

**INSTITUTO DE PESQUISAS TECNOLÓGICAS DO ESTADO DE SÃO PAULO**

**MARIA CRISTINA MACHADO DOMINGUES**

***Estudo comparativo de extensões UML  
nos artefatos do projeto de sistemas interativos***

São Paulo

2008

**Maria Cristina Machado Domingues**

**Estudo comparativo de extensões UML  
nos artefatos do projeto de sistemas iterativos**

Maria Cristina Machado Domingues

Estudo comparativo de extensões UML  
nos artefatos do projeto de sistemas interativos

Dissertação apresentada ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo – IPT, como parte dos requisitos para obtenção do título de Mestre em Engenharia da Computação.

Data da Aprovação 15/08/2008.

---

Profa. Dra Lucia Vilela Leite Filgueiras  
IPT – Instituto de Pesquisas Tecnológicas  
do Estado de São Paulo.

Membros da Banca Examinadora:

Profa. Dra Lucia Vilela Leite Filgueiras (Orientador)

IPT – Instituto de Pesquisas Tecnológicas do Estado de São Paulo.

Profa. Maria Cecília Calani Baranauskas

UNICAMP – Universidade Estadual de Campinas (Membro)

Prof. José Eduardo Zindel Deboni

IPT – Instituto de Pesquisas Tecnológicas do Estado de São Paulo (Membro)

INSTITUTO DE PESQUISAS TECNOLÓGICAS DO ESTADO DE SÃO PAULO

MARIA CRISTINA MACHADO DOMINGUES

Estudo comparativo de extensões UML  
nos artefatos do projeto de sistemas interativos

Dissertação apresentada ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo – IPT, como parte dos requisitos para obtenção do título de Mestre em Engenharia da Computação.

Área de concentração: Engenharia de Software.

Orientadora: Profa Dra. Lucia Vilela Leite Filgueiras.

São Paulo  
agosto/2008

Ficha Catalográfica

Elaborada pelo Departamento de Acervo e Informação Tecnológica – DAIT  
do Instituto de Pesquisas Tecnológicas do Estado de São Paulo - IPT

**D671e Domingues, Maria Cristina Machado**

Estudo comparativo de extensões UML nos artefatos do projeto de sistemas interativos. / Maria Cristina Machado Domingues. São Paulo, 2008.

74p.

Dissertação (Mestrado em Engenharia de Computação) - Instituto de Pesquisas Tecnológicas do Estado de São Paulo. Área de concentração: Engenharia de Software.

Orientadora: Profa. Dra. Lucia Vilela Leite Filgueiras

1. UML (Unified Modeling Language) 2. Interação usuário-computador 3. Engenharia de software 4. Tese I. Instituto de Pesquisas Tecnológicas do Estado de São Paulo. Coordenadoria de Ensino Tecnológico II. Título

## DEDICATÓRIA

Dedico este trabalho ao meu pai Paulo Roberto Machado (*in memoriam*).

## AGRADECIMENTOS

Minha sincera gratidão:

à Deus;

ao meu marido Marcos e meu filho Rafael, pela paciência e apoio;

aos meus pais, pela minha formação pessoal;

aos amigos do IPT/CIAM pelo incentivo e pela oportunidade de aprendizado e formação profissional;

à minha Orientadora, Dr<sup>a</sup>. Lúcia Filgueiras, conhecimento, paciência e compreensão;

a todos os demais que, de alguma forma, contribuíram para a realização deste trabalho.

## RESUMO

Este trabalho estuda as extensões UML utilizadas no design de interfaces com usuário. A UML é amplamente utilizada em artefatos do processo de desenvolvimento de software, mas ela não cobre todas as necessidades de um desenvolvimento de sistema interativo. Muitas extensões foram propostas na literatura, no entanto, uma representação única, não é suficiente para cobrir os diferentes níveis de abstração que deve ser abordado no desenho de sistemas interativos. Assim, este trabalho analisa comparativamente algumas extensões propostas no contexto de um verdadeiro projeto de interface do usuário. O resultado desta análise é então empregado na proposta de um conjunto de artefatos que podem melhorar o projeto de sistemas interativos.

Palavras chaves: Modelagem da interação, UML, Extensões UML; Interação Homem-Computador.



## **ABSTRACT**

### **Comparative review of UML extensions in the artifacts of design of interactive systems**

This research examines UML extensions used in the design of user interfaces. UML is broadly used in artifacts of software development process but it does not cover all the needs of an interactive system development. Many extensions have been proposed in the literature, however, a single representation is not sufficient to cover the various levels of abstraction that must be addressed in the design of interactive systems. Thus, this work has analysed comparatively some proposed extensions in the context of a real user interface project. The result of this analysis is then used to propose a set of artifacts that can improve the project of interactive systems.

Keywords: user interface, UML, UML extension, user interface design

## Lista de ilustrações

Figura 1 – Atividades no Design Centrado no Usuário. Fonte: ISO 13407.....	12
Figura 2 – Modelo Estrela. Fonte: Hix e Hartson (2003, com adaptações).....	14
Figura 3 – Exemplo de um diagrama de layout de janelas. Fonte: Page-Jones (2001, p. 201).....	35
Figura 4 – Diagrama de navegação de janelas com rotas alternativas de navegação. ....	36
Figura 5 – Diagrama de Atividade – Estado de Seleção. ....	38
Figura 6 – Notação WISDOM. ....	42
Figura 7 – Amostra do Diagrama de Classe UML – SISGAU. ....	46
Figura 8 – Modelo Perfil do Usuário – WISDOM. ....	48
Figura 9 – Modelo de diálogo – WISDOM.....	50
Figura 10 – Diagrama de navegação de janelas – Fonte: Page-Jones (2001). ....	51
Figura 11 – Caso de Uso Avaliação Tronco – Fonte: Page Jones (2001). ....	52
Figura 12 – Diagrama de Janela – Avaliação Tronco. ....	53
Figura 13 – Interação do Fluxo de Telas – UMLi.....	54
Figura 14 – Diagrama IU.....	56
Figura 15 – Fluxo da tela – WAE.....	57
Figura 16 – Mapa de caminho de navegação – WAE.....	58
Figura 17 – Caminhos de Navegação das interfaces selecionadas para o estudo comparativo.....	59
Figura 18 – Descrição de Telas e Conteúdo – WAE.....	60
Figura 19 – Cenários do Roteiro – Fluxo de navegação determinado pela entrada de dados do usuário. ....	61
Figura 20 – Fluxo de Casos de Uso – Modelo de Interação – WISDOM.....	62
Figura 21 – Modelo de desenho ou apresentação – WISDOM.....	63

## Lista de Tabelas

Tabela 1 – Visão geral das atividades de cada área durante o processo de desenvolvimento.....	21
Tabela 2 – Artefatos x Fases para construção da camada interativa.....	22
Tabela 3 – Visões x Artefatos UML.....	27
Tabela 4 – Fases para construção da camada interativa x Artefatos UML. ....	30
Tabela 5 – Extensões levantadas. ....	32
Tabela 6 – Estereótipo UX. ....	40
Tabela 7 – Artefatos das extensões.....	43
Tabela 8 – Fases da camada interativa x Artefatos extensões UML. ....	64
Tabela 9 – Artefatos sugeridos para a modelagem de projeto de sistemas interativos. .....	71

## LISTA DE ABREVIATURAS

CIAM	Centro de Tecnologia da Informação, Automação e Mobilidade
CVEU	Ciclo de Vida de Engenharia de Usabilidade
ES	Engenharia de Software
ER	Engenharia de Requisitos
IEEE	International Electrical and Electronics Engineering
IHC	Interação Homem-Computador
IPT	Instituto Tecnológico do Estado de São Paulo
ISO/IEC	International Organization for Standardization/ International Electrotechnical Commission
IU	Interface com o Usuário
MoLIC	Modeling Language for Interaction as Conversation
OMG	Object Management Group
OMT	Object Modeling Technique
SISGAU	Sistema de Gerenciamento de Árvores Urbanas
SIMGAU	Sistema Móvel de Gerenciamento de Árvores Urbanas
SSC	Seção de Software Corporativos
UCD	User Centred Design
UML	Unified Modeling Language
UMLi	The Unified Modeling Language for Interactive Applications
WAE	Web Application Extension
WISDOM	Whitewater Interactive System Development with Object Models

## SUMÁRIO

1	INTRODUÇÃO .....	1
1.1	MOTIVAÇÃO.....	1
1.2	OBJETIVO.....	3
1.3	METODOLOGIA .....	3
1.4	ORGANIZAÇÃO DO TRABALHO.....	4
2	ARTEFATOS DO PROCESSO DE DESENVOLVIMENTO DE SISTEMAS INTERATIVOS .....	5
2.1	ENGENHARIA DE REQUISITOS E IHC .....	5
2.2	ARTEFATOS PROPOSTOS PARA O PROCESSO DE DESENVOLVIMENTO DE SISTEMAS INTERATIVOS.....	6
2.2.1	<i>Engenharia cognitiva</i> .....	7
2.2.2	<i>Engenharia de Usabilidade</i> .....	9
2.2.3	<i>Modelos do design de software</i> .....	13
2.2.4	<i>Design Participativo</i> .....	15
2.2.5	<i>Artefatos para modelagem</i> .....	16
2.3	SELEÇÃO DOS ARTEFATOS DOS AUTORES ESTUDADOS .....	18
2.4	FASES DO PROCESSO DE DESENVOLVIMENTO DE SISTEMAS INTERATIVOS .....	21
2.5	ARTEFATOS X FASES DO PROCESSO DE DESENVOLVIMENTO DE SISTEMAS INTERATIVOS.....	22
3	A UML COMO APOIO AOS ARTEFATOS DE INTERAÇÃO .....	24
3.1	MODELAGEM COM A UML .....	24
3.1.1	<i>Visões UML</i> .....	24
3.1.2	<i>Visão de caso de uso</i> .....	25
3.1.3	<i>Visão Lógica</i> .....	25
3.1.4	<i>Visão de Concorrência</i> .....	26
3.1.5	<i>Visão de Componentes</i> .....	26
3.1.6	<i>Visão de Implantação</i> .....	26
3.2	MECANISMOS DE EXTENSÃO DA UML .....	27
3.2.1	<i>Estereótipos</i> .....	27
3.2.2	<i>Valores atribuídos</i> .....	28
3.2.3	<i>Restrições</i> .....	28
3.3	UML E SISTEMAS INTERATIVOS.....	28
3.3.1	<i>Artefatos UML no Processo de Modelagem de Sistemas Interativos</i> .....	29
4	EXTENSÕES DA UML PARA SISTEMAS INTERATIVOS .....	32
4.1	DEFINIÇÃO DAS EXTENSÕES A SEREM ESTUDADAS.....	33
4.1.1	<i>Page-Jones</i> .....	34
4.1.2	<i>UMLi (The Unified Modeling Language for Interactive Applications)</i> .....	36

4.1.3	WAE ( <i>Web Application Extension</i> ).....	38
4.1.4	WISDOM ( <i>Whitewater Interactive System Development with Object Models</i> ).....	40
4.2	ARTEFATOS DAS EXTENSÕES .....	42
5	ESTUDO COMPARATIVO DAS EXTENSÕES UML APLICADAS NO PROJETO SISGAU ...	44
5.1	CRITÉRIOS PARA COMPARAÇÃO .....	46
5.2	ANÁLISE DE REQUISITOS .....	47
5.3	DESIGN .....	50
5.3.1	<i>Page-Jones</i> .....	50
5.3.2	<i>UMLi</i> .....	53
5.3.3	<i>WAE</i> .....	56
5.3.4	<i>WISDOM</i> .....	61
5.4	TESTE.....	63
5.5	ANÁLISE COMPARATIVA .....	64
5.5.1	<i>Page-Jones</i> .....	64
5.5.2	<i>UMLi</i> .....	65
5.5.3	<i>WAE</i> .....	66
5.5.4	<i>WISDOM</i> .....	67
5.6	PROPOSTA DE ARTEFATOS PARA MODELAGEM DE SISTEMAS INTERATIVOS .....	68
6	CONCLUSÕES .....	72
6.1	TRABALHOS FUTUROS .....	74
	REFERÊNCIAS.....	76

## 1 INTRODUÇÃO

Este trabalho faz um estudo das extensões UML para representação da interação homem-computador, por meio de uma análise comparativa aplicada a um projeto real de software interativo. O foco da comparação é a representatividade da extensão em cada artefato, que deve oferecer ao projetista um meio de expressar precisamente suas intenções em todo o processo de desenvolvimento.

### 1.1 Motivação

Para permanecerem competitivas, as equipes de projetos de software buscam na modelagem do sistema os diversos níveis de abstração do processo de desenvolvimento de software. O uso de modelos auxilia no entendimento e na comunicação da equipe. A elaboração de modelos elimina detalhes irrelevantes para o processo e serve como catalisador, possibilitando a criação de visões simplificadas de realidades complexas.

Como consequência do progresso na área de hardware e de software, aumentaram as opções de estilo de interação e a complexidade dos softwares projetados e diversificou-se o perfil dos usuários (BARCELOS e ALUÍSIO, 2002), tendo ocorrido também o aumento da busca por usabilidade.

Uma das principais funções dos modelos no projeto de sistemas computacionais é representar as decisões de projeto, de forma que possam ser facilmente compartilhadas, analisadas e discutidas pela equipe de projeto e de desenvolvimento. É importante observar que, enquanto utiliza modelos, o projetista está sendo induzido e conduzido por esses modelos a compreender melhor o sistema que está sendo projetado (SILVA, 2003).

Tendo em vista a preocupação com a qualidade na interação, a comunidade de desenvolvedores está adotando modelos para representar a interação. Modelos são desenhos ou imagens que representam o que se pretende reproduzir (Dicionário Michaelis, 1991). Quando se usam modelos, na verdade, está-se demonstrando e decompondo o problema em perspectivas diferentes, para poder simplificá-lo e tratar mais facilmente cada uma dessas perspectivas. Além dessa função, cada

perspectiva pode ser tratada em diferentes níveis de abstração, às quais são endereçadas diferentes questões de projeto. Em outras palavras, cada modelo trata da realidade sob uma determinada perspectiva e em determinado nível de detalhes, utilizando, portanto, as duas estratégias (SILVA, 2003).

Diante desse quadro, autores, como Fowler (2004) e Thomas (2004) vêm estendendo a UML, para facilitar a representação da interação e da interface de sistemas interativos. No entanto o projeto de sistemas interativos, como qualquer projeto de software, é uma construção intelectual que passa por diferentes níveis de abstração. A UML, útil para capturar diversas abstrações do projeto de software, pode ser adequada à representação das abstrações específicas da interação homem-computador.

Na experiência profissional da autora como analista e desenvolvedora de sistemas interativos, observou-se que a representação da interação assume diferentes aspectos, a saber:

- representação do perfil dos diversos usuários que interagem com o sistema;
- representação do diálogo entre sistema e usuário;
- representação da seqüência das telas e de sua navegação;
- representação dos símbolos; e
- representação das funções do sistema.

A consistência dessa representação e sua comunicabilidade no ambiente de projeto são aspectos importantes, não apenas para o sucesso do projeto de um sistema interativo, mas também, para a eficácia e a produtividade de uma equipe de projeto.

Contudo há a preocupação da comunidade de desenvolvimento de software em garantir que a UML seja completa para o desenvolvimento de software e que se preencham todas as lacunas. Para tanto, autores, como Page Jones (2001), Jim Conallen (2003), Silva e Paton (2003), Nunes (2000), dentre outros, utilizaram mecanismos de extensão, os quais são elementos de modelagem que permitem tratamento visual e especializado e definem como criar novas semânticas na



construção da interação do usuário com o sistema.

## **1.2 Objetivo**

O objetivo deste trabalho consiste em fazer um estudo comparativo das extensões UML para modelagem de interação das interfaces, no desenvolvimento de software interativo. O foco da comparação é a representatividade dos artefatos gerados, os quais devem oferecer ao projetista um meio de expressar precisamente suas intenções em todo o processo de desenvolvimento.

Para tanto, examina-se um conjunto de extensões propostas na literatura e se desenvolve a análise comparativa por meio de sua aplicação a um sistema real, desenvolvido e implementado no Centro Tecnologia da Informação, Automação e Mobilidade (CIAM), por meio da Seção de Sistemas Corporativos (SSC) do Instituto de Pesquisas Tecnológicas do Estado de São Paulo – IPT.

## **1.3 Metodologia**

Para o atendimento dos objetivos expostos, realizou-se uma revisão bibliográfica, a qual levantou modelos de processos fundamentados nos princípios da IHC e da representatividade dos artefatos, o metamodelo da UML e as extensões da Linguagem de Modelagem Unificada (UML) existentes, para o mapeamento de interação.

Para a realização deste trabalho, foram propostas três questões a serem respondidas:

- Quais os artefatos necessários ao projetista de IHC, durante cada fase do projeto de um sistema interativo?
- A UML possui diagramas para representar a IHC?
- As extensões UML podem satisfazer todas as fases do projeto?

## 1.4 Organização do Trabalho

Além do presente capítulo, este trabalho é composto por outros cinco, como segue:

Capítulo 2: Artefatos do processo de desenvolvimento de sistemas interativos – introduz as características da modelagem de interface em engenharia de requisitos e uma lista dos artefatos necessários ao projetista de IHC, para a construção de um sistema interativo.

Capítulo 3: A UML como apoio aos artefatos de interação – apresenta o metamodelo da UML usado para representar a interface e os elementos previstos para sua extensão.

Capítulo 4: Extensões da UML propostas para Sistemas Interativos – expõe extensões da UML para modelagem de interfaces em sistemas de informação interativos, conforme descritos por trabalhos encontrados na literatura.

Capítulo 5: Estudo comparativo das extensões UML aplicadas no projeto SISGAU - apresenta aplicação das extensões UML em um projeto da Seção de Sistemas Corporativos do IPT, de acordo com as fases do projeto. O foco da comparação é a representatividade dos artefatos gerados. Para orientar a avaliação do estudo comparativo, utilizou-se o conceito de qualidade em uso da ISO/IEC 9126-4.

Capítulo 6: Conclusões e trabalhos futuros – aborda os resultados da aplicação das extensões da UML.

## **2 ARTEFATOS DO PROCESSO DE DESENVOLVIMENTO DE SISTEMAS INTERATIVOS**

A fundamentação teórica sobre os assuntos que serão abordados nesta pesquisa serve para fornecer um embasamento para realização das análises e obtenção dos resultados.

Neste Capítulo, são descritos de forma sucinta os principais conceitos de IHC de interesse para o tema desta dissertação. Destacam-se os artefatos gerados no processo de desenvolvimento de sistemas interativos propostos na literatura de IHC, para representar a interação no projeto de um sistema interativo. As definições e o vocabulário foram obtidos em Nielsen (1993), Sommerville (2007), Shneiderman (1998), Rosson e Carroll (2002), Constantine (1999), Booch (1999), Furlan (1999), OMG (2004), Filgueiras (2004), Mayhew (1999), dentre outros.

### **2.1 Engenharia de Requisitos e IHC**

Tanto a área de Engenharia de Requisitos (ER) quanto a área de Interação Humano-Computador (IHC), duas áreas dentro da Ciência da Computação, propõem modelos de processo, métodos e técnicas para o desenvolvimento de sistemas interativos. Contudo essas áreas diferem em sua visão com relação a tal tarefa, pois, enquanto o foco principal da ER é o processo, a IHC foca os aspectos de interação do homem com a máquina. No entanto essas áreas são complementares e existe necessidade de se integrarem as duas visões no desenvolvimento de sistemas interativos, de modo que esses abordem de maneira clara e definida o projeto da interação e da implementação conjuntamente, visando ao sucesso do sistema interativo. Estudos têm mostrado que o sucesso dos sistemas interativos depende, na sua grande maioria, da sua usabilidade.

Segundo Nielsen (1993), um sistema, no qual seu manuseio seja fácil e rapidamente aprendido e dificilmente esquecido, não pode provocar erros operacionais e deve oferecer alto grau de satisfação para seus usuários, a fim de resolver eficientemente as tarefas para as quais ele foi projetado e possa, assim, ser considerado um sistema com usabilidade.

A análise da qualidade de um sistema interativo, no que diz respeito à sua usabilidade, tem, portanto, que considerar tanto os usuários do sistema como as pessoas envolvidas no processo de desenvolvimento (programadores, analistas, arquitetos e testadores).

A ER propõe essa análise da qualidade e admite que aspectos sociais e humanos desempenhem um importante papel no processo de desenvolvimento (NUSEIBEH e EASTERBROOK, 2000). Segundo o IEEE (STD830, 1984), ER corresponde ao processo de aquisição, refinamento e verificação das necessidades do cliente para um sistema de software, objetivando-se ter uma especificação completa e correta dos requisitos de software.

A ER é uma área ampla e multidisciplinar (NUSEIBEH e EASTERBROOK, 2000). A fim de facilitar a relação entre equipes multidisciplinares, modelos e notações são fundamentais para se descreverem as atividades do usuário e a estrutura do diálogo homem-máquina, os quais lhes servem de apoio. O processo de ER pode ser considerado um conjunto estruturado de atividades seguidas com o objetivo de derivar, validar e manter um documento de requisito (SOMMERVILLE, 2007). Existem várias propostas para modelos de processo de engenharia de requisitos, todavia não existe um processo considerado ideal.

Para tanto, torna-se necessário estabelecer mecanismos de comunicação entre as duas áreas, IHC e ER e uma alternativa seria alinhar os modelos de processo da ER com a IHC. Esses modelos de processo poderão ser desenvolvidos sob o ponto de vista da interação entre o sistema, seus usuários e o contexto que os rodeia, podendo ser utilizados nas fases de desenvolvimento, com a finalidade de ajudar a definir o sistema a ser implementado.

## **2.2 Artefatos propostos para o Processo de Desenvolvimento de Sistemas Interativos**

As atividades da ES, em geral, envolvem uma grande quantidade e variedade de tipos de artefatos, os quais são gerados durante o processo de desenvolvimento de software. Um artefato pode ser visto como qualquer informação gerada, alterada ou usada no decorrer do processo de desenvolvimento (JACOBSON et al., 1999). Sendo assim, um artefato pode estar na forma de um modelo, um elemento do

modelo (um gráfico, uma classe, um subsistema etc), um caso de teste, uma ata de reunião, um manual etc.

Quando os artefatos produzidos são claros, a tendência é a de que os atores do desenvolvimento de sistema se entendam. Por sua vez, um documento, ainda que esteja bem escrito, pode causar dificuldades de entendimento a quem o lê, por causa de determinada escolha ou decisão para se constituir daquela maneira. Por esse motivo, são produzidos diversos artefatos ao longo do processo de desenvolvimento de sistemas interativos, algumas vezes, com aspectos repetidos em diferentes níveis de detalhamento e com relação entre si. (BOMBANI et al., 2000).

Esta pesquisa considerou os artefatos sugeridos por diferentes níveis de trabalhos, os quais foram selecionados por vários motivos: em primeiro lugar, porque todos tratam de processos de desenvolvimento de sistemas interativos. Nesse sentido, a coleção de trabalhos captura as diversas gerações de processos de desenvolvimento de sistemas interativos, surgidas ao longo dos últimos vinte anos – desde o trabalho sobre a Engenharia Cognitiva, de Norman e Draper (1986), até a abordagem de Brown (2007) que detalha os artefatos produzidos modernamente na documentação do projeto de *websites*.

Na seção seguinte serão apresentados os paradigmas que têm fundamentado as bases do design em IHC e seu artefatos.

### **2.2.1 Engenharia cognitiva**

Norman e Draper (1986) definem o modelo de Engenharia Cognitiva, que enfatiza o entendimento e a modelagem de uma atividade na forma como ela é entendida pelo usuário. Nesse modelo, o projetista está interessado em entender o porquê de os usuários se comportarem de tal modo ou o porquê de um projeto estar melhor do que outro (BROWN, 1996).

Segundo Norman e Draper<sup>1</sup> (apud BROWN, 1996), o designer institui o seu modelo mental sobre o sistema, o qual é chamado de modelo de *design*, a partir de

---

<sup>1</sup> NORMAN, D. A.; DRAPER, S. W. **User Centered System Design**: New Perspectives on Human-Computer Interaction. Hillsdale (EUA): Lawrence Erlbaum Associate Publishers, 1986.

dois outros modelos: o de quem é o usuário e o de quais são as tarefas que ele terá de atingir. A implementação desse modelo de design é a imagem do sistema.

Considerando a realização de tarefas complexas por usuários não experientes como uma atividade de resolução de problemas, a Engenharia Cognitiva visa facilitar a realização da tarefa por meio de um bom modelo conceitual do sistema físico (ROCHA e BARANAUSKAS, 2003).

Na Engenharia cognitiva, o designer cria um modelo (mental) sobre o sistema, chamado modelo de design. Esse modelo é construído tomando como base dois outros modelos: um modelo de tarefas, representando as tarefas que os usuários realizarão utilizando o sistema; e um modelo de usuários, representando características e necessidades dos usuários. A versão do modelo de design implementada é chamada de imagem do sistema e é com ela que o usuário interage. Por meio dessas interações, o usuário constrói o seu próprio modelo mental, chamado de modelo de uso e, com base nesse modelo, o usuário planeja como realizar as tarefas utilizando o sistema e como interpretar as respostas apresentadas pelo sistema.

A Engenharia Cognitiva se concentra nos processos cognitivos ocorridos durante a interação usuário-sistema. Do ponto de vista do processo de design, ela se concentra no produto do design.

Podem-se abstrair duas atividades desse processo: o modelo de usuário – características e necessidades dos usuários e modelo de tarefas – representa as tarefas que os usuários realizarão utilizando o sistema e que, em conjunto, formam o modelo de design. O modelo de uso é um modelo mental, construído pelo usuário final a partir da interação com o sistema.

Os autores desse modelo não propõem artefatos, mas é possível sugerir artefatos, tendo em vista suas duas atividades: modelo de usuário e modelo de tarefas.

No modelo de usuário, é possível gerar cenários com descrição detalhadas dos tipos e perfis de usuários; e, para representar o modelo de tarefas, é possível produzir diagramas de tarefas.

## 2.2.2 Engenharia de Usabilidade

Nielsen (1993) apresenta um modelo de processo da Engenharia de Usabilidade composto por onze atividades, chamadas de estágios, divididos em três fases: pré-projeto, projeto e pós-projeto.

Os estágios do modelo de processo proposto por Nielsen (op. cit.) seguem as seguintes atividades:

**Pré-projeto** – tarefas são realizadas para a compreensão do usuário e de suas atividades, além das funcionalidades a serem implementadas no sistema e o contexto de trabalho. As tarefas dessa atividade são:

- Conhecendo o usuário – compreende características individuais dos usuários, análise de tarefas, análise funcional, evolução do usuário e do trabalho;
- Fazendo análise competitiva; e
- Determinando metas de usabilidade – compreende da análise financeira do impacto.

**Projeto** – nesse estágio, preocupa-se em explorar ao máximo as possíveis soluções de projeto antes da escolha de solução final para o sistema. Integram esse estágio as seguintes atividades:

- Projeto paralelo;
- Design participativo;
- Projeto coordenado da interface inteira;
- Aplicação da análise de heurísticas e *guidelines*;
- Prototipação;
- Teste empírico; e
- Projeto iterativo – compreende a captura das razões do projeto.

**Pós-projeto** – esse estágio visa obter dados sobre usabilidade a serem aplicados em próximas versões ou em produtos futuros.

- Coleta do *feedback* de estudo de campo

Segundo Nielsen (1993), uma vantagem desse modelo é a realização do trabalho de usabilidade antes do sistema ser projetado, impedindo o desenvolvimento de características desnecessárias. Outra justificativa é a diminuição de custos, pois o modo mais barato para as atividades de usabilidade influenciar em um produto é fazer tudo o que é possível antes do projeto começar (NILSEN, op. cit.). Esse autor recomenda atividades, mas não diz quais artefatos devem ser produzidos em cada atividade. A partir da leitura que propõe, é possível sugerir artefatos para três atividades:

- a) Conhecendo o Usuário – nessa atividade, é necessário apresentar as características individuais dos usuários, a análise de tarefas, a análise funcional, a evolução do usuário e a do trabalho. Tendo em vista essas premissas, é possível sugerir como artefatos cenários, personas (COOPER, 1999) e diagramas de tarefas;
- b) Prototipação – é possível sugerir protótipos; e
- c) Teste empírico – é possível sugerir plano de teste.

### **2.2.2.1 Ciclo de Vida de Engenharia de Usabilidade**

O Ciclo de Vida de Engenharia de Usabilidade (CVEU), tal como definido por Mayhew (1999), é um processo estruturado com base na Engenharia de Usabilidade e considera o design da IU como um produto dividido em três etapas: análise de requisitos, projeto/desenvolvimento/teste e instalação.

Análise de requisitos – compreende obtenção do perfil do usuário, análise contextual da tarefa, restrições e potencialidades da plataforma, princípios gerais do projeto e estabelecimento de metas de usabilidade. Nessa fase, o maior interesse é a análise contextual da tarefa, incluídos o registro visual e também os princípios de projeto.

Projeto/desenvolvimento/teste – etapa dividida em três fases: projeto de alto nível da interface, normas de projeto da interface e projeto detalhado.

Instalação – é a instalação final da IHC, de onde se obtém o *feedback* do usuário para a identificação de problemas e validação do projeto. Nessa etapa, é de especial interesse identificar os problemas que podem ocorrer associados às variáveis do ambiente de trabalho do usuário.



No CVEU, ocorre uma grande atenção desta autora para os artefatos do projeto. Cada etapa descrita apresenta uma coleção de artefatos que expressam entradas e saídas dessa. Os artefatos principais dessas etapas apresentam bom nível de detalhe.

Concentrando-se nos artefatos principais e ignorando aqueles intermediários de cada atividade, podem-se destacar os seguintes:

- Análise de requisitos – questionário com o usuário, cenários e modelo de tarefas;
- Projeto/desenvolvimento/teste – protótipos e planos de teste; e
- Instalação – manual de instalação e manual do usuário.

Constata-se que, na maioria dos processos, os artefatos gerados dependem da solicitação do cliente ou da negociação entre cliente e desenvolvedor, os quais produzem os artefatos que lhes são convenientes, sem que haja padronização.

### **2.2.2.2 Projeto centrado no usuário**

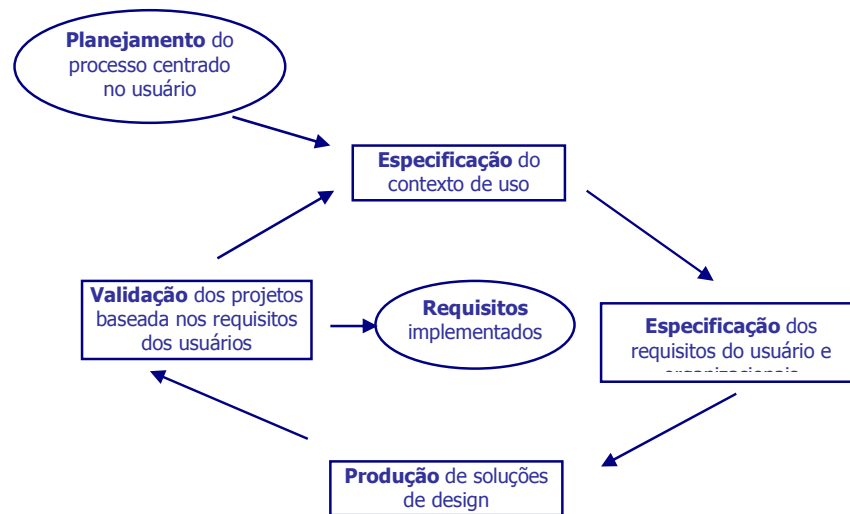
Em 1999, a ISO publicou uma norma, a ISO 13407 – Human-Centred Design Processes for Interactive Systems – que define o modelo de processo centrado no usuário. Os princípios desse processo haviam sido estabelecidos por Gould e Lewis (1985). Esse modelo destaca a necessidade de desenvolver softwares mais usáveis, focando a expectativa do usuário no processo de desenvolvimento, por meio de uma visão geral para planejamento e gerenciamento de projetos centrados no usuário.

Algumas regras são descritas pela norma ISO 13407, para auxiliar na incorporação das perspectivas do usuário no processo:

- alocação apropriada de funções entre o usuário e o sistema;
- envolvimento participativo do usuário;
- iteração de soluções de projeto; e
- equipe de projeto multidisciplinar.

A norma ISO 13407 define também quatro atividades essenciais para o

Design Centrado no Usuário (UCD), cujas atividades são apresentadas na Figura 1.



**Figura 1** – Atividades no Design Centrado no Usuário. Fonte: ISO 13407.

a) Especificação do contexto de uso – essa atividade visa capturar informações sobre características dos usuários, tarefas que eles realizam e o ambiente em que os usuários utilizarão o software. Essas informações, além de subsidiarem o projeto, fornecerão uma base para as atividades de avaliações posteriores.

b) Especificação dos requisitos do usuário e organizacionais – nesse caso, são especificados os requisitos do usuário e da organização, além de uma classificação dos requisitos por ordem de prioridade.

c) Produção de soluções de design – nessa atividade, são exploradas possíveis soluções de projeto, descritas por meio da utilização de protótipos. As primeiras soluções de projeto podem ser baseadas em experiências anteriores.

d) Validação dos projetos baseada nos requisitos dos usuários – é a última atividade da iteração e tem como objetivo confirmar o nível em que os objetivos da organização e dos usuários foram alcançados, fornecendo também informações para o refinamento do projeto.

Esses objetivos são alcançados por meio da realização e da análise de testes com usuários ou com especialistas em testes de usabilidade, documentando os resultados obtidos e realizando as mudanças necessárias.

A atividade de especificação do contexto de uso a ISO 13407 não sugere artefatos, porém, para capturar as informações sobre as características de usuários, tarefas e ambiente, é possível sugerir como artefatos: cenários ou especificação de casos de uso e modelo de tarefas.

Na atividade de produção de soluções de design, a própria ISO 13407 sugere como artefatos protótipos. E, para a atividade de validação dos projetos baseada em requisitos dos usuários a ISO 13407, sugere análise de testes que pode ser realizada por meio de um plano de teste e de testes de usabilidade.

### **2.2.3 Modelos do design de software**

O processo de design na Engenharia de Software, de acordo com Rocha e Baranauskas (2003) parte de três pressupostos básicos: o resultado do design é um produto, o produto é derivado de especificações fornecidas pelo cliente, e uma vez que o cliente e o designer concordaram com as especificações, há pouca necessidade de contato entre eles até a entrega do produto.

Na literatura há vários modelos para o processo de design, o modelo cascata (Boehm, 1995) caracteriza bem a visão tradicional da Engenharia de software, mas não é possível entender e expressar os requisitos do usuário antes que algum design tenha sido feito. Em resposta aos problemas deste modelo, Bohem(1995) propõe o modelo espiral, o qual mostra que várias interações são necessárias e introduz a idéia de prototipagem para maior entendimento dos requisitos.

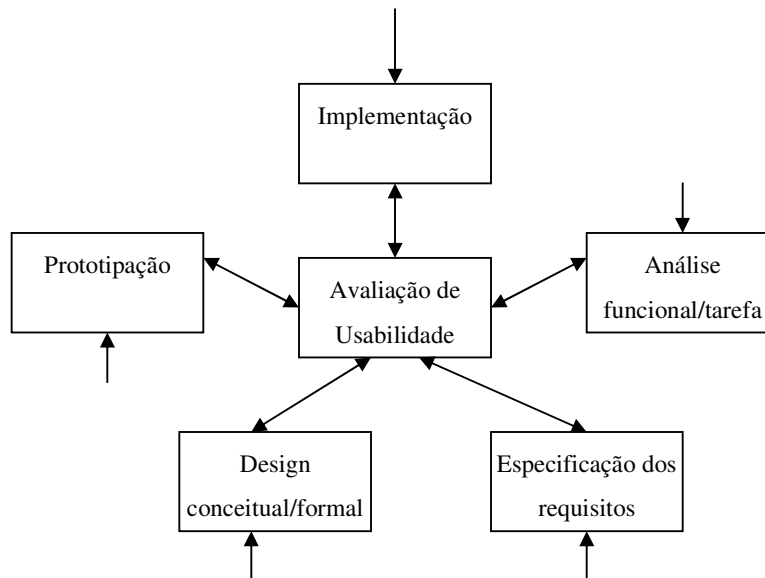
Neste trabalho será apresentado o modelo estrela, por ser bastante popular entre a comunidade IHC.

#### **2.2.3.1 O modelo estrela**

Hix e Hartson (1993) propuseram o modelo Estrela para o desenvolvimento de interação com o usuário. O modelo possui esse nome devido ao formato de sua representação, pois é centrado na etapa de Avaliação e as outras atividades estão dispostas ao redor da avaliação de usabilidade, conforme constam na Figura 2.

A avaliação estende os conceitos de validação, verificação e teste. Nesse caso, não há necessidade de se especificarem todos os requisitos antes de começar a trabalhar no projeto do sistema. Pode-se iniciar com um protótipo rápido das telas

do sistema e ir posteriormente incrementando-o com novos requisitos descobertos. De acordo com esse modelo, o ciclo de vida de desenvolvimento pode ser iniciado em qualquer fase, como indicado pelas setas de entrada na Figura 1. Cada fase pode, então, ser seguida por qualquer outra, conforme as duplas setas.



**Figura 2** – Modelo Estrela. Fonte: Hix e Hartson (2003, com adaptações).

Esse modelo tem como ponto central a participação do usuário na forma de avaliações feitas ao término de cada etapa ou na passagem de uma fase para a seguinte.

Hix e Hartson (op. cit.) citam vantagens de seu modelo:

- ocorrem *loops* da iteração mais curtos e em menor quantidade do que na metodologia espiral, proposta pela Engenharia de Software.
- ocorre a minimização do número de restrições dispostas entre as atividades de desenvolvimento, como o fato de não conhecer todos os requisitos antes de realizar o projeto.

Esse modelo de processo define seis etapas: prototipação, implementação, avaliação de usabilidade, análise funcional e análise de tarefas, especificação dos requisitos, design conceitual; e ou design formal.

Os autores não sugerem artefatos destinados ao processo do modelo estrela, desse modo e analisando as atividades, é possível sugerir protótipo que demonstre atividade de prototipação, planos de teste para a atividade avaliação de usabilidade,

diagrama de tarefa para a atividade de análise de tarefa e, quanto à atividade de especificação dos requisitos, esses podem ser gerados em cenários ou em caso de usos detalhados.

#### 2.2.4 Design Participativo

Originário da Escandinávia, do início da década de 70 (MÜLLER et al, 1997; EHN, 1992), esse modelo de processo tem como principal característica incorporar o usuário final como um membro do time de projeto, permitindo-lhe participar de todas as atividades de desenvolvimento, não somente da etapa de testes e avaliação.

Originalmente o DP visa promover a cooperação entre usuários e designers na definição de novas tecnologias para o contexto de trabalho. As atividades realizadas no DP buscam uma compreensão comum da tecnologia e do contexto social de uso explorando novas estruturas, formulando os requisitos do sistema e construindo protótipos, colocando, assim, a tecnologia como um mecanismo de aprimoramento do contexto social em uso (GRØNBÆK et al., 1997).

Para Melo e Baranauskas (2006), o Design Participativo (DP) provê um conjunto de técnicas para apoiar diferentes fases do processo de design como identificação e clarificação do problema, requisitos e análise, design de alto nível, design detalhado, avaliação, customização pelo usuário e *redesign*.

Ainda segundo esses autores, no DP, um produto não é apenas desenhado para o usuário, mas também, com ele, colaborativamente. O engajamento do usuário é considerado valioso para alcançar a qualidade do produto na medida em que possibilita um melhor entendimento do seu contexto de uso e das atividades que o usuário realiza, mediante a combinação de diferentes experiências (MÜLLER et al., 1997,<sup>2</sup> apud MELO e BARANAUSKAS, 2006).

O DP em geral acontece no contexto do usuário, incorporando-o experimentos, mas como membro da equipe de design. Três características específicas resumem o DP: ele é orientado ao contexto, envolve colaboração em

---

<sup>2</sup> Müller, M. J., Haslwanter, J. H., Dayton, T. Participatory Practices in the Software Lifecycle. In: HELANDER, M. G., LANDAUER; T. K., PRABHU; P. V. (eds.). **Handbook of Human-Computer Interaction**. 2nd edition. Amsterdam: Elsevier, 1997, p. 255-297.

vários níveis e apresenta uma abordagem iterativa ao design (SCHULER e NAMIOKA, 1993).

O DP auxilia na identificação e na validação de requisitos e na compreensão da tarefa do usuário, bem como, na aprovação de artefatos durante o processo de desenvolvimento de sistemas interativos. Entretanto, os artefatos do DP não estão explícitos, contudo no trabalho desenvolvido por de Melo e Baranauskas (2003), protótipos foram produzidos.

### **2.2.5 Artefatos para modelagem**

Duas propostas para modelagem de sistemas interativos foram selecionadas a MoLIC, proposta por Paula (2003) que trata do planejamento da interação dos usuários com o sistema (SILVA, 2007) e documentos para modelagem de *websites* (*Design Documents*) de Brown (2007).

#### **2.2.5.1 Modeling Language for Interaction as Conversation - MoLIC**

Diferentemente dos trabalhos apresentados até aqui, o trabalho de Paula (2003) trata especificamente de um artefato. Essa autora propôs a MoLIC (Modeling Language for Interaction as Conversation) idealizada para modelar a interação dos usuários com os sistemas computacionais, seguindo os princípios da IHC.

Segundo Silva (op. cit.), a MoLIC foi proposta como ponte entre representações da atividade de análise e a definição da interface. E, segundo esse autor, a MoLIC é composta por quatro elementos inter-relacionados, a saber:

- Ontologia de Signos (tabela de signos) – define e organiza todos os conceitos envolvidos com o sistema, incluindo os artefatos ou as informações envolvidas em cada ação realizada pelo usuário ao sistema. A tabela de signos proposta na MoLIC, lista os signos usados e os seus atributos, durante a interação;
- Especificação textual – descrição textual detalhada dos cenários;
- Diagrama de metas – tem como objetivo organizar hierarquicamente as metas do usuário, indicando o que ele pode realizar com o apoio do sistema; e

- Diagrama de Interação – define como as metas do usuário podem ser alcançadas durante sua conversa com o preposto do desenvolvedor para, então, ser detalhado na especificação textual.

A expectativa dos autores é a de que o desenvolvedor utilize a MoLIC e as representações empregadas na atividade de análise (por exemplo, cenários e modelo de tarefas) sem ainda pensar em detalhes de interface, como, os *widgets*, os quais compõem a interface ou comportamentos de baixo nível (clique, arrastar, navegar, digitar etc).

Depois de definida a interação, o desenvolvedor continua seu trabalho, modelando uma interface que permita ao usuário e ao desenvolvedor dialogarem conforme o projetado e, neste momento, os detalhes de interface deverão ser considerados.

A MoLIC propõe os seguintes artefatos: cenários, tabela de signos, especificação textual, diagrama de metas e diagrama de interação.

### **2.2.5.2 Os artefatos para websites**

Assim como Paula (2003), Brown (2007) não propõe um modelo de processo, mas sim, diversos documentos para modelagem (*Design Documents*): *wireframes* de interface, mapas do sistema, fluxos de interação e diagramas de navegação e documentos menos comuns, tais como: perfis de personas, análises competitivas, modelos conceituais, inventário de conteúdo e outros.

Brown (op. cit.) discute a importância da documentação e afirma que um documento captura uma ideia. Se não houver essa captura de ideias, não haverá como desenvolver sistemas que atendam às necessidades dos usuários. A equipe participante do desenvolvimento do sistema expressa suas ideias, necessidades e expectativas por meio de artefatos. Esse autor propõe quatro artefatos principais, os quais apresentam por diferentes ângulos o sistema a ser desenvolvido, os quais são: mapa do sistema, fluxos de interação, *wireframes* e o desenho das telas.

## 2.3 Seleção dos artefatos dos autores estudados

A partir de uma leitura dos autores foi possível extrair os artefatos para o desenvolvimento de sistemas interativos

a) Norman e Draper (1986)

- cenários e
- modelos ou diagramas de tarefas

b) Nielsen (1993)

- modelo ou diagrama de tarefas,
- cenários,
- protótipos
- plano de teste

c) Hix e Hartson (1993)

- protótipo
- diagrama de tarefa,
- plano de teste,
- cenários ou casos de uso detalhados

d) ISO 13407 (1999)

- cenários ou especificação de casos de uso
- modelo de tarefas
- protótipos
- plano de teste
- teste de usabilidade

e) Mayhew (1999)

- questionário com o usuário,
- cenários,



- modelo de tarefas,
- protótipo,
- plano de teste,
- manual de instalação
- Manual do usuário

f) Paula (2003)

- cenários,
- tabela de signos,
- especificação textual,
- diagrama de metas e
- diagrama de interação

g) Brown (2007)

- mapa do sistema,
- fluxo de interação,
- *wireframes* e
- desenho das telas.

Observando-se atentamente, é possível verificar que muitos artefatos se repetem e, se forem eliminado os artefatos repetidos, é possível extrair a seguinte listagem:

a) perfil dos usuários – é o documento que descreve o tipo de usuários, perfil e atividades por eles desempenhadas;

b) cenários – são composições realistas da tarefa do usuário, relacionadas com o objetivo da avaliação, as quais simulam uma tarefa real, em que o usuário interage com o sistema e onde é registrado cada procedimento tomado;

c) personas – são perfis de usuários fictícios, mas representativos;

- d) especificação de casos de uso – são narrativas de texto do caso de uso. Caso de uso são descrições de interações típicas entre os usuários e o sistema. Eles representam a interface externa do sistema e especificam um conjunto de exigências do que o sistema deve fazer;
- e) protótipo – tem a função de demonstrar os requisitos de um sistema;
- f) plano de testes – consiste na definição das metas e dos objetivos dos testes no escopo da iteração (ou projeto), os itens-alvo, a abordagem adotada, os recursos necessários e os produtos que serão liberados;
- g) diagrama de tarefas – descreve detalhes de como o processo é implementado, bem como, as instruções e ordens para a sua execução;
- h) questionário com o usuário;
- i) manual de instalação – contém instruções para instalação do sistema;
- j) manual do usuário – contém instruções de como o sistema pode ser operado pelo usuário;
- k) tabela de signos – descreve os signos integrantes da conversa usuário-preposto do designer;
- l) diagrama de metas – especifica as metas que os usuários terão ao utilizarem o sistema e a relação entre essas;
- m) diagrama de interação – representa uma visão global das possíveis conversas que podem ocorrer entre o usuário e o preposto do designer para alcançarem as metas representadas no diagrama de metas;
- n) mapa do sistema (*site maps*) – organiza e estrutura o conteúdo do sistema e seus acessos por meio de um mapa, ampliando sua visualização e proporcionando modificações em sua navegação;
- o) fluxo de interação (*flow charts*) – trata tanto do fluxo da interação das telas do sistema, quanto da interação entre o sistema e o usuário; e
- p) *wireframes* – desenvolve esboços das telas do sistema, com pré-disposições e posicionamento de conteúdos e funcionalidades.

## 2.4 Fases do processo de desenvolvimento de sistemas interativos

O processo de desenvolvimento de sistemas interativos envolve basicamente as fases de levantamento de requisitos, análise, projetos, teste, implementação e implantação (PAULA, 2007). O processo conta com a participação de profissionais de várias áreas e cada um possui seu foco e objetivo. Dentre essas áreas, é possível citar a de IHC (PREECE et al., 1994) e a Engenharia de Software (ES), Pressman (2004) e Sommerville (2007). A Engenharia de Requisitos, neste trabalho, está no contexto da Engenharia de Software.

Mapeando as atividades dessas áreas nas fases do processo de desenvolvimento, tal como consta na Tabela 1, pode-se perceber a existência de mais de uma área atuando em uma mesma fase do processo de desenvolvimento. É importante destacar que existem outras áreas, fases e atividades envolvidas no processo de desenvolvimento de um sistema interativo. Neste trabalho, é oferecida apenas uma visão geral e simplificada dessas fases, para indicar quais artefatos são necessários para cobrir cada fase do processo de desenvolvimento de sistemas interativos.

**Tabela 1** – Visão geral das atividades de cada área durante o processo de desenvolvimento.

Fase/ Área	Levantamento	Análise	Design	Implementação	Teste	Implantação
IHC (PREECE E et al., 1994).	Levantamento das necessidades e características dos usuários e seu ambiente de trabalho	Análise e modelagem dos usuários, suas tarefas e ambiente de trabalho	Projeto da IHC e da interface com o usuário e avaliação formativa		Teste de usabilidade	Teste de usabilidade
ES Pressman, (2004); Sommerville (2007)	Expectativas e necessidades dos <i>stakeholders</i> com relação ao software a ser desenvolvido	Distribui os requisitos em categorias, explora as relações entre eles, e classifica a importância de cada um dos requisitos de acordo com as necessidades dos <i>stakeholders</i> .	Projeto das funcionalidades internas do software, entradas e saídas e sua arquitetura	Construção do sistema interativo.	Teste funcional	Instalação do sistema interativo.

Considerando a visão geral das atividades de cada área durante o processo de desenvolvimento, tal como apontado na Tabela 1, é possível estabelecer três fases principais: Análise de requisitos, Design e Testes.

Nessas fases, o levantamento e a análise chamariam de Análise de Requisitos – nesta fase, o projetista (engenheiro de requisitos) especifica as funções e desempenho do software, estabelece as restrições de projeto do software, estuda as características que o sistema deverá ter para atender às necessidades e expectativas do cliente (PRESSMAN, 2004); a implementação se agregaria à fase de Design – tendo em vista que a fase de implementação tem como objetivo codificar programas, implementando o design; e a fase de Implantação ficaria agregada àquela de Testes - os artefatos gerados na fase de implantação são comparados com aqueles elaborados nas fases de análise de requisitos, a fim de verificar se os requisitos foram mantidos durante a descrição do sistema pelos outros artefatos propostos para essa descrição, estes artefatos são utilizados para a execução dos testes.

## 2.5 Artefatos x Fases do processo de desenvolvimento de sistemas interativos

Feito o levantamento dos artefatos e o levantamento das fases do processo de desenvolvimento de sistemas interativos, é possível identificar e classificar quais artefatos são possíveis para garantir as fases. E esses artefatos estão descritos na Tabela 2, a seguir.

**Tabela 2** – Artefatos x Fases para construção da camada interativa.

Fases	Artefatos
<b>Análise de Requisitos</b>	solicitação dos envolvidos; perfis dos usuários; personas; cenários; especificação de casos de uso; modelos de tarefas; pesquisa de campo; tabela de signos; diagrama de metas. protótipo ou desenho das telas;

<b>Design</b>	mapa do sistema; fluxo de interação ou diagrama de interação; <i>wireframes</i> .
<b>Testes</b>	questionário com o usuário; documentação de testes; manual de instalação; manual do usuário.

### **3 A UML COMO APOIO AOS ARTEFATOS DE INTERAÇÃO**

Neste capítulo, será apresentado como a UML – *Unified Modeling Language* – dá suporte ao processo de desenvolvimento de sistemas interativos e como ela pode contribuir na representação dos artefatos de interação.

#### **3.1 Modelagem com a UML**

A idéia central da UML é a utilização de uma só linguagem para o desenvolvimento de sistemas, de tal forma que analistas desenvolvendo diferentes sistemas poderiam compreender rapidamente o projeto uns dos outros.

Ao modelar a interação, o engenheiro de software deveria poder representar desde o diálogo até os objetos da interface em diagramas UML, facilitando, assim, a interpretação por terceiros.

A UML propõe visões, como formas de observar um mesmo problema de diferentes pontos de vista, com a finalidade de melhor entendê-lo, para então, atribuir-lhe a solução mais adequada (LARMAN, 2004). Os diferentes graus de abstração em que se deve pensar a interação devem ser mapeados nas visões da UML.

Ainda, como a UML não é suficiente para representar os elementos da interação (FOWLER, 2004; THOMAS, 2004), usa-se a possibilidade de criar extensões à linguagem. As extensões baseiam-se em mecanismos definidos pela própria UML para aumentar seu grau de representatividade.

Nas próximas seções, descrevem-se esses aspectos da UML e os diferentes graus de abstração com que se deve pensar a interação e os objetos ou elementos que povoam estes aspectos, necessários ao entendimento do trabalho.

##### **3.1.1 Visões UML**

Um sistema é composto por diversos aspectos: funcional, não funcional, requisitos de tempo, confiabilidade, desenvolvimento, aspectos organizacionais, organização do trabalho, mapeamento dos módulos de código e outros.

Um sistema é descrito em certo número de visões, cada uma representando uma projeção da descrição completa e mostrando aspectos particulares do sistema. Na UML, a arquitetura de um sistema pode ser descrita por meio de cinco visões interligadas. Cada visão é descrita por um número de diagramas (Tabela 3), que contêm informações que dão ênfase aos aspectos particulares do sistema. Existe em alguns casos certa sobreposição entre os diagramas, o que significa que um deles pode fazer parte de mais de uma visão. Os diagramas que compõem as visões contêm os modelos de elementos do sistema e as visões que compõem um sistema, tal como recomendadas pela UML (OMG, 2004), são:

### **3.1.2 Visão de caso de uso**

Descreve a funcionalidade do sistema desempenhada pelos atores externos ao sistema (usuários). A visão caso de uso é central, já que seu conteúdo é base para o desenvolvimento das outras visões do sistema. Essa visão mostra o comportamento de um sistema devendo ser invisível a todos os envolvidos: gerentes, analistas, programadores e usuários. Ela é montada sobre os diagramas de caso de uso e eventualmente diagramas de atividades (Booch, Jacobson e Rumbaugh, 1999).

### **3.1.3 Visão Lógica**

Descreve como a funcionalidade do sistema será implementada. É feita, sobretudo, por analistas e projetistas. Em contraste com a visão de caso de uso, a visão lógica observa e estuda o sistema internamente. Ela descreve e especifica a estrutura estática do sistema (classes, objetos e relacionamentos) e as colaborações dinâmicas quando os objetos enviarem mensagens uns para os outros para realizarem as funções do sistema.

Propriedades, como persistência e concorrência, são definidas nessa fase, bem como as interfaces e as estruturas de classes. A estrutura estática é descrita pelos diagramas de classes e objetos. O modelo dinâmico é descrito pelos diagramas de seqüência, colaboração e atividade (Booch, Jacobson e Rumbaugh, op. cit.).

### **3.1.4 Visão de Concorrência**

Trata a divisão do sistema em processos e processadores. Esse aspecto, que é uma propriedade não funcional do sistema, permite uma melhor utilização do ambiente onde o sistema se encontrará, verificando as execuções paralelas e a gestão de eventos assíncronos. Uma vez dividido o sistema em linhas de execução de processos concorrentes (*threads*), a visão de concorrência deverá mostrar como ocorrem a comunicação e a concorrência desse processo, já que focaliza as questões de desempenho e escalabilidade do sistema. Essa visão é suportada pelos modelos dinâmicos, que são os diagramas de estado, seqüência, colaboração, atividade e pelos diagramas de implementação, que são os diagramas de componentes e execução (BOOCH, JACOBSON e RUMBAUGH, op. cit.).

### **3.1.5 Visão de Componentes**

É uma descrição da implementação dos módulos e suas dependências. É executada, sobretudo, por projetistas e consiste nos componentes dos diagramas. Ela focaliza artefatos físicos (programas, bibliotecas, banco de dados) para o efetivo desenvolvimento do sistema (BOOCH, JACOBSON e RUMBAUGH, op. cit.).

### **3.1.6 Visão de Implantação**

Finalmente, a visão de implantação mostra a organização física do sistema, os computadores, os periféricos e como eles se conectam. Essa visão será executada pelos projetistas, integradores e testadores e focaliza topologia do hardware, liberação e instalação do sistema (BOOCH, JACOBSON e RUMBAUGH, op. cit.).



**Tabela 3 – Visões x Artefatos UML.**

<b>Visões</b>	<b>Artefatos UML</b>
Visão de Caso de Uso	descrição de caso de uso, diagrama de caso de uso, eventualmente com diagrama de atividades.
Visão Lógica	diagramas de classe, objeto, seqüência, colaboração e atividade.
Visão de Concorrência	diagramas de estado, seqüência, colaboração, atividade e implementação.
Visão de Componentes	diagrama de componentes.
Visão de Implantação	diagrama de implantação.

### **3.2 Mecanismos de Extensão da UML**

A UML pode ser estendida ou adaptada para um método específico, uma organização ou um usuário. De acordo com a OMG (2004), os mecanismos são elementos de modelagem que permitem o tratamento visual especializado e definem como criar novas semânticas.

A UML dispõe de um conjunto de elementos e conceitos de modelagem, porém seus usuários podem precisar de notações adicionais além daquelas definidas pela UML. Ela possui três mecanismos de extensão: estereótipos, valores atribuídos e restrições, os quais podem ser usados em conjunto ou separadamente para representar novos elementos de modelagem que possuem semântica, características e notação própria.

#### **3.2.1 Estereótipos**

De acordo com Booch, Jacobson e Rumbaugh (1999), os estereótipos ampliam o vocabulário da UML e permitem a criação de novos tipos de blocos de construção, derivados de outros já específicos a determinado problema. Além disso, definem um novo elemento de modelo, cuja origem está em elementos existentes na UML. Por esse motivo, a semântica de um estereótipo é a extensão daquela definida para os elementos existentes.

E não há restrições quanto ao uso do estereótipo de um elemento, o qual poderá ser usado em uma mesma situação com o elemento original.

Os estereótipos são usados com diversos elementos, tais como: classes, componentes e notas, bem como, com relacionamentos, que podem ser associações, generalizações e dependências.

Conforme a OMG (2004), o estereótipo é um dos mais importantes mecanismos de extensão, pois permite que sejam criados novos elementos de modelagem, facilitando, assim, a inclusão de metaclasses com novos metatributos e semântica.

Ao aplicar um estereótipo a um elemento, como um nó ou uma classe, ocorre a extensão da UML, mediante a criação de um novo bloco de construção semelhante a algum já existente com suas propriedades especiais específicas. Cada estereótipo pode fornecer seu próprio conjunto de valores atribuídos e suas semânticas. Cada estereótipo poderá fornecer seu próprio ícone. Enfim, com o uso de estereótipos, é possível acrescentar itens novos à UML.

### **3.2.2 Valores atribuídos**

*São propriedades de elementos explicitamente definidos e representam informações sobre o modelo ou elementos de modelo, e não, do sistema que está sendo modelado. Com o uso de valores atribuídos, é possível acrescentar novas propriedades à UML.*

### **3.2.3 Restrições**

Uma restrição consiste na especificação de semântica associada a um conjunto ordenado de elementos do modelo. Tal especificação é escrita em determinada linguagem de restrições, que pode ser mais ou menos formal. A partir dessas restrições, pode-se adicionar uma nova semântica ou modificar regras existentes.

## **3.3 UML e Sistemas Interativos**

Tal como foi citado na Introdução deste trabalho, têm surgido diversas

propostas de extensão da UML para melhor modelar o processo de desenvolvimento de sistemas interativos.

Há propostas que apóiam representações para a modelagem dos sistemas interativos em UML por meio das extensões, dentre as quais, podem-se citar: Page-Jones (2001), Silva e Paton (2000), Conallen (2003).

A adoção dessas propostas por parte da comunidade tem se mostrado lenta e esse fato deve-se, por um lado, à natural resistência em aprender novas linguagens de modelagem. Por outro lado, deve-se à resistência em apreender novas representações gráficas, como é o caso das extensões da notação UML.

Como a UML é utilizada e suportada por uma vasta comunidade de usuários, torna-se, assim, relevante explorar qual é a melhor forma de retirar o máximo benefício fornecido pela linguagem para a resolução dos problemas enfrentados por um engenheiro de software, quando confrontado com a necessidade de desenvolver um sistema interativo.

### **3.3.1 Artefatos UML no Processo de Modelagem de Sistemas Interativos**

Segundo Booch, Rumbaugh, e Jacobson (1999), no processo de desenvolvimento de um software, é benéfica a produção dos seguintes artefatos: requisitos, arquitetura, projeto, código-fonte, planos de projetos, planos de testes, protótipos e versões. Dependendo da cultura de desenvolvimento, alguns desses artefatos são tratados de maneira mais ou menos formal do que outros.

Booch, Rumbaugh, e Jacobson, (1999, p. 17) descrevem:

*a UML abrange a documentação da arquitetura do sistema e de todos os seus detalhes. A UML proporciona também uma linguagem para a expressão de requisitos para a realização de testes. Por fim, a UML oferece uma linguagem para a modelagem das atividades de planejamento do projeto e de gerenciamento de versões.*

A partir do estudo dos modelos de processos realizados no Capítulo anterior, foram selecionadas três fases que contribuem para a construção da camada interativa no processo de modelagem e desenvolvimento de interface: análise de requisitos, design e testes.

a) Análise de requisitos – utiliza-se de descrições de Caso de Uso para capturar atores e papéis que os usuários de casos de uso desempenham, quando interagem com esses casos de uso. Utilizam-se também os diagramas de Caso de Uso para capturar os requisitos expressos pelos usuários do sistema. Esse é o diagrama mais adequado para a fase, até pela definição de Caso de Uso, que se refere a uma descrição informal dos requisitos funcionais, dos atores envolvidos e dos resultados produzidos (OMG, 2004). A partir dessas informações, é possível extrair dados necessários para especificação das tarefas.

Depois do levantamento feito pelas descrições de Caso de Uso e pelos diagramas de caso de uso, é possível fazer a modelagem dos diagramas de classe e de seqüência, no qual é possível apresentar as informações sobre métodos, atributos, nome das funções e como serão integradas e também expressar os objetos.

b) **Design** – nessa fase, por meio da informação existente nos Diagramas de Caso de Uso e de Atividade, identificam-se as interfaces necessárias para aplicação e, em consequência das descrições existentes nos Diagramas de Atividade, desenvolvem-se Diagramas de Estados.

c) **Testes** – muitos tipos de testes podem ser feitos em um sistema interativo: teste de usabilidade, testes unitários, teste funcional e teste de integração, dentre outros.

Embora a UML não se refira a elaboração de teste, é possível fazer testes unitários por meio dos diagramas de classe (*Unit test*), além de teste integrado com os diagramas de componentes (*component test*). A Tabela 4, a seguir, apresenta os artefatos UML para cada fase do projeto.

**Tabela 4** – Fases para construção da camada interativa x Artefatos UML.

Fases para construção da camada interativa	Artefatos UML
Análise de Requisitos	<p>Diagramas de caso de uso para capturar os requisitos expressos pelos usuários do sistema.</p> <p>Informações obtidas nos casos de uso para extrair a informação necessária, destinada à especificação das tarefas e demonstrada por meio dos Diagramas de Atividades.</p>
Design	<p>Diagramas de classe e de seqüência para expressar os objetos, seus atributos e o fluxo da interação.</p> <p>Diagramas de atividades, diagramas de estado de transição e diagramas de pacotes, contendo componentes de design.</p>
Testes	Diagramas de classe e de componentes. Casos de uso para orientar os testes de usabilidade.

#### 4 EXTENSÕES DA UML PARA SISTEMAS INTERATIVOS

Alguns autores, preocupados em garantir que a UML seja completa para o desenvolvimento de software e preencha todas as lacunas de representação necessárias nesse processo, utilizaram mecanismos de extensão para criar novas semânticas necessárias para representar a interação do usuário com o sistema, a análise de requisitos, a construção de interfaces e a interação entre objetos que compõem o sistema.

Para representar essa preocupação, foi feito um levantamento de extensões da UML, cujo resultado é apresentado a seguir, na Tabela 5.

**Tabela 5** – Extensões levantadas.

Extensão	Ano	Foco	Referência	Autores	Origem
Diagramas de Layout e de Navegação de Janelas.	1996	<b>Prototipação e navegação de tela.</b>	PAGE-JONES (2001)	Page-Jones	New York.
UMLi	1998	Modelagem de interface e modelagem do comportamento UI.	SILVA e PATON (2000)	Paton e Silva	Universidade de Manchester.
WAE	1998	Internet	CONALLEN (2003)	Jim Conallen	Rational Software Corporation.
WISDOM	2000	Modelagem de Sistemas Interativos – prototipação evolutiva.	NUNES (2000)	Nuno Jardim Nunes	Universidade de Madeira – Portugal.
AUML	2000	Modelagem de sistemas envolvendo objetos e Modelagem de comunicação entre agentes (objetos).	PARUNAK e BAUER (2000)	Odell, Parunak e Bauer	FIPA – Foundation for Intelligent Physical Agents.
Libra	2004	Comportamento de elementos: objetos e aspectos.	CHAVES (2004)	Chaves	Universidade Federal de Santa Catarina.
IMML	2003	Modelagem visual de interfaces de usuário. Descrição abstrata de IU.	LEITE (2003)	Jair Cavalcante Leite	Universidade Federal do Rio Grande do Norte
TUML	1997	Modelos temporais.	SVINTERIKOU	Svinterikou	UMIST – U.K

			e THEODOULIDI S (1997)		
UMP <sup>2</sup> D	2000	Desenvolvimento de aplicações paralelas.	OLIVETE e TRINDADE (2000)	André Luís Olivete, Dr. Onofre Trindade Júnior	ICMC – Instituto de Ciências Matemáticas e Computação da Universidade de São Paulo.
Xchart	1997	Modelo de Especificação e implementação de gerenciadores de diálogo.	LUCENA (1997)	Fábio Nogueira de Lucena	Instituto de Computação da Unicamp – SP.
UWE	2000	Extensão para modelar Sistema Hiperídia Adaptativa (SHAs).	KOCH et al. (2000)	Nora Parcus de Koch	Munique – Alemanha.
UML-MC	2002	Extensão da UML para representar Mapas Conceituais.	Robinson e Crespo (2002)	Robinson e Crespo	Universidade do Vale do Rio dos Sinos – RS.

#### 4.1 Definição das extensões a serem estudadas

Com o estudo das extensões, foi possível constatar que nem todas elas colaboram para a modelagem de sistemas interativos. Pôde-se perceber que:

- Agente UML (AUML) – foi projetado para modelar protocolos de agentes;
- Libra – provê um conjunto de extensões orientadas a aspectos e possibilita a geração automática de sistemas executáveis a partir de especificações com alto grau de abstração e modularidade;
- Interactive Message Modeling Language (IMML) – está ainda em fase de desenvolvimento e sua sintaxe acompanha uma semântica informal, descrita em linguagem natural;
- Linguagem Unificada de Modelagem Temporal (TUML) – fornece o conjunto de construtores para a modelagem efetiva de informações temporais, as quais variam no decorrer do tempo.
- Unified Methodology for Parallel Programs Development (UMP<sup>2</sup>D) – metodologia orientada a objetos para o desenvolvimento de aplicações paralelas e atende aos principais requisitos de desenvolvimento de programas paralelos, tais

como: o balanceamento eficiente de carga, a minimização da comunicação entre os elementos de processamento e a maximização da eficiência computacional.

- Xchart – essa extensão é utilizada no domínio de sistemas reativos – caracterizados pela interação com o ambiente ao qual estão inseridos. A reatividade caracteriza-se pela interação do software com seu ambiente por meio de entradas e saídas produzidas pelo software.

- UWE – foi projetado para modelar o processo de desenvolvimento de Sistema Hipermedia Adaptativa (SHAs) e permite representar visualmente aspectos relevantes de SHAs.

- UML-MC – foi projetada para a representação de Mapas Conceituais.

Desse modo, restam da lista original as extensões Page-Jones, UMLi, WAE e Wisdom, que têm foco em artefatos de interação. A seguir, descrevem-se sucintamente essas extensões consideradas.

#### **4.1.1 Page-Jones**

Page-Jones (2000) desenvolveu dois diagramas adicionais para representar a IU na UML, o diagrama para *layout* de janelas e diagrama para navegação de janelas.

##### **4.1.1.1 Extensão proposta**

a) Diagrama de Layout de Janelas – captura as propriedades de cada janela individual do sistema. É um tipo de protótipo de baixa fidelidade, que exhibe campos, botões e menus das janelas, como se vêem na Figura 3, abaixo.

Esse diagrama define o conteúdo de uma janela a ser apresentada como parte da IU de um sistema; aproxima o *layout* das janelas, mas não contém todo seu eventual refinamento estético; serve tanto como um protótipo de janela como para uma estrutura, sobre a qual são especificadas validações de campos, comportamento janela/campo e mensagens, dentre objetos de negócio e assim por diante.



Trata-se de uma extensão do diagrama de Caso de Uso e pode ser também uma extensão do Diagrama de Pacote, quando representa agrupamentos exclusivamente de elementos de software.

O diagrama mostra a interface de uma janela intitulada "Modificar Lista de Preços". No topo, há uma barra de menu com as opções "Arquivo", "Editar", "Relatórios", "Janela" e "Ajuda". Abaixo do menu, há campos de entrada para "Nome da Lista de Preços:", "Região de Marketing:", "Efetivo:", "Expira:" e "Ativo?". O campo "Ativo?" é um botão de opção. Abaixo desses campos, há uma seção "Detalhes:" que contém uma tabela com quatro colunas: "Código do Produto", "Descrição", "Preço Unitário" e "Comentários". A tabela possui sete linhas vazias. À direita da tabela, há uma barra de rolagem vertical. No canto inferior direito da janela, há um botão "Obter Detalhes".

**Figura 3** – Exemplo de um diagrama de layout de janelas. Fonte: Page-Jones (2001, p. 201).

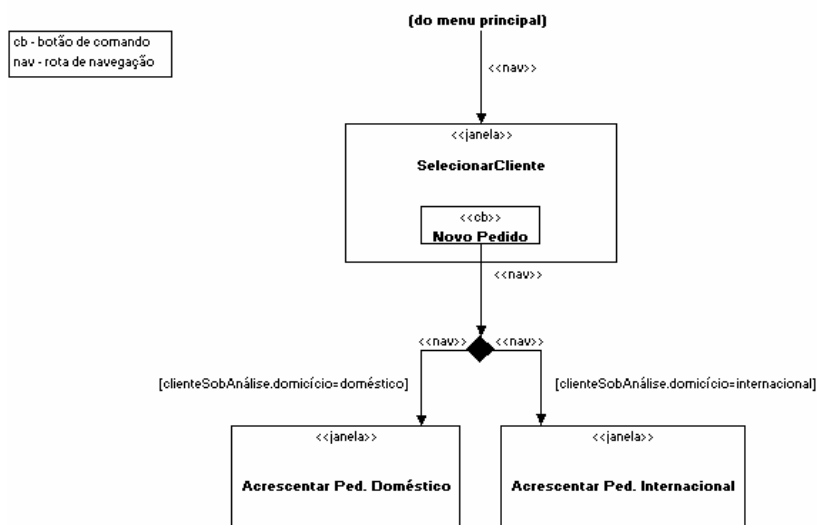
b) Diagrama de navegação de janelas – é uma extensão do diagrama de Estado da UML, cujo objetivo é mostrar como os usuários podem passar de uma janela para outra ao longo de rotas expressivas e importantes nas aplicações.

Um diagrama de navegação de janelas exhibe somente uma abstração de cada janela (o diagrama de *layout* de janelas tem a tarefa de mostrar os detalhes da janela). Ele mostra simplesmente os botões, os clique duplos e outros detalhes que ocasionam transições de uma janela para outra. De maneira similar, mostra apenas itens e subitens do menu que provocam transições.

O diagrama de navegação de janelas habilita desenhistas e usuários de interfaces à exploração da navegação de interfaces antes de o sistema ser construído; abstrai cada janela como um retângulo em forma de esqueleto, o qual suporta os botões de navegação de janelas.

As rotas entre janelas aparecem como setas e são colocadas onde necessárias, com seleções a partir de opções em menus. São utilizados estereótipos

da UML para indicar janelas, botões e rotas de navegação, como visto na Figura 4, a seguir.



**Figura 4** – Diagrama de navegação de janelas com rotas alternativas de navegação.  
Fonte: Page-Jones (2001, p. 204)

#### 4.1.1.2 Aplicação


As extensões propostas por Page Jones são diagramas adicionais da UML, um destinado a layout e outro, à navegação de janelas. Esse autor revela, por meio de estereótipos da UML, preocupação em demonstrar a interface com alguns detalhes, daí a utilização de protótipo e a preocupação em representar a navegação de IU.


#### 4.1.2 UMLi (The Unified Modeling Language for Interactive Applications)


UMLi é um projeto de pesquisa desenvolvido pelo grupo de gerência da informação da Universidade de Manchester, em 1998. Segundo Silva e Paton (2000), a UMLi foi desenvolvida para proporcionar interação amigável entre usuário e sistema. Ela propõe dois novos diagramas UML para o projeto de IU, descritos como Diagrama IU e Diagrama de Atividade Estado de Seleção.


#### 4.1.2.1 Extensão proposta


a) Diagrama IU – destina-se a modelar a interface com o usuário. Estende-se a partir do diagrama de classe da UML e, nele, as instâncias dessas Classes Interativas são os próprios objetos interagindo.


Esse diagrama contém um container vazio , uma janela de alto nível. Esse container é composto por ícones que representam: editores, *displayers*, entradas e *ActionInvokers*.

- *displayers* são representados pelo símbolo  e respondem pela emissão de informações visuais aos usuários;

- entradas são representadas pelo símbolo  e têm como objetivo receber informações emitidas pelos usuários;

- editores são representados pelo símbolo  e são, simultaneamente, *displayers* e entradas;

- *ActionInvokers* são representado pelo símbolo  e respondem pela recepção de informações emitidas pelos usuários na tela de eventos.

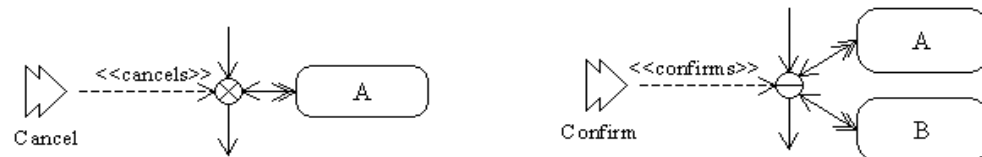
Recipientes são representados pelo símbolo  e podem conter: outros recipientes, editores, *displayers*, entradas e *actionInvokers*;

b) Diagrama Estado de Seleção – mostra uma seqüência de estados e eventos muito semelhante à de um fluxograma, expostos na Figura 5, aplicado na descrição de um processo que transcende a uma classe e que envolve estados de outras classes.

Esse diagrama é alternativa para representar um processo descrito por um Caso de Uso. Para tanto, o diagrama de atividades dispõe dos mesmos elementos dos diagramas de estado, ou seja, eventos e estados, mas não se restringe a um único objeto.

A UMLi criou um novo conjunto de construtores do diagrama de atividades

para modelagem do comportamento UI, chamado de Estado de Seleção. Esse novo diagrama de atividades de falso estado foi desenvolvido para construção de modelagem comum de IU. E, nele, há novos estereótipos para modelagem de fluxo de objetos de IU, os quais identificam a interação do fluxo do objeto e são responsáveis por especificar a colaboração entre os objetos de interação e objetos de domínio.



**Figura 5** – Diagrama de Atividade – Estado de Seleção.

Para o fluxo de interação dos objetos, são utilizados os seguintes estereótipos:

<<presents>> representa o processo de inicialização e torna visível uma atividade, uma ação ou um estado.

<<interacts>> controla o fluxo de interação de uma atividade, ação ou seleção.

<<cancel>> cancela um estado, atividade, ação ou navegação.

<<confirms>> confirma um estado e, geralmente, finaliza uma atividade, ação ou navegação.

<<activates>> ocorre quando o objeto executa uma atividade ou uma ação.

#### 4.1.2.2 Aplicação

A UMLi estende o Diagrama de Classe, transforma-o em dinâmico para modelar a interface com o usuário, criando um conjunto de construtores do diagrama de atividade para representar a interação do fluxo dos objetos.

#### 4.1.3 WAE (Web Application Extension)

Criada por Jim Conallen, a WAE (*Web Application Extension for UML*) estende

a notação UML e traz novos estereótipos com semântica e restrições adicionais, permitindo a modelagem de elementos específicos da arquitetura envolvida numa aplicação Web, incluindo-os nos modelos dos sistemas.

A WAE foi desenvolvida para representar páginas Web, utilizando os mecanismos de extensões: estereótipos, valores atribuídos e restrições.

Jim Conallen (2003) explica que aplicações Web são recursos de software que gerenciam o estado de negócio e adotam uma camada de apresentação centrada na Web, utilizando tecnologias como HTTP, HTML, navegadores e scripts do lado do cliente.

A WAE, em si, se concentra, sobretudo, camada de apresentação de aplicações (CONALLEN, op. cit.).

#### **4.1.3.1 Extensão proposta**



Conallen propõe Experiência do Usuário (UX), a qual é apresentada por meio de um conjunto de diagramas utilizados pra descrever a IU. Segundo seu autor,

*o termo experiência do usuário (UX) é usado para descrever a equipe e as atividades desses especialistas responsáveis em manter a interface com o usuário consistente com os paradigmas atuais e, principalmente, adequado ao contexto no qual se espera que o sistema seja executado. (op. cit. p. 201).*

#### **4.1.3.2 Aplicação**

O UX é um modelo separado, por ser a visualização do sistema de um ponto de vista particular; no caso, do sistema. Por meio de suas telas, ele expressa um modelo de interface com o usuário e fornece um conjunto de diagramas que definirão o *layout* e o modo de navegação das páginas. Os dois estereótipos desse modelo são: tela e formulário, como se vê a seguir, na Tabela 6.

**Tabela 6 – Estereótipo UX.**

Estereótipo	Ícone	Descrição
Tela		Interface apresentada para o usuário.
Formulário		Entrada de dados do usuário que acompanham as solicitações ao servidor.

#### **4.1.4 WISDOM (Whitewater Interactive System Development with Object Models)**

WISDOM propõe um método de desenvolvimento de software simplificado, especialmente adaptado para o desenvolvimento de sistemas interativos por pequenas ou médias equipes.

##### **4.1.4.1 Extensão proposta**

De acordo com Nunes (2000), a arquitetura de IU WISDOM é uma extensão do padrão de análise interface/entidade/controle (model-view-controller, MVC) da UML, a qual refina e estrutura requisitos definidos nos modelos centrados nos usuários. Sua arquitetura compreende três estereótipos de classes: entidade, tarefa e espaço de interação e foi desenvolvida tendo em consideração os seguintes critérios:

a) incluir o conhecimento existente no domínio de arquiteturas para sistemas interativos e que correspondem a padrões testados e bem conhecidos;

b) integrar o padrão existente no UP (*Unified Process*) e na UML, de forma a promover a cooperação, a troca de artefatos e para assegurar a rastreabilidade entre os modelos típicos na engenharia de software e na engenharia de usabilidade;

c) permitir a separação de conceitos entre funcionalidade interna e interface do usuário, não apenas em sua natureza intrínseca, mas também permitindo a co-evolução das arquiteturas de interface interna; e

d) Ser completamente conforme a norma UML, tanto no nível semântico como no nível notacional (sintático).

Estereótipos definidos pela WISDOM:

<<Interaction Space>> é um estereótipo de classe e representa o espaço contido na IU, na qual, o usuário interage com todos os instrumentos e objetos no decorrer de uma tarefa ou conjunto de tarefas inter-relacionadas;

<<navigates>> é um estereótipo de associação entre dois espaços de interação e denota mudança de espaço de interação por parte do usuário;

<<contains>> é um estereótipo de associação entre duas classes de interação e denota que a classe fonte contém a classe alvo (o conteúdo). Esse estereótipo de associação só pode ser utilizado entre dois espaços de interação e é unidirecional;

<<input element>> é um estereótipo de atributo que denota informação recebida pelo usuário, isto é, informação sobre a qual o usuário pode operar;

<<output element>> é um estereótipo de atributo que denota informação apresentada ao usuário, ou seja, informação que o usuário recebe, mas não manipula;

<<action>> é um estereótipo de operação; denota algo que o usuário pode realizar na IU e causa uma alteração significativa no estado interno do sistema.

#### **4.1.4.2 Aplicação**

É um modelo de prototipação evolutiva, no qual o processo de desenvolvimento de software evolui por meio de uma seqüência de protótipos incrementais até atingir um produto final.

Esse modelo está dividido em oito modelos distintos, como se podem ver na Figura 6, a seguir, e envolve 39 conceitos básicos e 29 conceitos de diagramas. Os modelos são: de perfil do usuário, de domínio/negócio, de casos de uso essenciais, de requisitos, de análise, de interação, de desenho e de diálogo.

A organização dos modelos favorece a separação entre conceitos funcionais e componentes de um sistema interativo e sua finalidade é suportar o

desenvolvimento de software centrado no usuário.

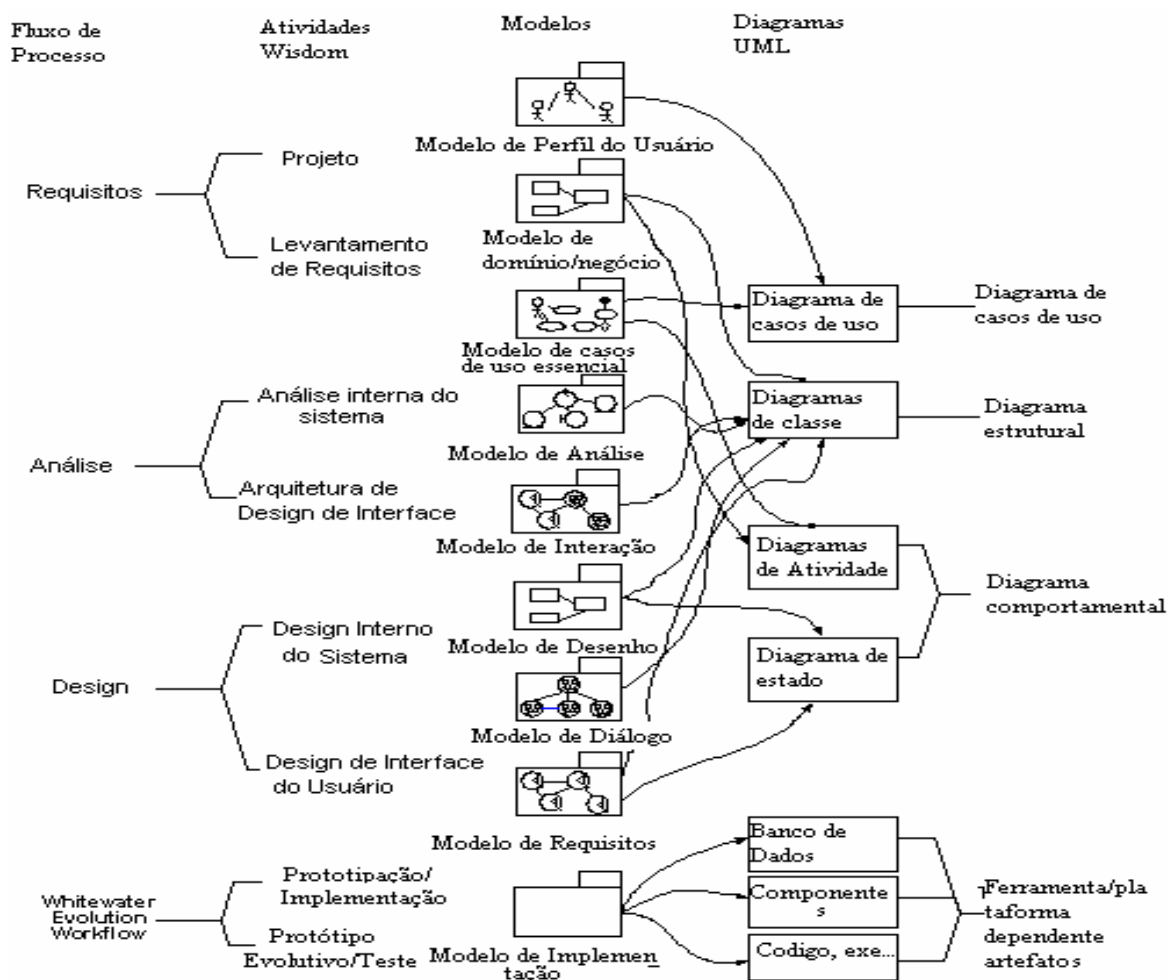


Figura 6 – Notação WISDOM.

## 4.2 Artefatos das Extensões

Considerando as extensões estudadas, verifica-se que todas as extensões UML partem de alguns dos diagramas UML, aos quais os autores acrescentam elementos de informações, para que possam ser mais bem compreendidos.

A Tabela 7, a seguir, é um resumo do estudo e mostra os artefatos das extensões e os diagramas estendidos.



**Tabela 7 – Artefatos das extensões.**

<b>Extensões</b>	<b>Artefatos</b>		<b>Diagrama UML</b>	
Page-Jones	Diagrama de Janela		Diagrama de caso de uso e diagrama de pacote	
	Diagrama de Layout		Diagrama de estado	
UMLi	Diagrama de Atividade		Diagrama de estado	
	Diagrama UI		Diagrama de classe	
WAE	Modelo UX	Descrições de telas e conteúdo	Diagrama de classe	
		Cenários do roteiro		Descrição de caso de uso
		Caminhos de navegação por meio de telas	Fluxo de tela	Diagrama de caso de uso e diagrama de estado
			Mapa de caminho de navegação	
Fluxo de navegação de telas				
WISDOM	Modelo perfil do usuário		Diagrama de caso de uso	
	Modelo de domínio/negócio		Diagrama de classe e diagrama de atividade	
	Modelo de casos de uso essencial		Diagrama de caso de uso	
	Modelo de Análise		Diagrama de classe	
	Modelo de Interação		Diagrama de classe	
	Modelo de desenho		Diagrama de classe e diagrama de estado	
	Modelo de diálogo		Diagrama de classe	
	Modelo de Requisitos		Diagrama de classe e diagrama de estado	
	Modelo de Implementação		-----	

## **5 ESTUDO COMPARATIVO DAS EXTENSÕES UML APLICADAS NO PROJETO SISGAU**

Para desenvolvimento deste estudo comparativo, foram utilizadas as quatro extensões descritas no Capítulo anterior. Essas extensões foram aplicadas em um projeto do Centro de Tecnologia da Informação, Automação e Mobilidade (CIAM) do Instituto de Pesquisas Tecnológicas do Estado de São Paulo (IPT), na Seção de Sistemas Corporativos (SSC). O projeto visa ao desenvolvimento de um sistema interativo denominado Sistema de Gestão de Árvores Urbanas (SISGAU).

O SISGAU é uma ferramenta destinada a auxiliar no processo de gestão da arborização de uma cidade, o qual permite o gerenciamento de um inventário das árvores existentes, contendo localização, espécie, família, gênero, condição de entorno, características dendrométricas e interferências, dentre outras informações relevantes à criação de estratégias para o manejo dos exemplares.

Entre outras características passíveis de serem inventariadas, destacam-se os itens de Sanidade Biológica, os quais permitem o cadastramento detalhado da ocorrência de organismos fitopatogênicos, tais como: fungos, cupins, brocas, formigas ou outros. A existência desses organismos pode ser cadastrada de acordo com a região da árvore em que esses surgem: raiz, colo, tronco ou copa, visando a um maior controle do processo de tratamento e permitindo o estudo da distribuição das espécies nas árvores.

É também permitido o cadastramento de defeitos internos e externos que possam interferir na estabilidade dos exemplares, possibilitando a realização de estudos a respeito de seus riscos de queda. E, por fim, é possível a elaboração de relatórios gerenciais e estatísticos.

Para uma análise do exemplar arbóreo, é necessário o cadastramento das seguintes informações:

- a) nome do responsável pelo cadastramento e data;
- b) endereço do exemplar arbóreo, juntamente com dados de georreferenciamento;
- c) condição de entorno – tipo de calçada, canteiro, permeabilidade, distanciamento das interferências;

- d) especificação – gênero, espécie, família, nome popular, origem, porte;
- e) características dendrométricas – medidas da árvore, como: altura, perímetro altura do peito, diâmetro altura do peito, diâmetro da copa e outros;
- f) avaliação da raiz, colo, tronco e copa – interferências, características específicas (tais como presença de flores e de broto ladrão, dentre outras), diagnósticos, prospecção (quando necessário) e sanidade biológica (presença de fungos, cupins, brocas ou formigas, dentre outros parasitas);
- g) avaliação geral – situação, recomendação, risco de queda e observação; e
- h) fotos – identificação por meio de fotografia do exemplar ou de partes dele.

O cadastro do exemplar arbóreo no SISGAU segue uma seqüência que é parte do protocolo de pesquisa. Para facilitar o trabalho do pesquisador, adotou-se a abordagem de *wizard* – o usuário tem de passar tela por tela até chegar à última, que é a interface de fotos. E, somente no fim do cadastramento, há confirmação do cadastro do exemplar arbóreo.

O SISGAU foi desenvolvido em plataforma *.Net*, com banco de dados Microsoft SQL Server. E, por se tratar de um sistema complexo, selecionou-se uma pequena parte do todo e, dentro das funções de sanidade biológica, escolheu-se a Avaliação do Tronco, cuja função é a de registrar os dados de diagnósticos e os dados de prospecção<sup>3</sup> do tronco. Os dados da prospecção são inseridos no SISGAU via cabo USB.

O estudo comparativo será focalizado nas operações de avaliação, diagnóstico e prospecção do tronco, conforme apresentado no diagrama de classe, exposto na Figura 7, a seguir. As operações de avaliação e diagnóstico do tronco são obrigatórias, enquanto a operação de prospecção depende da avaliação de necessidade feita pelo usuário.

---

<sup>3</sup> Prospecção é uma avaliação mais profunda, com interferência de um equipamento que fura o exemplar arbóreo e detecta a presença de oco interno e possíveis enfermidades existentes

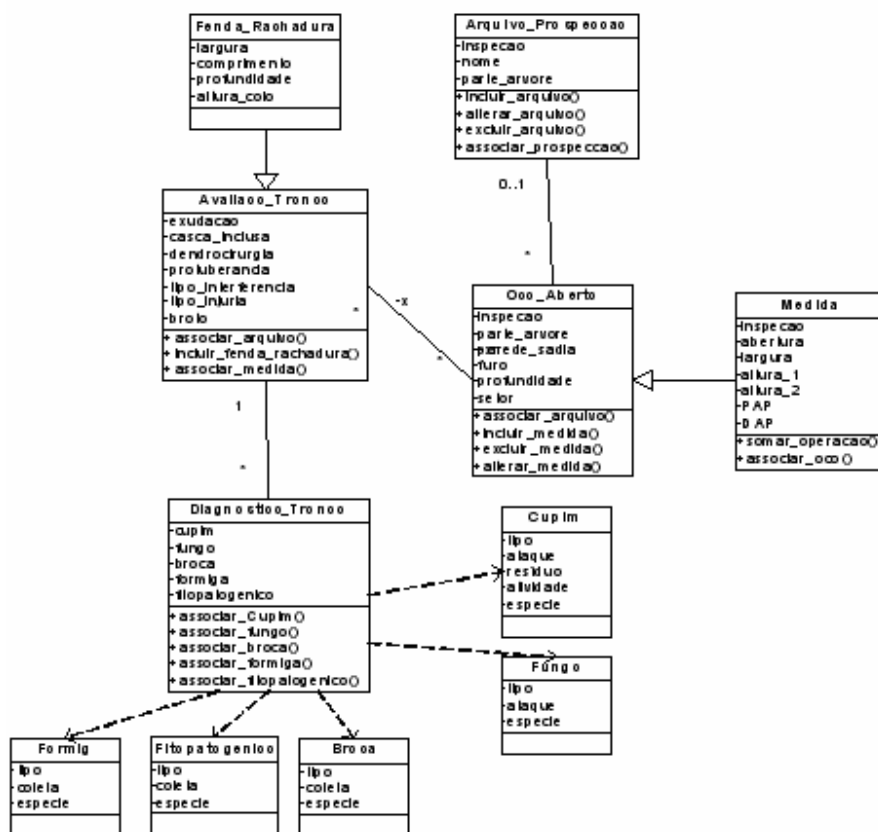


Figura 7 – Amostra do Diagrama de Classe UML – SIGAU.

## 5.1 Critérios para comparação

Para orientar a avaliação comparativa, recorreu-se ao conceito de Qualidade em Uso, o qual se caracteriza por ser o grau em que o produto pode ser utilizado por usuários específicos, para alcançar objetivos explícitos, com eficácia, produtividade, segurança e satisfação, em um contexto específico de uso (ISO/IEC 9126-4).

Embora as extensões da UML não sejam propriamente um produto de software, o conceito de qualidade em uso é útil para analisar o relacionamento entre as extensões e seus usuários – os projetistas de sistemas interativos.

Os critérios de comparação utilizados para o estudo foram estas métricas da ISO 9126-4 (2001): eficácia, produtividade e satisfação. A segurança, neste estudo, não foi considerada por não ser aplicável, já que é definida como a capacidade de

oferecer níveis aceitáveis de risco de danos a pessoas, negócios, software, propriedade ou ao ambiente, em um contexto de uso especificado.

- a) Eficácia – é a capacidade do produto de permitir que usuários atinjam metas específicas com acurácia e completude, em um contexto de uso específico. Para o uso de uma extensão ser eficaz, o modelo ou diagrama precisa ser capaz de comunicar as informações necessárias de forma clara e simples, para ser compreendido plenamente por seus leitores. No fim da sua modelagem, ele deverá chegar ao resultado proposto com qualidade.
- b) Produtividade – é a capacidade do produto de permitir que seus usuários empreguem quantidade apropriada de recursos em relação à eficácia obtida, em um contexto de uso especificado. A modelagem da extensão será produtiva se o projetista não perder tempo para modelá-la e se os *stakeholders* não tiverem de gastar tempo, tentando entendê-la.
- c) Satisfação – é a capacidade do produto de satisfazer usuários em um contexto de uso especificado. Satisfação é o sentimento de prazer ou de desapontamento resultante da comparação do desempenho esperado pela modelagem da extensão (ou resultado dela) em relação às expectativas do projetista ou de outra pessoa envolvida no processo.

O estudo comparativo analisa as extensões UML de acordo com o levantamento das fases para a construção da camada interativa, apresentada no Capítulo 2, a qual foi dividida em: análise de requisitos, design e teste.

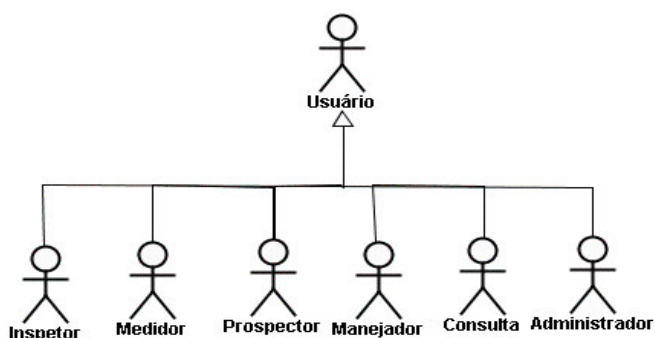
## **5.2 Análise de Requisitos**

No Capítulo 2, foram identificados artefatos necessários para a análise de requisitos: solicitação dos envolvidos, perfis dos usuários, personas, cenários, especificação de casos de uso, modelo de tarefas, pesquisa de campo, tabela de signos e diagramas de meta.

Nas extensões estudadas, foram encontrados para a fase de análise de requisitos, o modelo de perfil do usuário e o modelo de diálogo, ambos da WISDOM.

Modelo de Perfil do Usuário – é representado por um diagrama de casos de uso, com os estereótipos de classificação, ação e composição. Esse modelo segue a perspectiva definida pelo mapa de papéis do usuário, definidos por Constantine e Lockwood (1999), o qual ilustra e organiza características dos usuários e seus papéis, tipo de usuário, conjunto de interesses, comportamento, responsabilidades e expectativas em relação ao sistema.

A Figura 8, Modelo de Perfil do Usuário da WISDOM, a seguir, demonstra os seis usuários do SIGAU, a saber: inspetor, medidor, prospector, manejador, consulta e administrador.



**Figura 8** – Modelo Perfil do Usuário – WISDOM.

a) Inspetor

<<classificação>> – inspeciona os exemplares arbóreos.

<<tipo de acesso>> – cadastra, altera e consulta os exemplares.

<<ação>> – navega por todas as telas do sistema.

b) Medidor

<<classificação>> – mede os exemplares arbóreos (altura, diâmetro do tronco e da copa).

<<tipo de acesso>> – cadastra, altera e consulta.

<<ação>> – navega somente em interface de Dendrometria, após logar e localizar a árvore.

c) Prospector –

<<classificação>> – é responsável pela prospecção dos exemplares arbóreos.

<<tipo de acesso>> – cadastra, altera e consulta.

<<ação>> – navega somente em telas de Prospecção do Colo e do Tronco, após logar e localizar a árvore.

d) Manejador

<<classificação>> – é responsável pelo manejo dos exemplares arbóreos.

<<tipo de acesso>> – consulta.

<<ação>> – navega somente em telas de Manejo, após logar e localizar a árvore.

e) Consulta

<<tipo de acesso>> – consulta.

<<ação>> – navega em todas as telas do sistema.

f) Administrador

<<classificação>> – é responsável por todo o sistema.

<<tipo de acesso>> – cadastra, altera, consulta.

<<ação>> – cadastra as tabelas auxiliares (espécies, usuários ou outros), além de navegar por todas as telas do sistema.

Essa extensão mostra – por meio do diagrama de casos de uso – os usuários do sistema e seus papéis, acrescentando descrições textuais para explicar responsabilidades, expectativas e comportamentos.

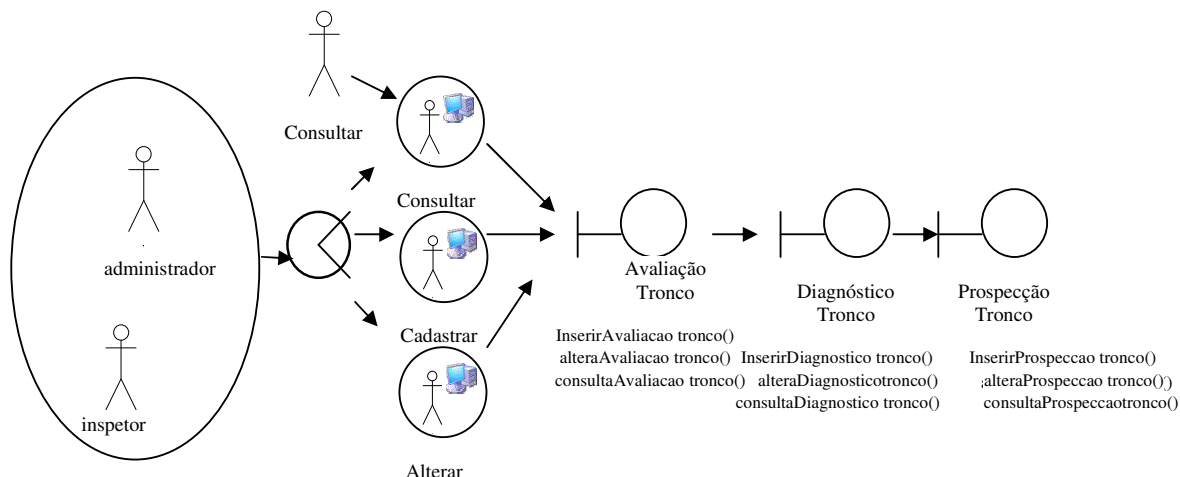
Modelo de Diálogo – esse modelo especifica a estrutura de diálogo da aplicação interativa, focando as tarefas e seus usuários.

A Figura 9, a seguir, descreve esta situação: um grupo de usuário (administrador e inspetor) pode consultar, cadastrar e alterar as telas *avaliação*, *diagnóstico* e *prospecção do tronco*, seguindo essa seqüência. No caso, a interface de Avaliação do Tronco tem as seguintes funções:

InserirAvaliacaotronco(),alteraAvaliacaotronco(),consultaAvaliacaotronco().

Nota-se que o usuário *Consulta* apenas consulta as telas *avaliação tronco*, *diagnóstico tronco* e *prospecção tronco*.

É interessante ressaltar neste modelo que, além de ele demonstrar os tipos de usuários, suas tarefas (consultar, alterar e cadastrar) e as telas que percorre, ele descreve também as funções contidas em cada tela.



**Figura 9** – Modelo de diálogo – WISDOM.

### 5.3 Design

Os artefatos identificados no Capítulo 2 como necessários para o Design são: protótipos, mapa do sistema, fluxo de interação, diagrama de interação e *Wireframe*.

As extensões que se aplicam a essa fase são:

- Page-Jones – Diagrama de navegação de Janelas e Diagrama de Janela;
- UMLi – Interação do Fluxo de Telas e Diagrama IU;
- WAE – Mapa de caminho de navegação, Descrição de telas e conteúdo, Fluxo de navegação; e
- WISDOM – Fluxo de telas – modelo de interação, modelo de desenho ou apresentação.

#### 5.3.1 Page-Jones

Page-Jones (2001) propõe o diagrama de navegação de janelas, que é uma extensão do diagrama de estado da UML, cujo objetivo é mostrar como os usuários podem passar de uma janela para outra ao longo de rotas expressivas e importantes na aplicação.

Seguindo o fluxo exposto na Figura 10 a seguir, verifica-se que a interface (denominada pelo autor de *janela*) Avaliação do Tronco foi acionada a partir da interface Avaliação do Colo. Nessa interface, é executada a função

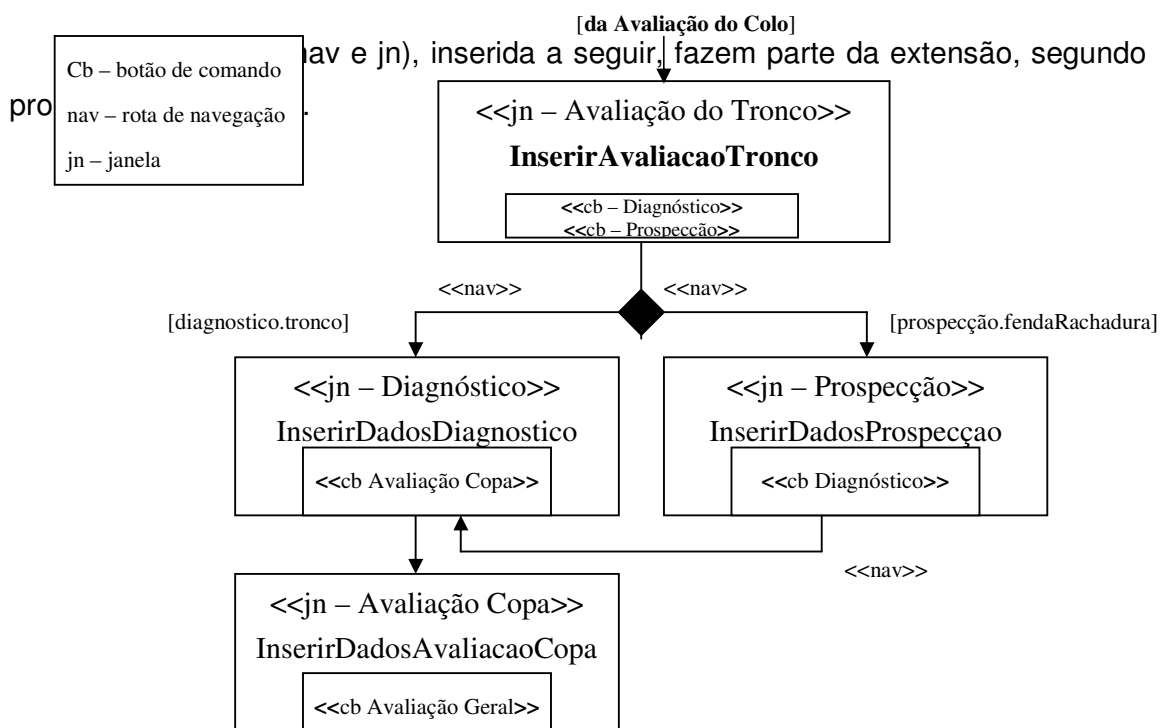


InserirAvaliacaoTronco e, a partir dos botões diagnóstico e prospecção, é possível navegar para outras duas telas: Diagnóstico ou Prospecção.

A interface Diagnóstico recebe os dados do tronco (demonstrado pelo colchete diagnostico.tronco, da classe diagnóstico) e a interface prospecção recebe os dados da FendaRachadura (classe).

Caso o usuário escolha a interface Prospecção, ele terá de passar necessariamente pela interface Diagnóstico e, se ele escolher a interface Diagnóstico, a próxima interface que ele percorrerá é a de Avaliação Copa.

Todos os estereótipos demonstrados na



**Figura 10** – Diagrama de navegação de janelas – Fonte: Page-Jones (2001).

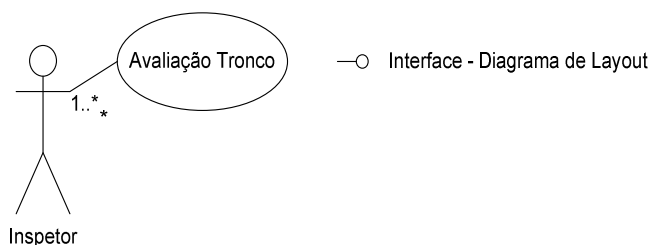
O que torna esta extensão atraente é o fato de ela ser simples em demonstrar a navegação de telas (prisma), as funções que a interface necessita para sua

execução e os eventos (botões). Outra vantagem é o fato de não haver necessidade de uma ferramenta específica para desenhar esse diagrama.

Outro diagrama proposto por Page-Jones (2001) é o de janelas, que tanto pode ser uma extensão do diagrama de caso de uso como pode ser uma extensão do diagrama de pacote, quando representa agrupamentos exclusivamente de elementos.

Para desenhar o diagrama de janelas, foi utilizada a ferramenta Microsoft Visio. Foi desenhado o diagrama de caso de uso e, nesse, foi indicada uma interface denominada Diagrama de Layout, exposto na Figura 11, a seguir.

No Diagrama de Layout – Figura 12, a seguir – foram colocados os elementos da interface sem nenhum refinamento.



**Figura 11** – Caso de Uso Avaliação Tronco – Fonte: Page Jones (2001).

**Avaliação Tronco**

**Interferência**

Tipos de Interferência...

**Diagnóstico**

Sem Dano Aparente

Injúria Mecânica

Não existe  
 Anelamento  
 Outros

Broto Ladrão    Exudação de Seiva    Casca Inclusa    Dendrocirurgia    Protuberância

Largura (m)	Comprimento (m)	Fendas/Rachaduras	
		Profundidade (m)	Altura em Relação ao Colo (m)

Voltar   Diagnóstico

**Figura 12** – Diagrama de Janela – Avaliação Tronco.

A proposta de Page-Jones (2001) é bem simples: ele estende o diagrama de caso de uso da UML para demonstrar o protótipo.

### 5.3.2 UMLi

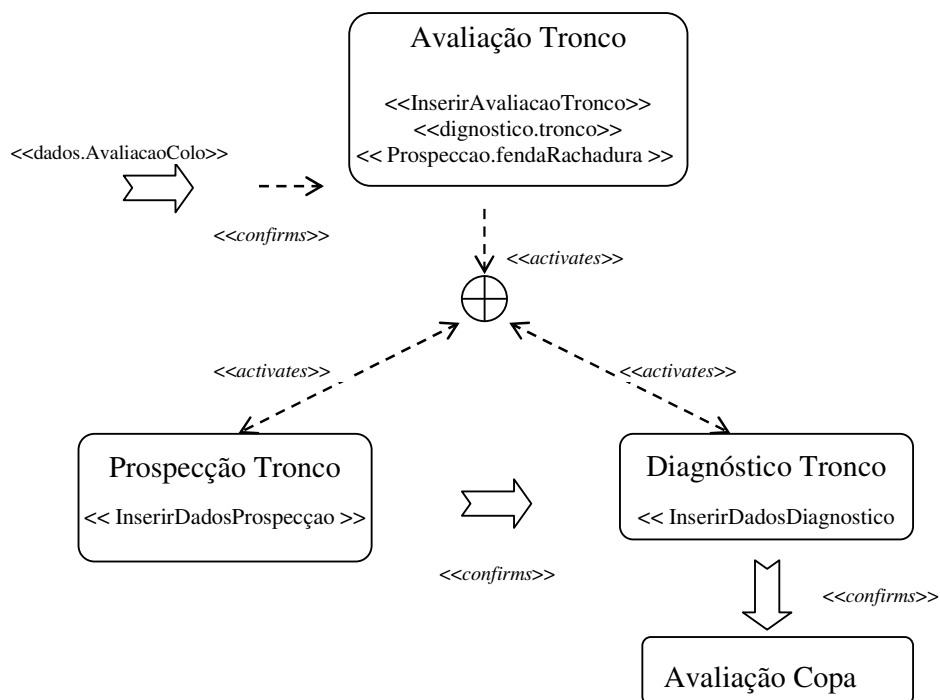
Silva e Paton (2000) propõem o diagrama de Interação do Fluxo de Telas, o qual é uma alternativa para representar o processo descrito por um caso de uso. Para tanto, esse diagrama dispõe dos mesmos elementos dos diagramas de estado, ou seja, eventos e estados, mas esses não se restringem a um único objeto.

O diagrama de Interação do Fluxo de Telas é muito parecido com o diagrama de navegação de janelas do Page-Jones (2001).

Na Figura 13 a seguir, está demonstrada a seguinte situação: os dados da interface Avaliação Colo (descrito pelo estereótipo << AvaliacaoColo.dados>>, classe AvaliacaoColo) são enviados, após confirmação, para a interface Avaliação Tronco que é composta pela função InserirAvaliacaoTronco e pelos dados do diagnóstico, presentes na classe diagnóstico (<<diagnostico.tronco>>) e da fendaRachadura, presente na classe prospecção (<<prospeccao.fendaRachadura

>>).

De acordo com os dados recebidos, o sistema ativa a interface Prospecção Tronco ou Diagnóstico Tronco e, se ativar a interface Prospecção, a seqüência posterior será a interface Diagnóstico Tronco e Avaliação Copa.



**Figura 13** – Interação do Fluxo de Telas – UMLi.

Em suma, a extensão apresenta o nome das telas: Avaliação Tronco, Prospecção Tronco e Diagnóstico Tronco; o fluxo das telas (  $\Rightarrow$  ); os eventos <<activates>> e <<confirms>>; as funções <<InserirAvaliacaoTronco>>, << InserirDadosProspeccao >> e << InserirDadosDiagnostico>>; e os dados <<dados.AvaliacaoColo>>, <<dignostico.tronco>> e <<


Prospeccao.fendaRachadura >>.


Outro diagrama proposto por Silva e Paton (2000) é o UI, utilizado para modelar interface. Esse diagrama é estendido a partir do diagrama de classe da UML e as instâncias dessas classes são os próprios objetos interagindo.


Para desenhar a UMLi, seus autores indicam uma ferramenta própria, denominada ARGOi, desenvolvida por eles mesmos, a ser baixada do site <http://trust.utep.edu/umli/software.html>. Entretanto tal ferramenta não funcionou, daí o fato de ter havido dificuldade na modelagem.

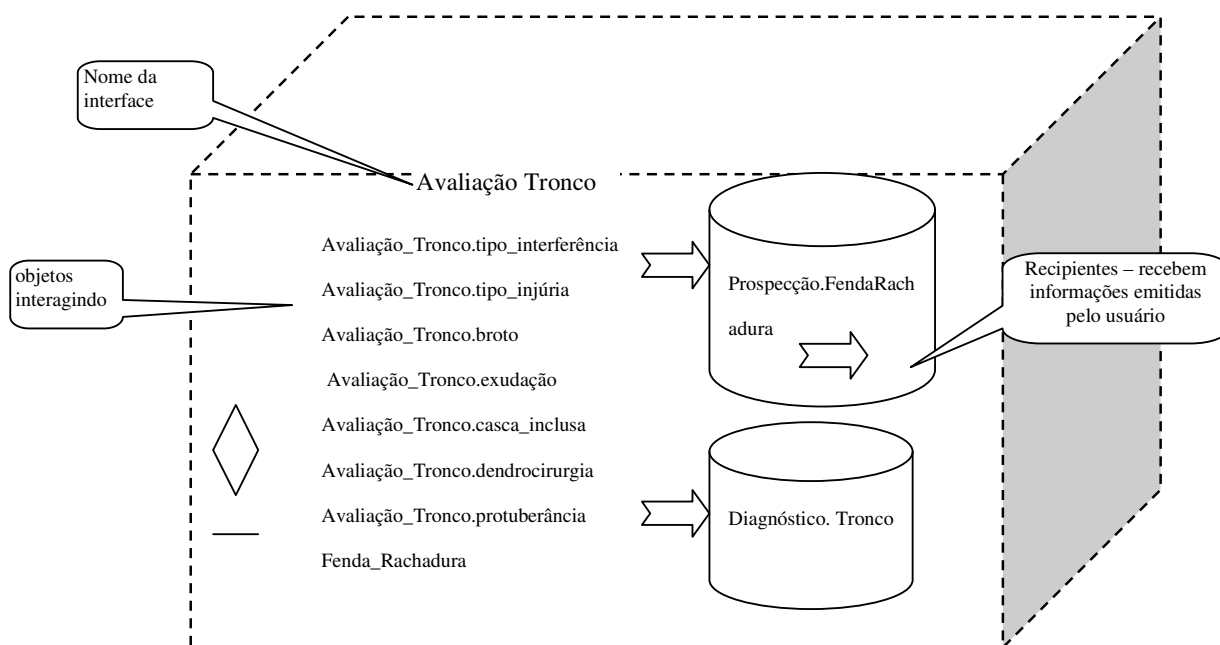
Como descrito anteriormente, para modelar o Diagrama IU, é necessário ter o diagrama de classes da UML. O diagrama de classe do SISGAU está demonstrado na Figura 7.

O Diagrama IU, representado na Figura 14 a seguir, apresenta a interface Avaliação Tronco, na qual, estão os objetos extraídos do diagrama de classe da

UML, da Figura 7 e, nesse, constam os ícone editores , que é um ícone de entrada de dados.

Dentro da interface, há também dois recipientes  e, dentro desses recipientes, há informações emitidas pelos usuários – do diagnóstico e da prospecção. Essas serão utilizadas em outras interfaces.

O símbolo  é responsável por receber informações emitidas por usuários tanto fora como dentro dos recipientes.



**Figura 14 – Diagrama IU.**

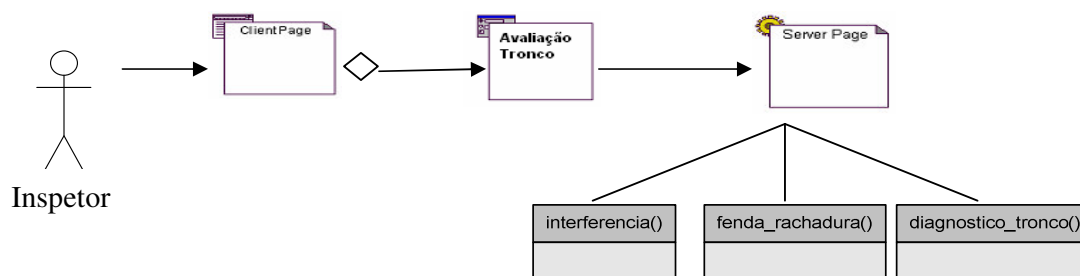
### 5.3.3 WAE

Conallen (2003) propõe um novo conceito na modelagem de sistemas e apresenta o modelo de experiência do usuário (UX), o qual é usado para descrever a equipe e as atividades dos responsáveis em manter a interface com o usuário.

Para a fase de Design, Conallen (op. cit.) propõe na UX quatro artefatos: Fluxo de tela, Mapa de Caminho de Navegação, Descrição de Telas e Conteúdo e Cenários do Roteiro.

#### a) Fluxo de Tela

Um artefato da UX é o Fluxo da tela, na Figura 15 a seguir, o usuário inspetor acessa a interface Avaliação Tronco, por meio de uma página *client*, que acessa uma página *server* a qual contêm as funções do sistema: interferência, fenda\_rachadura e diagnostico\_tronco.



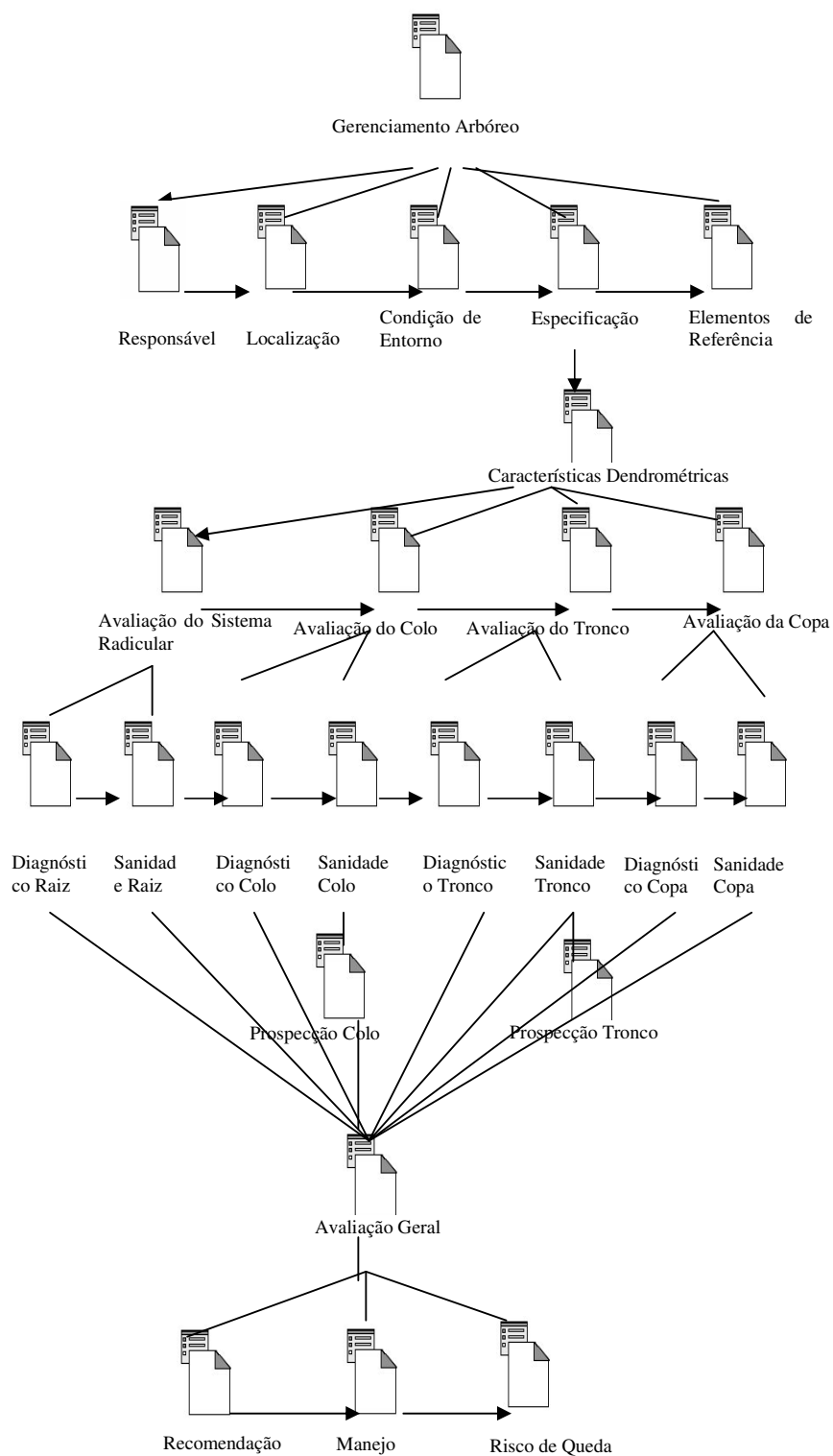
**Figura 15 – Fluxo da tela – WAE.**

O fluxo de tela demonstra o tipo de usuário, a interface, as funções executadas e o tipo de armazenamento – cliente ou servidor.

#### b) Mapa de Caminho de Navegação

O artefato mapa de caminho de navegação é bem simples e expressa o nome das interfaces e o caminho que o usuário deve percorrer para navegar no sistema. A Figura 16, a seguir, apresenta o mapa de navegação completo do SISGAU.

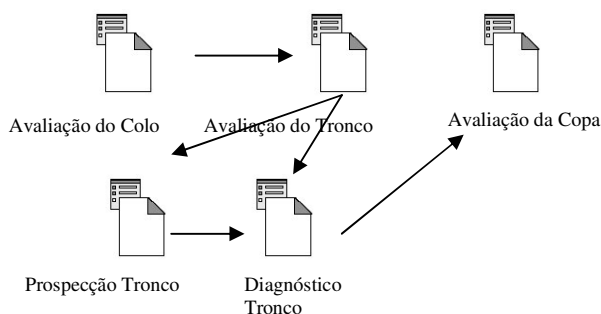
Como descrito anteriormente, no SISGAU, não há um menu onde o usuário seleciona a tela que ele quer cadastrar. Desse modo, ele tem de seguir uma seqüência de telas (*Wizard*) e, quando modelada no Mapa de Caminho de Navegação, essa seqüência de telas tornou-se confusa.



**Figura 16** – Mapa de caminho de navegação – WAE.



A Figura 17, a seguir, demonstra apenas a parte do sistema utilizada para o estudo comparativo. Uma dificuldade encontrada nesse modelo é o fato de não possuir simbologia específica para quando o fluxo é alternativo – no caso do estudo, o usuário cadastra dados na interface avaliação tronco e, dependendo dos dados que ele insere, esse vai para a interface de prospecção tronco ou para a interface de diagnóstico tronco.



**Figura 17** – Caminhos de Navegação das interfaces selecionadas para o estudo comparativo.

### c) Diagrama de Descrições de Telas e Conteúdo

Esse artefato do modelo UX é uma extensão do diagrama de classe da UML. A interface é representada com uma classe estereotipada, com ícones e atributos estereotipados, como se vêem a seguir na Figura 18. O nome da interface é usado como nome da classe – tem um símbolo indicando que é uma interface – e a descrição da interface capturada é usada como a descrição da classe. As classes podem ser organizadas em pacotes. Neste estudo, são analisadas duas interfaces: Avaliação Tronco e Diagnóstico Tronco, os dados de entrada da interface são indicados pelo estereotipo <<dexibido>> e os dados são gerenciados pelo estereotipo <<dbanco>>.

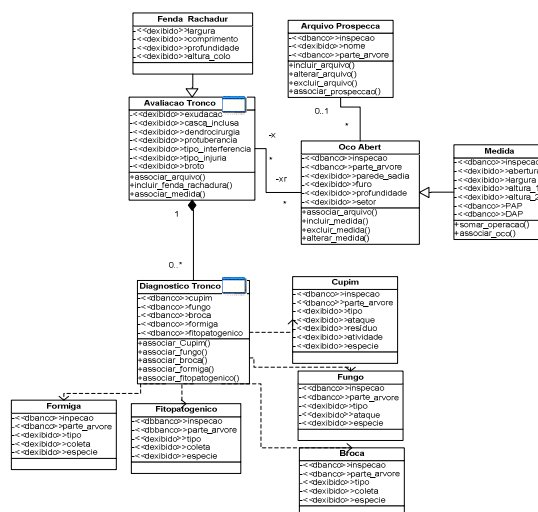
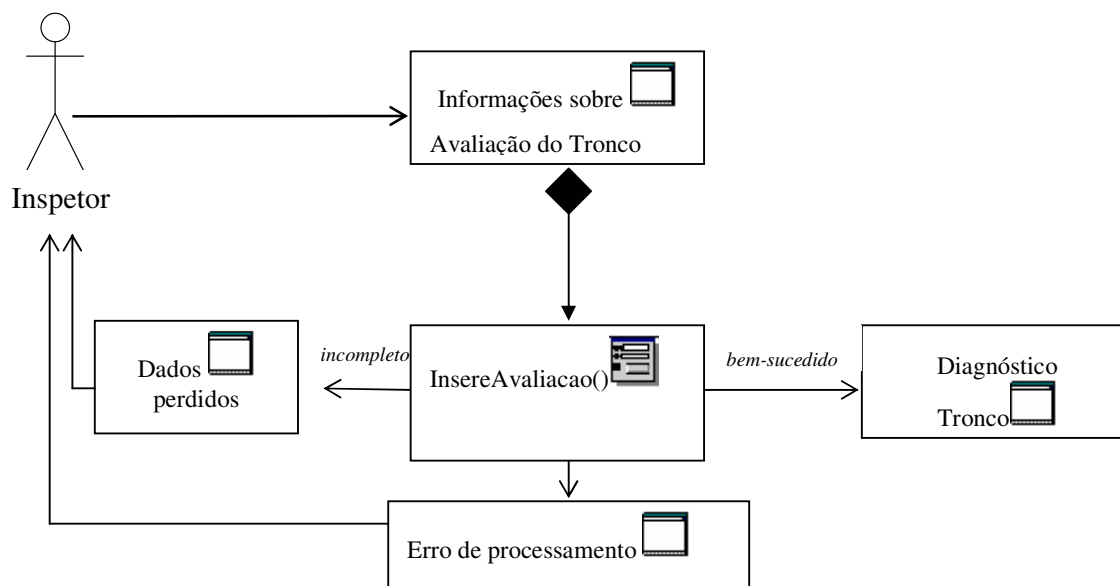


Figura 18 – Descrição de Telas e Conteúdo – WAE.

Exemplificando, o diagrama de Descrições de telas e conteúdo é um diagrama de classe da UML com estereótipos indicando quais dados são de entrada – aqueles produzidos pelo usuário – e quais dados são gerenciais – aqueles produzido pelo sistema, além de um ícone, indicando quais classes são interfaces.

#### d) Cenários do Roteiro

Conalen (2003), no Artefato Cenários do Roteiro, descreve um fluxo de telas com entrada de dados do usuário, visto a seguir, na Figura 18. Nesse, o usuário fornece informações sobre a interface (Avaliação do Tronco) e a navegação é orientada para a próxima interface, desde que se tenha obtido sucesso na inserção de dados. Ou, explicando melhor, a Figura 19 demonstra esta situação: o usuário inspetor acessa a interface Avaliação do Tronco, que dispara a função `InserAvaliacao()`. Se os dados forem consistentes, aparece a interface Diagnóstico Tronco; caso os dados estejam incompletos, aparece a interface Dados Perdidos (interface de erro); e, se houver erro de processamento (problemas externo ao sistema, por exemplo conexão da rede), aparece a interface Erro de Processamento.

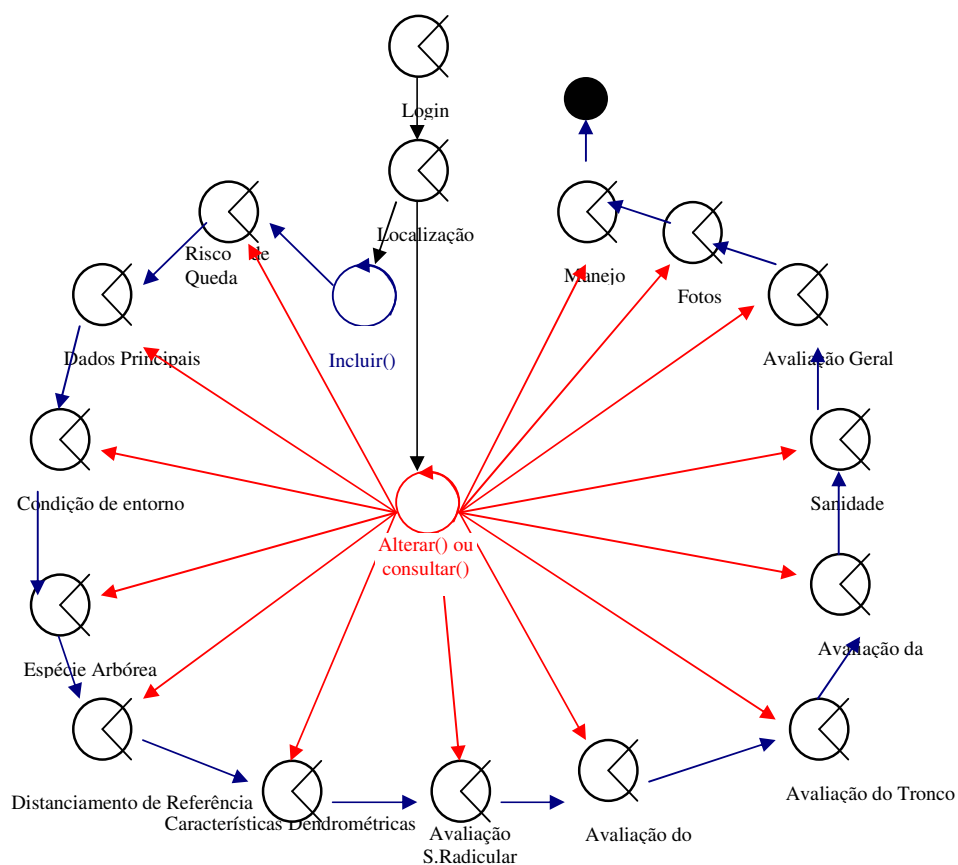


**Figura 19** – Cenários do Roteiro – Fluxo de navegação determinado pela entrada de dados do usuário.

Essa extensão apresenta um cenário de roteiro de forma bem macro, sem muito detalhamento de interface e não há descrições das funções do sistema nem dos eventos.

#### 5.3.4 WISDOM

Nunes (2000) propõe dois artefatos para essa fase: modelo de interação e modelo de apresentação. O modelo de interação é um fluxo de casos de uso. Na Figura 20, está demonstrada a seguinte situação: se o usuário acessar o caso de uso localização, terá duas opções, incluir ou alterar/consultar. Caso ele opte por incluir, ele segue uma seqüência de telas até finalizar (*wizard*), caso contrário, ele pode acessar a interface que quer alterar ou consultar.



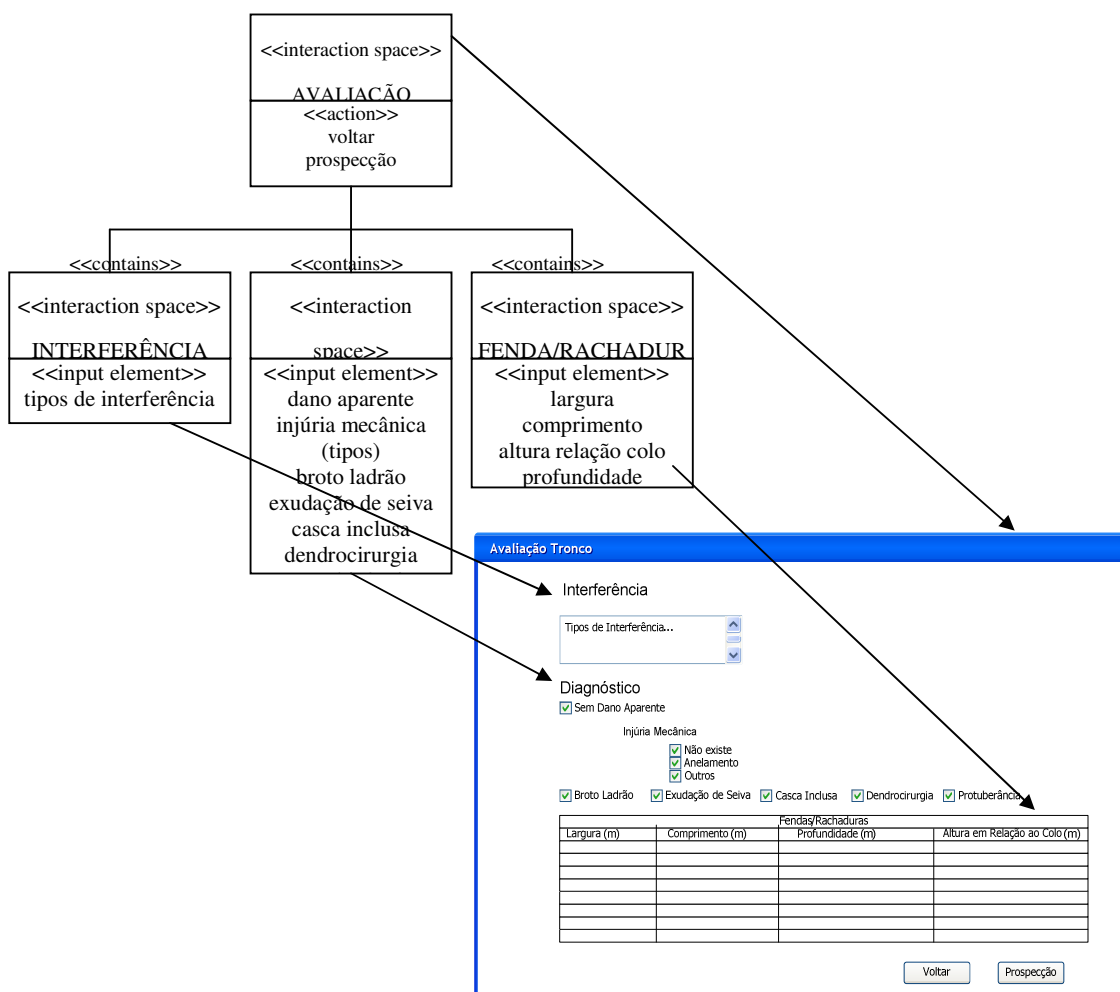
**Figura 20** – Fluxo de Casos de Uso – Modelo de Interação – WISDOM.

A proposta desse artefato não é descrever o fluxo de interfaces, mas demonstrar a seqüência dos casos de uso. Na modelagem desse fluxo, constatou-se que, quando os casos de uso seguem obrigatoriamente uma seqüência que independe da ação do usuário, fica inviável a sua visualização.

Outra proposta de Nunes (2000) é o modelo de apresentação, uma extensão do diagrama de classe. Esse modelo apresenta o nome da Interface, seus eventos e objetos. Após modelar o modelo de apresentação, é possível desenhar o protótipo da interface.

A Figura 21, a seguir, expõe o modelo de apresentação, o qual traz um diagrama de classe estereotipado, em que está descrita a interface Avaliação Tronco. Essa possui dois eventos: voltar ou prospecção. Dentro da interface Avaliação Tronco, há três divisões: Interferência, Diagnóstico e Fenda/Rachadura e

os objetos que compõem cada divisão. O protótipo é construído a partir do diagrama de classe estereotipado.



**Figura 21** – Modelo de desenho ou apresentação – WISDOM.

## 5.4 Teste

Os artefatos identificados no Capítulo 2 como necessários para o Teste são: questionário com o usuário, documentação de testes, manual de instalação e manual do usuário. Não há artefatos para representação dessa fase nas extensões estudadas.

## 5.5 Análise Comparativa

Após ter aplicado as extensões no projeto do SISGAU, foi possível construir um mapa das extensões de acordo com as fases do projeto. A Tabela 8, a seguir, mostra que as extensões UML se concentram na fase de Design, a fase de análise de requisitos está pouco representada e não há nenhum artefato para a fase de teste. Esse fato demonstra duas possibilidades; a preocupação dos autores com a fase de design e/ou a deficiência de artefatos da UML no projeto de UI.

**Tabela 8** – Fases da camada interativa x Artefatos extensões UML.

Fases	ARTEFATOS			
	Page-Jones	UMLi	WAE	WISDOM
Análise de Requisitos	----	----	----	Modelo de Perfil do Usuário e Modelo de Diálogo
Design	Diagrama de Janela e Diagrama de Layout.	Diagrama de Atividade e Diagrama UI.	Fluxo de tela, Mapa de caminho de navegação, Diagrama de Descrições de Telas e Conteúdo e Cenários do Roteiro.	Modelo de interação e modelo de apresentação.
Testes	-----	----	----	----

No momento da modelagem dos artefatos das extensões, foram medidas a eficácia, a produtividade e a satisfação. E foi constatada uma série de vantagens e desvantagens. As seções seguintes discutem esses aspectos.

### 5.5.1 Page-Jones

Os dois artefatos propostos por Page-Jones (2001), o Diagrama de Layout de Janela e o Diagrama de Navegação de Janela, são artefatos simples de modelar e

que não precisam de ferramenta específica para desenhá-los.

O objetivo principal do Diagrama de Layout de Janela é adicionar o protótipo aos diagramas UML. Esse protótipo apresenta detalhes da interface que, muitas vezes, não são descritos nos casos de uso nem nos diagramas de classe.

A vantagem principal que esse diagrama oferece é a de detalhar o caso de uso por meio de um protótipo. Outra vantagem é promover uma estrutura para uma posterior especificação da interface, a qual inclui validações cruzadas de campos requeridos, sincronizações entre campos, verificações de banco de dados e assim por diante. Um ponto negativo é que alguns requisitos não aparecem na especificação, por exemplo, os requisitos não funcionais.

O objetivo do Diagrama de Navegação de Janela é mostrar como os usuários podem passar de uma janela para outra ao longo de rotas expressivas e importantes da aplicação. O Diagrama de Navegação de Janela mostra as rotas de interação para um caso de uso. A vantagem desse diagrama está na representação da navegação e na opção de rotas alternativas. Uma desvantagem é que não há descrição do tipo de usuário.

Aplicando as extensões de Page-Jones (2001), foi possível medir a qualidade de uso nos seguintes casos:

- a) Eficácia – nesse caso, as extensões são classificadas como eficazes, pois cumprem o objetivo proposto pelo autor;
- b) Produtividade – são usadas para modelar as extensões e dispensam ferramenta específica. Não houve dificuldade em modelar as extensões que são simples, fato que as torna produtivas;
- c) Satisfação – no contexto de especificação de interfaces, o desempenho esperado pela modelagem das extensões e seus respectivos resultados foram satisfatórios e a expectativa foi alcançada.

### **5.5.2 UMLi**

Os dois artefatos propostos por Silva e Paton (2000) – Diagrama UI e Diagrama de Atividade – são artefatos não tão simples de modelar, pois requerem

ferramenta específica. Os autores desenvolveram a ARGOi, mas não foi possível a sua utilização neste trabalho.

O objetivo do artefato Diagrama UI é modelar a interface com o usuário e, para tanto, ele apresenta entradas e saídas de dados, além dos objetos da tela.

Uma desvantagem é que somente quem conhece a simbologia entende o que o diagrama quer demonstrar, pois não se trata de algo ontológico.

O objetivo do Diagrama de Atividade é apresentar a seqüência de estados e eventos da tela, deixando sua aparência muito semelhante à de um fluxograma.

Uma vantagem desse artefato é que ele está diretamente ligado ao Diagrama UI. Seguindo o Diagrama de Atividade e interpretando o Diagrama UI, é possível compreender a interação e projetar a interface. Outra vantagem desse diagrama é o fato de ele apresentar um processo descrito por um caso de uso.

Aplicando as extensões de Silva e Paton (2000), foi possível medir a qualidade de uso nos seguintes casos:

- a) Eficácia – qualidade classificada como eficaz, porque cumpre o objetivo proposto pelo autor;
- b) Produtividade – dispensa o uso de ferramentas específicas para modelar as extensões, mas apresenta dificuldade para se modelarem as extensões e demanda domínio da simbologia, o que a torna improdutiva;
- c) Satisfação – apresenta resultado favorável após aplicação, mas se torna insatisfatória pela dificuldade em exprimir os detalhes da interface.

### **5.5.3 WAE**

Os artefatos proposto por Conallen (2003) serão eficazes para modelar a interface, se utilizados conjuntamente, pois um completa o outro.

O objetivo do artefato Fluxo de Tela é descrever os caminhos ao longo de todas as telas que participam dos cenários e os caminhos de navegação esperados são modelados com associações entre as telas.

Uma vantagem desse artefato é o fato de ser de fácil modelagem e entendimento. Sua desvantagem é não dispor de simbologia para demonstrar um



fluxo alternativo.

O objetivo do Mapa de Caminho de navegação é apresentar uma visão da aplicação cujo objetivo é mostrar como usuários do sistema navegarão nele. Esse mapa é representado em diagrama hierárquico em árvore.

Uma vantagem do Mapa de Caminho de Navegação é a apresentação de rotas de navegação, juntamente com as funcionalidades do sistema e os usuários.

O objetivo do artefato Descrição de Telas e Conteúdo é apresentar os objetos de cada interface e as classes que interagem com a mesma. É uma desvantagem desse artefato é a não-apresentação de eventos (“voltar”, “próxima página” etc.).

O objetivo do Cenário do Roteiro é compreender e estruturar os cenários, mostrando o ponto de vista do usuário.

A vantagem desse artefato é que ele conta a história da aplicação por meio das telas. Outra vantagem é a oferta de um esquema de roteiro mapeado para o modelo de caso de uso. Sua desvantagem é fornecer visão pouco detalhada das interfaces e da interação.

Aplicando as extensões WAE, foi possível medir a qualidade de uso nos seguintes casos:

- a) Eficácia – extensões classificadas como eficazes, porque cumprem os objetivos propostos pelo autor;
- b) Produtividade – descrevem as telas, o fluxo de interação, mas não são completas para modelar sistemas interativos. É necessária a utilização de todos os diagramas para demonstrar parte da modelagem, o que as torna improdutivas;
- c) Satisfação – o usuário fica insatisfeito em modelar as extensões WAE, pois é necessário utilizar todos os diagramas, inclusive os da UML, para demonstrar parte da modelagem e esse fato torna o processo de modelagem muito demorado.

#### **5.5.4 WISDOM**

O objetivo do artefato Modelo do Perfil do Usuário é apresentar os usuários do

sistema, mediante descrição de: tipo de usuário, comportamento, responsabilidades e expectativas em relação ao sistema. A vantagem desse diagrama é apresentar o usuário e suas tarefas aos *stakeholders*.

O objetivo do Modelo de diálogo é especificar a estrutura de diálogo da interface, ou seja, explicitar as seqüência das tarefas e fornecer relações que assegurem consistência dos diferentes elementos de apresentação.

A vantagem desse artefato é o fato de descreve as funções contidas em cada tela, além de demonstrar os tipos de usuários, suas tarefas (consultar, alterar e cadastrar) e as telas por ele percorridas.

O objetivo do Modelo de Interação é organizar os elementos da arquitetura do sistema, diretamente relacionados com sua estrutura de utilização.

A vantagem desse artefato é que ele demonstra a seqüência dos casos de uso. E uma desvantagem é que ele inviabiliza a modelagem e sua visualização, quando os casos de uso seguem obrigatoriamente uma seqüência.

O objetivo do Modelo de Apresentação é definir a parte perceptível do sistema, ou seja, a interface, focando como as diferentes entidades de apresentação estão estruturadas de forma a realizar a interação com o usuário.

Aplicando as extensões da WISDOM, foi possível medir a qualidade de uso nos seguintes casos:

- a) Eficácia – complexidade para se atingir o objetivo dos artefatos no estudo comparativo, o que as classificam com ineficazes;
- b) Produtividade – a construção do modelo de perfil do usuário foi produtiva. A modelagem dos outros quatro modelos foi mais demorada, o que as classifica como improdutivas;
- c) Satisfação – a modelagem é prazerosa, mas o acúmulo de informação e o tempo despendido para a modelagem deixam o usuário sem vontade de usá-las, embora seu resultado final seja satisfatório.

## **5.6 Proposta de Artefatos para Modelagem de Sistemas Interativos**

Este trabalho levantou os artefatos extraídos dos modelos de processo; em

seguida, descreveu os artefatos UML, para demonstrar a relação entre a construção da camada interativa e as visões UML; e, por fim, aplicou os artefatos das extensões UML em um projeto real.

A seguir, é apresentado um mapa dos artefatos levantados neste trabalho, de acordo com as fases do processo de sistemas interativos e suas respectivas visões UML.

a) Análise de Requisitos – Visão de caso de uso.

Artefatos extraídos dos modelos de processos – Perfis dos usuários, solicitação dos envolvidos, personas, cenários, especificação de casos de uso, modelos de tarefas, pesquisa de campo, tabela de signos e diagrama de metas;

Artefatos UML – Descrição de caso de uso, diagrama de caso de uso e, eventualmente, diagrama de atividade;

Artefatos das extensões UML – modelo perfil do usuário e modelo de diálogo.

b) Design – Visão Lógica e de Concorrência

Artefatos extraídos dos modelos de processos – Definição da arquitetura, protótipo ou desenho das telas, mapa do sistema, fluxo de interação ou diagrama de interação, *Wireframes*;

Artefatos UML – Diagramas de classe, objeto, seqüência, estado, colaboração, atividade e implementação

Artefatos das extensões UML – Diagrama de Layout de Tela, Diagrama de Navegação, Diagrama UI, Diagrama de Atividade, Mapa de caminho de navegação, Fluxo de tela, Modelo de interação Descrição de telas e conteúdo, Cenários de roteiro, Modelo de desenho e Fluxo de Navegação de Telas.

c) Testes – Visão de componentes e de implantação

Artefatos extraídos dos modelos de processos – questionário com o usuário, documentação de testes, manual de instalação e manual do usuário;

Artefatos UML – Diagramas de componentes e implantação.

Artefatos das extensões UML – não foram encontrados.

Analisando os artefatos levantados, os quais foram descritos anteriormente, e a modelagem das interfaces em um projeto real, verifica-se que não há necessidade de produzir todos os artefatos na modelagem de sistemas interativos. A pressão dos clientes por prazos menores obriga os desenvolvedores a buscarem soluções de modelagem mais rápidas, eficazes e que atendam às expectativas dos clientes e da equipe de desenvolvimento.

Tendo em vista essas premissas, foram extraídos dos Capítulos 2 e 5 um conjunto de artefatos classificados por esta pesquisadora, no contexto da qualidade em uso, como eficazes, produtivos e satisfatórios. Esses artefatos foram extraídos dos modelos propostos pelos autores estudados no Capítulo 2 e na aplicação das extensões descritas no Capítulo 5.

Para a fase de Análise de Requisitos – Visão de Caso de Uso –, é sugerido um questionário do usuário, para conhecer as necessidades do mesmo, a partir deste questionário faria um mapeamento das personas e desenharia os cenários e por fim para modelar a interação utilizaria o modelo de tarefas..

Para a fase de Análise de Design – Visão lógica e de concorrência –, é sugerida uma definição de arquitetura, diagrama de estado estendido por meio do modelo de diálogo da WISDOM, construção do protótipo a partir dos casos de uso, mediante a utilização do diagrama de janela de Page-Jones (2001) em conjunto com um mapa de caminho de navegação e fluxo de telas da WAE.

Para a fase de **Testes** – Visão de componentes e de implantação –, são necessários Cenários do Roteiro para execução dos testes, manual de instalação e manual do usuário.

A Tabela 9, a seguir, apresenta as fases do processo de sistemas interativos juntamente com visões, artefatos propostos e autores que sugerem esses artefatos.

**Tabela 9** – Artefatos sugeridos para a modelagem de projeto de sistemas interativos.

Fases e visões	Artefatos Propostos	Autores/ extensões
Análise de Requisitos – visão de caso de uso.	Questionário com o usuário.	Mayhew (1999)
	Modelo de Perfil do usuário.	Nunes (2000)
	Cenários.	ISO 13407 (1999), Norman e Draper (1986), Nielsen (1993), Mayhew (1999) e Paula (2003).
	Personas	Cooper (1999)
	Modelo de tarefas.	Norman e Draper (1986), Nielsen (1993), Hix e Hartson (1993) e Mayhew (1999).
Análise de Design – Visão lógica e de concorrência.	Definição da Arquitetura – <i>Deployment</i> .	Booch, Jacobson e Rumbaugh (1999).
	Diagrama de estado estendido por meio do modelo de diálogo.	WISDOM de Nunes (2000).
	Diagrama de Janela.	Page-Jones (2001).
	Mapa de caminho de navegação.	WAE de Conallen (2003).
	Fluxo de telas.	WAE de Conallen (2003).
Testes – Visão de componentes e de implantação	Cenários do roteiro.	WAE de Conallen (2003).
	Manual de instalação e manual do usuário.	Mayhew (1999).

Verifica-se que uma parte expressiva dos artefatos propostos foi extraída das extensões UML. As extensões deixam lacunas nas fases de análise de requisitos e testes, porém é possível complementar esse vazio com os artefatos dos autores estudados no Capítulo 2.

## 6 Conclusões

Artefatos se constituem em uma atividade essencial no processo de desenvolvimento de software. A evolução do projeto é registrada por meio deles, para que sejam criadas as bases necessárias às etapas de desenvolvimento do projeto, aí incluídos: levantamento, desenvolvimento, treinamento, utilização e manutenção do sistema, como ensina Sanches (2001).

Muitos tipos de artefatos são produzidos ao longo do processo de desenvolvimento, tais como: solicitação dos clientes, especificações de requisitos e manual do usuário, dentre outros. Assim, é comum uma organização gastar de 20 a 30% de todo o esforço de desenvolvimento na elaboração de artefatos (PRESSMAN, 2004).

Os artefatos, por sua vez, têm de apresentar qualidade, para que atividades de desenvolvimento, avaliação e manutenção possam ser realizadas com sucesso (SANCHES, op. cit.), além de ter de demonstrar precisamente as intenções do cliente, dos projetistas e dos desenvolvedores.

Entretanto este trabalho levantou os artefatos dos modelos de processo, tal como sugere a literatura – artefatos gerados pela UML e artefatos das extensões UML – os quais são propostas para modelagem de interação das interfaces, para o desenvolvimento do sistema interativo.

Fez-se um estudo comparativo das extensões, por meio de sua aplicação a um sistema real, no qual, o foco da comparação foi a representatividade dos artefatos gerados, de modo a oferecer ao projetista um meio de expressar precisamente suas intenções em todo o processo de desenvolvimento, medindo, em um contexto de uso, a eficácia, a produtividade e a satisfação.

Uma das conclusões obtidas a partir deste trabalho é o fato de os artefatos gerados pelos processos de desenvolvimento de sistemas interativos não serem explícitos. Foi necessário que a autora inferisse nos artefatos das atividades sugeridas por vários dos autores pesquisados, porque não há padronização de artefatos propostos pelos autores.

Outra conclusão apresentada nesta Dissertação é a de que a modelagem de sistemas interativos não pode se basear somente em elementos definidos pelo metamodelo da UML. A UML, por ser uma linguagem de modelagem, não atende às necessidades de representação de um sistema interativo, já que é originalmente destinada a oferecer uma forma visual gráfica de comunicação para representação dos principais conceitos e elementos de um sistema.

Embora seja utilizada amplamente pela indústria de desenvolvimento de softwares, que reconhece suas muitas qualidades, a UML é dotada de complexidade quando da identificação de elementos característicos de uma IU em diagramas, tais como, as tarefas que o usuário deve executar;

Entretanto uma das fortes características da linguagem de modelagem unificada, presente em seu metamodelo é o mecanismo de extensão que essa linguagem implementa.

Entende-se ser impraticável modelar artefatos de sistemas interativos somente a partir de elementos da UML e seus diagramas. Uma das alternativas, então, é utilizar extensões da UML em conjunto com os diagramas da UML, tendo em vista suas visões.

Os autores das extensões estudadas, Page-Jones (2001), Silva e Paton (2000), Nunes (2000) e Conalen (2003), recomendam a modelagem de sistemas interativos por meio dos diagramas UML, aos quais eles acrescentam extensões e adicionam artefatos para facilitar o entendimento dos *stakeholders*.

De modo geral, a análise e a implementação dos artefatos das extensões mostraram que:

- o primeiro passo para promover o desenvolvimento de softwares interativos com qualidade e fácil de utilizar deve ser a garantia de que os envolvidos nos projetos (*stakeholders*) gerem artefatos de qualidade e compreendam essa atividade de forma contextualizada;
- para modelar a interação, em geral, os autores adaptaram o diagrama de estado da UML e incluíram estereótipos (elementos de modelos), para facilitar a visualização e a contextualização;

- não há produtividade no desenvolvimento dos artefatos propostos pelas extensões por faltarem ferramentas que apóiem a modelagem com os elementos específicos das extensões;
- falta manual de instrução ou uma documentação sintetizada que indique como aplicar as extensões em um projeto de sistemas interativos;
- as extensões têm, em geral, simplicidade na semântica;
- as extensões possibilitam a construção de mecanismos adaptados a necessidades específicas.
- somente os artefatos das extensões não conseguem satisfazer todas as fases do projeto de sistemas interativos. A fase mais representada é a de Design.

Enfim, este estudo demonstrou que não há na literatura uma regra. Ademais os autores estudados não fazem indicações de quais artefatos são necessários para modelagem do processo de sistemas interativos. Esse trabalho procurou nos autores dos processos de desenvolvimento de software, na UML e em suas extensões, o melhor dos artefatos propostos e, assim, sugeriu um conjunto desses que sejam capazes de demonstrar especificação, visualização, modelagem e documentação.

## **6.1 Trabalhos Futuros**

As conclusões deste trabalho apontam para a necessidade ou oportunidade de trabalhos futuros que possam confirmar e aprofundar os resultados aqui apresentados. Esses trabalhos são:

- a) Realização do estudo das formas de aproveitamento do uso dos artefatos em projetos de sistemas interativos;
- b) Produção de artefatos, utilizando extensões UML para representar:
  - perfil dos usuários;
  - diálogo da interação; e
  - plano de testes de usabilidade.



- c) Realização de um novo estudo comparativo, usando a capacidade de representação multimídia ou multimodais, como os sistemas de voz; e
- d) Estender a avaliação realizada por esta autora a outros grupos de desenvolvedores, projetistas e usuários, verificando a eficácia, a produtividade e a satisfação dos artefatos propostos neste trabalho.

## REFERÊNCIAS

- BARCELOS, I. F.; ALUÍSIO, S. M. **Aplicação de um processo de avaliação de usabilidade em sistemas interativos**. São Paulo: ICMC-USP, 2002.
- BOEHM, B.W. "A Spiral Model of Software Development and Enhancement". In: **Readings in Human-Computer Interaction: Toward the Year 2000**, San Mateo (CA), 1995.
- BOMBANI, L.; CIANCARINI, P.; VITALI, F. "Sophisticated hypertext functionalities for software engineering". In: **Proceedings of the 22nd International Conference on Software Engineering**. Limerick (Irlanda), 2000.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **The Modeling Language User Guide**. Boston (Massachusetts): Addison-Wesley, 1999.
- BROWN, J. "Methodologies for the Creation of Interactive Software, Technical Report CS-TR-96/1". Victoria University of Wellington. Wellington (New Zealand), 1996. (in: <http://www.mcs.vuw.ac.nz/comp/Publications/archive/CS-TR-96/CS-TR-96-1.ps.gz>.) acessado em 20/03/2007, às 22h).
- CHAVES, R. **Aspectos e MDA – Criando modelos executáveis baseados em aspectos**. Dissertação de Mestrado. Florianópolis, Universidade Federal de Santa Catarina, 2004.
- COOPER, A. **The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity**. SAMS. McMillan, 1999.
- CONALLEN, J. **Desenvolvendo aplicações web com UML**. São Paulo: Campus, 2003.
- CONSTANTINE, L. L.; LOCKWOOD, L. A. D. **Software for use: A Practical guide to the models and methods of usage-centered design**. Boston: Addison-Wesley, 1999.
- DIX, A. J. **Formal methods and interactive systems – Principles and practice**. New York: New York University /Department of Computer Science, 1987.
- DIX, A. J.; FINLAY, J. E.; ABOWD, G. D.; BEALE, R. **Human-computer interaction**. 2.ed. London (England): Prentice Hall Europe, 1998.
- EHN, P., "**Scandinavian Design: On Participation and Skill**", in Usability: Turning Technologies into Tools, P. S. Adler and T. A. Winograd, editors, Oxford University Press, New York, NY, 1992.
- FILGUEIRAS, L. V. L. "Interface Homem x Computador" – Notas de Aula. Instituto de Pesquisas Tecnológicas do Estado de São Paulo. São Paulo, 2004.
- FOWLER, M. "Model driven architecture". Disponível em: <http://martinfowler.com/bliki/ModelDrivenArchitecture.html>. Acesso em 22/11/2007.
- FURLAN J. D. **Modelagem de objetos por meio da UML**. São Paulo: Makron Books, 1998.
- GOULD, J. D., LEWIS, C. **Designing for Usability: Key Principles and What Designers Think**. Communications of the ACM, New York (USA), 1985.
- GRØNBÆK, K.; KYNG, M.; MOGENSEN, P. Toward a cooperative experimental system development approach. In: Kyng, M., & Mathiassen, L. (Eds.), Computers

and design in 116 context, p. 201-238, 1997. Cambridge, MA: The MIT Press. Disponível em <<http://www.daimi.au.dk/~kgronbak/homepage/pubs/PB-412.pdf>> Acesso em 29 ago 2008.

HIX, D.; HARTSON, H. R. **Developing user interfaces**: Ensuring Usability Through Product & Process. New York (EUA): Wiley & Sons Inc., 1993.

IEEE Std. 830. **IEEE Guide to software requirements specification**. New York (EUA): The Institute of Electrical and Electronics Engineers, 1984.

ISO/IEC 9126/4. **Software engineering – Software product quality- Part 4: Quality in use metrics**. International Standard ISO 9126-4, 2000.

ISO 13407 – **Human-Centred Design process for interactive systems**. International Organization for Standardization, 1999.

JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. **The unified software development process**. Boston (Massachusetts): Addison Wesley, 1999.

KOCH, N.; BAUMEISTER, H.; HENNICKER, R.; MANDEL, L. "Extending UML to Model Navigation and Presentation in Web Applications". **Proceedings of Modelling Web Applications in the UML Workshop – UML'2000**, York (UK), oct., 2000.

LARMAN, C. **Utilizando UML e padrões**. 2ª Ed. Rio de Janeiro: Bookman, 2004.

LEITE, J. C. "Specifying a user interface as an interactive message". **HCI International 2003 - 10th International Conference on Human-Computer Interaction**, Creta, 2003.

LUCENA, F. N. **Xchart – Um modelo de especificação e implementação de gerenciadores de diálogo**. Tese de Doutorado. Instituto de Computação da Universidade Estadual de Campinas Campinas (SP). 1997.

MAYHEW, D. J. **The usability engineering lifecycle**. San Francisco: Morgan-Kaufmann, 1999.

MELO, A. M.; BARANAUSKAS, M. C. C. "Construindo um portal para crianças com crianças: Uma abordagem participativa ao design". **3rd International Conference on Engineering and Computer Education**. São Paulo, 2003.

\_\_\_\_\_. "Design para a Inclusão: desafios e proposta". In: **Simpósio sobre Fatores Humanos em Sistema Computacionais**, Natal, RN, 2006.

MULLER, M.J., HASLWANTER, J.H., DAYTON, T. Participatory Practices in the Software Lifecycle, Second Edition, Handbook of Human-Computer Interaction, Elsevier Science B.V., 1997.

NIELSEN, J. **Usability engineering**. New York: Academic Press, 1993.

NORMAN, D. A.; DRAPER, S. W. **User centered system design**: New perspectives on Human-Computer Interaction. Hillsdale (EUA): Lawrence Erlbaum Associate Publishers, 1986.

NUNES N. J. "Object modeling for user-centered development and user interface design – The Wisdom". **WISDOM Approach**, 2001.

NUNES, N. J., CUNHA; J. F. "WISDOM: A software engineering method for small software development companies". **IEEE Software**, outubro, 2000.

NUSEIBEH, B.; EASTERBROOK, S. "Requirements engineering – A roadmap". **Proceedings of the 22nd International Conference on Software Engineering**. Limerick (Ireland), Jun., 2000.

ODELL, James; PARUNAK, H. Van Dyke; BAUER, Bernhard. "Extending UML for agents". **Proceedings of the agent-oriented information systems workshop at the 17th National Conference on artificial intelligence – AaiAAI – 2000**. Encontrado em: <http://www.jamesodell.com/ExtendingUML.pdf>. acessado em março/2005.

OLIVETE A. L. S.; TRINDADE Jr, O. "UMP2D – A utilização da UML no desenvolvimento de aplicações paralelas". In: XXVII Seminário Integrado de Software e Hardware, 2000, Curitiba (PR), 2000.

OMG (Object Management Group). "UML 2.0 Infrastructure specification". October, 2004. Encontrado em <http://www.omg.org/docs/ptc/04-10-02.pdf> . Acesso em janeiro/2005.

PAGE-JONES, M. **Fundamentos do desenho orientado a objeto com UML**. São Paulo: Makron Books, 2001.

PAULA, M. G. **Projeto da interação humano-computador baseado em modelos fundamentados na engenharia semiótica** – Construção de um modelo de interação. Dissertação de Mestrado, Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro, 2003.

PAULA, M. G.; BARBOSA, S. D. J.; LUCENA, C. J. P. "Relating Human-Computer interaction and software engineering concerns – Towards extending UML through an interaction modeling language", in: **Closing the Gaps: Software Engineering and Human-Computer Interaction, Workshop at Interact 2003**, Zurich (Switzerland), 2003, p. 40-46.

PAULA, M. G. **ComunIHC-ES** – Ferramenta de apoio à comunicação entre profissionais de IHC e Engenharia de Software. Tese de doutorado. Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro, 2007.

PREECE, J.; Rogers, Y.; Sharo, H.; Benyon, D.; Holland, S.; Carey, T. **Human-Computer Interaction**. Workingham: Addison-Wesley Publishing, 1994.

PRESSMAN, Roger S. **Software engineering** – A practitioner's approach. 6th ed. New York: McGraw-Hill, 2004.

ROBINSON, G.; CRESPO, S. "UML-MC – Estendendo a notação gráfica da UML para suportar mapas conceituais de sistemas de auxílio em ambientes de Ensino a Distância". **VII Simpósio de Informática e II Mostra Regional de Software Acadêmico**, Uruguaiana, 2002.

ROCHA, H. V.; BARANAUSKAS, M. C. C. **Design e avaliação de interfaces Humano-Computador**. Campinas: Nied/Unicamp, 2003.

ROSSON, M. B.; CARROLL, J. M. **Usability engineering** – Scenario Based Development of Human-Computer Interaction. San Francisco (USA): Morgan Kaufmann, 2002.

SANCHES, R. "Gerência de Configuração", In: SANCHES, R. **Qualidade e produtividade em software**. 4a edição. São Paulo: Makron Books, 2001.

SCHULER, D. AND NAMIOKA, A. “**Participatory Design: Principles and Practices**”, Lawrence Earlbaum, NJ, 1993.

SHNEIDERMAN, B. **Designing the user interface** – Strategies for effective Human-Computer interaction. 3rd Ed. Boston (MA): Addison-Wesley, 1998.

SILVA, B. S. **MoLIC. 2ª. ed. Revisão de uma linguagem para modelagem da interação humano-computador**. Dissertação de Mestrado. Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro, 2005.

SILVA, P. P.; PATON, W. User interface modeling in UMLi. **IEEE Software**. Stanford: Stanford University (CA, USA), 2003.

\_\_\_\_\_. **User interface modeling with UML**. Manchester: University of Manchester/ Department of Computer Science, 2000.

SOMMERVILLE, Ian. **Engenharia de software**. 8ª edição. Rio de Janeiro: Pearson Education, 2007.

SVINTERIKOU, M.; THEODOULIDIS, B. “The temporal unified modelling language – TUML”. **Timelab Technical Report** – UMIST. Manchester (UK), outubro, 1997.

THOMAS, D. **MDA** – “Revenge of the modelers or UML utopia?”. **IEEE Software**, Los Alamitos (CA, USA) 2004.

ZOWGHI, D. “Does global software development need a different requirements engineering process?” **Proceedings of International Workshop on Global Software Development** – In conjunction with (ICSE 2002), Orlando (Florida, EUA), 2002.