentes.

Instruções:

- Vá na área de manipulação de projetos e dê duplo click no modelo da tarefa.

Atividade 03: Editar Modelo da Tarefa

Roteiro: Nesta atividade, você irá modificar o modelo da tarefa previamente modelado.

Instruções:

- Localize a tarefa Cadastrar Empresa e verifique se os requisitos do usuário estão sendo atendidos;
- Acrescente um irmão a direita da ação complemento. O nome desta Ação será CEP;

Atividade 04: Gerar Modelo de Interface Abstrata

Roteiro: Nesta atividade, gerar-se-á o modelo de interface abstrata baseado no modelo da tarefa previamente modelado.

Instruções:

- Abra a aba de Protótipo Me-Fi;
- Verifique se há uma árvore de esboços no lado esquerdo da aba e um protótipo corrente do lado direito;
- Verifique se a aba de protótipos de alta fidelidade reflete os de média.

Atividade 05: Editar Protótipo de Média Fidelidade

Roteiro: Nesta atividade você irá editar os protótipos de média fidelidade de modo a deixá-los fiéis aos requisitos do usuário. Nesta etapa, há possibilidade de modificar localização, tipo do objeto e o tamanho deles.

Instruções

- Varra nodo por nodo da árvore de esboço, dando dois cliques em cima da figura, e edite os protótipos de média fidelidade correspondente a ele. Esta edição deve refletir a tela que você deve disponibilizar para o usuário validar;
- Para edição no esboço, use as opções da barra de menu e o botão direito em cima dos objetos.

Atividade 06: Manutenção da Consistência entre AUI e Modelo da Tarefa

Roteiro: Nesta atividade você irá realizar operações que disparam o mecanismo de consistência da ferramenta. O foco da manutenção é do nível AUI para o nível Tarefas e Conceitos.

Instruções

- Localize o esboço Cadastrar Empresa e remova o objeto de interação CEP;
- Rearrange a disposição dos elementos
- Vá para a aba da Tarefa e verifique se o sistema removeu a tarefa CEP

Atividade 07: Editar Protótipo de Alta Fidelidade

Roteiro: Nesta atividade você irá editar os protótipos de alta fidelidade de modo a deixá-los fieis aos requisitos do usuário. Nesta etapa, há possibilidade de modificar os estilos dos protótipos. Caso o protótipo não esteja bem representado em nível de localização e tamanho, edite o protótipo de média fidelidade correspondente a ele

Instruções

- Varra nodo por nodo da árvore de protótipos, dando duplo click em cima da figura ou utilizando a opção de navegação disponibilizado na toolbar;
- Clicando com o botão direito em cima das figuras, é possível editar os protótipos;

Atividade 08: Gerar Interface Final

Roteiro: Nesta atividade você irá exportar as telas editadas

Instruções

- Clique no botão de exportação localizado na aba de protótipo de alta fidelidade;

Roteiro de Teste de Usabilidade do Ambiente TERESA

Karolyne Maria
Alves de Oliveira

Atividade 01: Abrir Modelo da Tarefa já existente

Roteiro: Nesta atividade você irá entrar no sistema *Teresa* e abrir um arquivo, no formato FastInterface XML, previamente existente.

Instruções:

- Escolha a opção de File -> Open
- Escolha o caminho do arquivo – C:\Consultor_teresa.CTT

Atividade 02: Habilite a área de edição da tarefa

Roteiro: Ao abrir um projeto, o sistema habilita a opção de edição nos modelos existentes.

Instruções:

- Vá na área de manipulação de projetos e dê um *click* no botão de habilitar edição do modelo da tarefa.

Atividade 03: Editar Modelo da Tarefa

Roteiro: Nesta atividade modificar o modelo da tarefa previamente modelado.

Instruções:

- Localize a tarefa Cadastrar Empresa e verifique se os requisitos do usuário estão sendo atendidos;
- Acrescente um irmão a direita da ação de interação complemento. O nome desta Ação será CEP;
- Coloque os operadores de independência de ordem que fica localizado no menu lateral esquerdo da tela.

Atividade 04: Gerar Modelo de Interface Abstrata

Roteiro: Nesta atividade, gerar-se-á o modelo de interface abstrata baseado no modelo da tarefa previamente modelado.

Instruções:

- Vá até o menu transformações, gere as PTS e em seguida o modelo AUI.

Atividade 05: Editar Modelo de Interface Abstrata

Roteiro: Nesta atividade você irá editar o modelo de interface abstrata de modo a deixá-los fiéis aos requisitos do usuário. Nesta etapa, há possibilidade de modificar os links dos objetos, remover ou adicionar novos elementos a interface e mudar os tipos de apresentação dos objetos.

Instruções

- Varra todas as telas de apresentação disponibilizadas pela aplicação e verifique se os elementos são agrupados de maneira correta;
- Caso necessário, realize modificações em termos de navegação e apresentação.

Atividade 06: Manutenção da Consistência entre AUI e Modelo da Tarefa

Roteiro: Nesta atividade você irá realizar operações que disparam inconsistências entre o modelo da tarefa e o modelo AUI. O foco da manutenção é do nível AUI para o nível Tarefas e Conceitos.

Instruções

- Localize o esboço Cadastrar Empresa e remova o objeto de interação CEP;
- Vá para a aba da Tarefa e verifique a manutenção da consistência.

Atividade 07: Editar Protótipo de Alta Fidelidade

Roteiro: Nesta atividade você irá editar os protótipos de alta fidelidade de modo a deixá-los fieis aos requisitos do usuário. Nesta etapa, há possibilidade de modificar os estilos dos protótipos.

Instruções

- Varra todas as telas de apresentação disponibilizadas pela aplicação e edite os estilos e aspectos de apresentação do sistema Consultor;

Atividade 08: Gerar Interface Final

Roteiro: Nesta atividade você irá exportar as telas editadas

Instruções

- Clique no botão de exportação localizado na tela de edição AUI/CUI;

APÊNDICE G – E-mail para Recrutamento de Usuários para o Estudo de Caso

Olá pessoal,

como todos sabem, estou desenvolvendo no meu mestrado um ambiente de desenvolvimento de interfaces do usuário baseado em modelos. Este ambiente está sendo disponibilizado em sua primeira versão essa semana e precisa ser validado através de um estudo de caso.

O estudo de caso consiste na construção de protótipos e código da interface final para uma ferramenta de apoio a tomada de decisão sobre adoção de novas tecnologias da informação nas empresas. Para tal, faremos o uso de ambientes de suporte à concepção de interfaces do usuário.

Os participantes trabalharão em duplas e terão como ponto de partida os requisitos do sistema e sua modelagem através de um modelo da tarefa (artefato base para o desenvolvimento de interfaces baseado em modelos).

O tempo necessário para o nivelamento não deve exceder 2 horas. A execução do estudo de caso também não deve exceder sessões de 2 horas.

O nivelamento realizar-se-á na próxima quinta-feira (24/07) às 9h e o estudo de caso na sexta-feira (25/07/2008) a partir das 8:30 hs.

Gostaria de contar com a colaboração de todos na realização desta etapa do meu trabalho e já agradeço a disponibilidade. Caso alguém tenha algum problema com horário ou qualquer outra dúvida, me comunique.

Um grande abraço,

Karolyne Oliveira.

APÊNDICE H - Questionário *USer (UserSketcher)*

1. Seu grau de instrução

Graduando Graduado Mestrando Mestre Outros

2. Você é do sexo

Masculino Feminino

3. Você pertence à faixa etária de

Menor que 18 Entre 18 e 23 Entre 23 e 28 Acima de 28

4. Quando você desenvolve interfaces do usuário, costuma fazer protótipos?

Sim Não

5. Em caso afirmativo, como você cria esses protótipos?

Lápis e papel Power-Point HTML Outro. Qual? _____

6. Você conhece os conceitos relacionados à modelagem da tarefa do usuário?

Sim Não

7. Você já modelou alguma tarefa do usuário?

Sim Não

8. Você utilizou alguma ferramenta para modelar a tarefa do usuário?

Sim Não

9. Em caso afirmativo, qual(is) ferramenta(s) de modelagem da tarefa você já utilizou?

iTAOS CTTE Outra(s). Qual(is)? _____

APÊNDICE I - Questionário *USE (UserSatisfactionEnquirer)*

- 1. O uso do ambiente de desenvolvimento na construção de protótipos é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil

- 2. O uso da ferramenta na edição de protótipos é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil

- 3. O uso da ambiente na construção e edição de protótipos é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil

- 4. A transformação e edição entre os modelos em diferentes etapas do projeto utilizando o ambiente é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil

- 5. A visualização e compreensão dos diferentes modelos modelos envolvidos no processo de concepção é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil

- 6. A recuperação de situações de erro ao utilizar o ambiente é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil

- 7. A recuperação de situações de travamento ao utilizar o ambiente é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil

- 8. A compreensão das mensagens de erro apresentadas pelo ambiente é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil

- 9. A navegação através das diferentes opções de menu, janelas de diálogo e barra de ícones do ambiente é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil

10. O uso das funcionalidades mais comuns do sistema é

Muito Fácil Fácil Nem Fácil Nem Difícil Difícil Muito Difícil

11. A compreensão da estruturação dos menus e barras de ícones é

Muito Fácil Fácil Nem Fácil Nem Difícil Difícil Muito Difícil

12. A compreensão das informações de interesse existentes na ajuda da ferramenta é

Muito Fácil Fácil Nem Fácil Nem Difícil Difícil Muito Difícil

13. O uso de protótipos de baixa-fidelidade auxiliou a modelagem da tarefa

Concordo Totalmente Concordo Nem Concordo Nem Discordo
 Discordo Discordo Totalmente

14. O uso de diferentes técnicas de prototipagem em diferentes estágios do processo de concepção de interfaces do usuário auxilia o desenvolvimento

Concordo Totalmente Concordo Nem Concordo Nem Discordo
 Discordo Discordo Totalmente

15. As respostas da ferramenta às minhas ações são muito lentas

Concordo Totalmente Concordo Nem Concordo Nem Discordo
 Discordo Discordo Totalmente

16. Consigo executar as tarefas de modo direto ao usar a ferramenta

Concordo Totalmente Concordo Nem Concordo Nem Discordo
 Discordo Discordo Totalmente

17. Acho que a ferramenta atende plenamente às minhas necessidades

Concordo Totalmente Concordo Nem Concordo Nem Discordo
 Discordo Discordo Totalmente

18. Perco muito tempo tentando aprender a usar a ferramenta

Concordo Totalmente Concordo Nem Concordo Nem Discordo
 Discordo Discordo Totalmente

19. Consigo fazer exatamente o que desejo com os recursos oferecidos pela ferramenta

- Concordo Totalmente Concordo Nem Concordo Nem Discordo
 Discordo Discordo Totalmente

20. A ferramenta apresenta tantas funcionalidades que as vezes acho que nunca aprenderei ou terei necessidade de usá-las todas

- Concordo Totalmente Concordo Nem Concordo Nem Discordo
 Discordo Discordo Totalmente

21. Sinto-me receoso em alguns momentos ao usar a ferramenta

- Concordo Totalmente Concordo Nem Concordo Nem Discordo
 Discordo Discordo Totalmente

22. Não acho que as informações da ajuda são suficientemente eficazes para tirar minhas dúvidas

- Concordo Totalmente Concordo Nem Concordo Nem Discordo
 Discordo Discordo Totalmente

23. Em alguns momentos sinto-me frustrado com o modo como a ferramenta executa tarefas de meu interesse

- Concordo Totalmente Concordo Nem Concordo Nem Discordo
 Discordo Discordo Totalmente

24. Sinto-me satisfeito com o resultado que obtive ao final da sessão

- Concordo Totalmente Concordo Nem Concordo Nem Discordo
 Discordo Discordo Totalmente

25. Acredito que o modelo da tarefa seja um ótimo ponto de partida para a construção de interfaces do usuário

- Concordo Totalmente Concordo Nem Concordo Nem Discordo
 Discordo Discordo Totalmente

26. Acredito que a abordagem baseada em modelos favorece na redução de esforço na construção de interfaces do usuário

- Concordo Totalmente Concordo Nem Concordo Nem Discordo
 Discordo Discordo Totalmente

