

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Modelagem de Sistemas de Informação na
Internet como Sistemas de *Workflow* sobre
Documentos Estruturados**

por

MARISTELA REGINA WEINFURTER

Dissertação apresentada como
requisito para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Dr. Roberto Tom Price
Orientador

Porto Alegre, agosto de 2001.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Weinfurter, Maristela Regina

Modelagem de Sistemas de Informação na Internet como Sistemas de *Workflow* sobre Documentos Estruturados por Maristela Regina Weinfurter – Porto Alegre: PPGC da UFRGS, 2001.

104 f.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2001. Orientador: Price, Roberto Tom.

1. Sistemas de informação na Web. 2. *Workflow*. 3. Documentos estruturados. 4. Modelagem de sistemas de informação na Web. 5. Internet. I. Price, R.T. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof^a Wrana Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitor Adjunto de Pós-Graduação: Prof. Philippe Olivier Alexandre Navaux

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

Agradecimentos

Quero agradecer, acima de tudo a Deus, Meu Senhor, que me conduziu, iluminou e encheu de coragem e perseverança.

Ao meu orientador, professor Dr. Roberto Tom Price, que conseguiu conduzir de forma excelente as idéias e a estruturação desta dissertação.

A meu marido Yan Karlo e a minha filhinha Larissa.

A vários amigos, colegas e familiares que de alguma forma, tiveram um momento de participação neste processo: Marlene Weinfurter, Ciléia de Oliveira, Alsira Weinfurter, Isabel Fernandes, Flávio Azevedo, Douglas Dutra, Abigail, Telma, Wesley, Pio, Musse, Mehran Misagi e Mariangela.

“...mas os que esperam no Senhor renovam as suas forças,
sobem com asas como águias, correm e não se cansam,
caminham e não se fatigam.” (Isaías 40:36).

Sumário

Lista de Abreviaturas e Siglas	7
Lista de Figuras	8
Lista de Tabelas.....	9
Resumo	10
Abstract	11
1 Introdução.....	13
1.1 Objetivos.....	15
1.1.1 Objetivos Específicos	16
1.1.2 Resultados e Contribuições da Proposta.....	16
1.2 Organização do Trabalho.....	17
2 Workflow.....	18
2.1 Metamodelo da definição de um workflow.....	18
2.2 Conceitos relacionados em um workflow segundo a WfMC	20
2.2.1 Processo	20
2.2.2 Participante	21
2.2.3 Papel	21
2.2.4 Atividade	21
2.2.5 Instância.....	21
2.2.6 Tarefa.....	21
2.2.7 Lista de tarefas.....	22
2.2.8 <i>Deadline</i>	22
2.2.9 Sincronismo	22
2.2.10 Evento.....	22
2.2.11 Transição	22
2.2.12 Gatilho (<i>Trigger</i>)	22
2.3 Técnicas de modelagem de workflow e o modelo de referência da WfMC	23
2.3.1 Modelo de Referência da WfMC.....	23
2.3.2 Modelo de Gatilhos	26
2.3.3 ICN – Information Control Net	28
2.3.4 Modelo de Ceri/Casati	29
2.3.4.1 Definições do modelo Ceri/Casati.....	30
2.3.4.2 Representação do modelo Ceri/Casati.....	30
2.3.5 Redes de Petri	31
2.3.5.1 Definições do modelo de Redes de Petri.....	32
2.4 Oracle Workflow Builder	33
2.5 Elementos encontrados na modelagem de workflow	35
2.6 Comparação entre as técnicas e definições de Workflow.....	35

2.6.1 Entendendo o metamodelo:	41
2.7 Considerações sobre modelagem de workflow	41
3 Documentos estruturados e modelagem de documentos... 42	
3.1 Documentos Estruturados	42
3.1.1 Padrão Composição (" <i>Composite Pattern</i> ").....	43
3.2 Hiperdocumentos	44
3.3 Modelagem de Hiperdocumentos	45
3.3.1 O modelo DTD	46
3.3.2 O modelo ODA.....	46
3.3.3 A SGML	47
3.3.4 Os formatos : HTML e XML	47
3.4 Outros padrões de modelagem de documentos	48
3.4.1 EDMS 98	49
3.4.2 GDOC	49
3.4.3 OHypA	50
3.4.4 PSM	50
3.4.5 TSIMMIS/OEM	50
3.5 Características importantes para modelagem de documentos	51
3.6 Proposta de metodologia de desenvolvimento de sistemas de informação para Web	51
3.7 Documentos e a Web.....	52
4 Modelagem de Workflow com Diagrama de Atividades da UML.....	55
4.1 Modelagem orientada a objetos padrão – UML (Unified Modeling Language) e as necessidades de extensão	55
4.2 Características agregadas ao diagrama de atividades da UML segundo técnicas de modelagem de workflow	56
4.2.1 O diagrama de atividades da UML.....	56
4.2.2 Elementos de modelagem de <i>workflow</i> que não se encontram no diagrama de atividades da UML	58
4.3 Considerações sobre a extensão da modelagem de workflow.....	60
5 A Integração da Visão Estrutural dos Hiperdocumentos com a Visão Dinâmica de Workflow como Mecanismo de Modelagem de Sistemas de Informação na Internet	62
5.1 A UML (Unified Modeling Language)	63
5.2 A proposta de ciclo de vida de sistemas de informação na Internet	65
5.3 A UML e as extensões propostas.....	67
5.3.1 Diagrama de estrutura de documentos preliminar	67
5.3.2 Diagrama de atividades	70
5.3.3 Diagrama de restrições de acesso em casos de uso para documentos	74
5.3.4 Diagrama de contextos de navegação.....	75

5.3.5 Diagrama de classes persistentes e banco de dados	77
5.3.6 Diagrama de classes de programação	77
5.3.7 Diagrama de documento final.....	78
5.4 Lista de estereótipos adotados com a extensão da UML.....	79
5.4.1 Estereótipos do diagrama de estrutura de documento (preliminar e final).....	79
5.4.2 Estereótipos do diagrama de atividades	81
5.5 Considerações finais sobre a proposta de extensão	82
5.5.1 Sobre a proposta da extensão	82
5.5.2 Sobre o <i>workflow</i> de atividades e seus tratamentos.....	83
5.5.3 Considerações sobre a modelagem.....	83
5.5.4 Regras de consistências do modelo final.....	84
5.5.5 Sobre a continuidade da proposta de extensão de Flávio Lima.....	84
6 Experimentação	86
6.1 Diagrama de casos de uso	87
6.2 Diagrama de estrutura de documentos preliminar.....	89
6.3 Diagrama de atividades	92
6.4 Diagrama de restrições de acesso em casos de uso para documentos	93
6.5 Diagrama de contextos de navegação.....	93
6.6 Diagrama de classes persistentes e banco de dados	94
6.7 Diagrama de classes de programação	96
6.8 Diagrama de estrutura de documento final.....	98
7 Conclusão	99
7.1 Características dos temas abordados no trabalho	100
7.2 Contribuições do trabalho	101
7.3 Sugestões para trabalhos futuros.....	102
Bibliografia	103

Lista de Abreviaturas e Siglas

ASP	Active Server Pages
CGI	Common Gateway Interface
CRC	Classes, Responsibilities and Collaborators
GUI	Graphics User Interface
DTD	Document Type Definition
EDI	Electronic Document Interchange
EDMS	Enterprise Document Management System
GDOC	Gerência de Documentos
HTML	HyperText Markup Language
ICN	Information Control Net
IDC	International Data Corporation
MIT/LCS	Massachusetts Institute of Technology, Laboratory for Computer Science
ODA	Office Document Architecture
ODIF	Office Document Interchange Formatted
ODL	Office Document Language
PSM	Products and Services Marketing
SGML	Standard Generalized Markup Language
TMO	Técnica de Modelagem orientada a Objetos
UML	Unified Modeling Language
URL	Universal Resource Locator
W3C	World Wide Web Consortium
WFDL	Workflow Data Language
WfMC	Workflow Manager Coalition
Wf-XML	Workflow Extensible Markup Language
WFMS	Workflow Management System
WIS	Web Information System
WWW	World Wide Web
XML	Extensible Markup Language

Lista de Figuras

FIGURA 2.1 -	Metamodelo da definição de um workflow [WMC 04].....	21
FIGURA 2.2 -	Modelo de Referência da WfMC – Componentes e Interfaces [WMC 99].....	27
FIGURA 2.3 -	Esquema do modelo de interoperabilidade da WfMC [BAM 97].....	27
FIGURA 2.4 -	Representação gráfica do modelo de Gatilhos [AMA 97].....	29
FIGURA 2.5 -	Representação gráfica do modelo ICn [AMA 97].....	31
FIGURA 2.6 -	Representação gráfica do modelo Ceri/Casati [AMA 97].....	33
FIGURA 2.7 -	Representação gráfica do modelo de Redes de Petri [AMA 97].....	35
FIGURA 2.8 -	Metamodelo de definições de <i>workflow</i> segundo as modelagens e padronizações pesquisadas.....	42
FIGURA 3.1 -	Padrão Composite [GAM 95].....	43
FIGURA 3.2 -	Metamodelo com as definições dos elementos necessários para modelagem de hiperdocumentos.....	54
FIGURA 4.1 -	<i>Swimlanes</i> - diagrama de atividades da UML.....	57
FIGURA 4.2 -	<i>Swimlanes</i> - diagrama de atividades da UML estendido.....	60
FIGURA 5.2 -	Relacionamentos entre o diagrama de casos de uso com o diagrama de atividades estendido e diagrama de documentos.....	66
FIGURA 5.3 -	Relacionamentos entre o diagrama de atividades estendido, diagrama de documentos, restrições de acesso de caso de uso e contextos de navegação.....	66
FIGURA 5.4 -	Exemplificação da utilização de um diagrama de estrutura de documentos preliminar.....	68
FIGURA 5.5 -	Esquema do diagrama de atividades estendido.....	72
FIGURA 5.6 -	Exemplificação do diagrama de atividades.....	73
FIGURA 5.7 -	Esquema do diagrama de restrições de acesso em casos de uso para documentos.....	74
FIGURA 5.8 -	Exemplificação do diagrama de restrições de acesso em casos de uso para documentos.....	75

FIGURA 5.9 -	Esquema do diagrama de contextos de navegação.....	76
FIGURA 5.10 -	Exemplificação do diagrama de contextos de navegação.....	76
FIGURA 5.11 -	Exemplificação do diagrama de estrutura de documento final.....	78
FIGURA 5.12 -	Representação de estereótipos em diagramas de classes.....	79
FIGURA 6.1 -	Diagrama de casos de uso de um tribunal de justiça(1).....	87
FIGURA 6.2 -	Diagrama de casos de uso de um tribunal de justiça(2).....	88
FIGURA 6.3 -	Diagrama de estrutura de documento preliminar de um tribunal de justiça.....	90
FIGURA 6.4 -	Diagrama de atividades par ao início de um processo no tribunal de justiça.....	91
FIGURA 6.5 -	Diagrama de restrições de acesso em casos de uso para documentos de um tribunal de justiça.....	92
FIGURA 6.6 -	Diagrama de contextos de navegação para as restrições de acesssos de caso de uso Selecciona-Modelo-Petição e Cria-Petição.....	93
FIGURA 6.7 -	Diagrama de Sistemas Legado que gerencia as partes do processo.....	94
FIGURA 6.8 -	Diagrama de classes persistentes do sistema de um tribunal de justiça.....	95
FIGURA 6.9 -	Diagrama de classes de programação de um tribunal de justiça.....	96
FIGURA 6.10 -	Esquema do diagrama de documento final de um tribunal de justiça.....	97

Lista de Tabelas

TABELA 2.1 - Elementos encontrados na modelagem de workflow	35
TABELA 3.1 - Participantes do metamodelo proposto por Gama.....	44
TABELA 4.1 - Comparativo de elementos encontrados na modelagem de workflow, padronização e no diagrama de atividades da UML.....	58

Resumo

A construção de aplicações *Web* é algo complexo, pois envolve requisitos não previstos na modelagem de sistemas convencionais. O ambiente *Web* agrega características como: navegabilidade, estruturação de hiperdocumentos, *workflow*, acessibilidade, escalabilidade, usabilidade, compatibilidade, interoperabilidade, segurança, aspectos culturais, entre outros.

O presente trabalho propõe que a modelagem de sistemas de informação na *Internet* incorpore características existentes na modelagem de sistemas de gerência de *workflow* manipulando hiperdocumentos.

A notação utilizada é uma extensão à notação da UML (*Unified Modeling Language*). Esta consiste de novos estereótipos que modelem hiperdocumentos e sistemas de gerência de *workflow* em sistemas de informação na *Web*. A extensão propõe a utilização de diagrama de atividades estendido para modelar o *workflow* de atividades, o qual manipula hiperdocumentos envolvidos no domínio de uma aplicação *Web*. Os hiperdocumentos são modelados através de diagramas de estrutura de documentos, que são derivados do diagrama de classes da UML. A navegabilidade dos documentos é desenhada através dos diagramas de contexto de navegação e restrições de acessos de em casos de uso. Finalizando a modelagem, o diagrama de estrutura de documentos recebe todas as características dinâmicas, estáticas e de ligação que são modeladas através dos diagramas estendidos bem como pelos diagramas comportamentais e estáticos da UML.

A modelagem segue a notação da UML, sendo sugerido que numa primeira etapa, modele-se os hiperdocumentos e sua estruturação preliminar. Na seqüência, com auxílio do diagrama de casos de uso, modela-se o fluxo de atividades através do diagrama de atividades estendido. Após este levantamento, tanto de documentos quanto de atividades envolvidas no domínio da aplicação, modela-se o contexto de navegação e restrições em acessos de casos de uso, que evidenciam os componentes de ligação necessários à navegabilidade entre os hiperdocumentos.

Através da extensão proposta à notação da UML, que considera aspectos de *workflow*, estruturação de hiperdocumentos e navegabilidade, as fases de projeto e implementação de software, sofrem menos alterações e interferências de adaptação às características das ferramentas de implementação para este novo ambiente.

Palavras-Chave: Sistema de Informação na *Web*, UML, *workflow*, hiperdocumento, metamodelo, modelagem.

TITLE: “Internet Information Systems Modelling how Workflow Systems with Structured Documents”

Abstract

Implementation of Web applications is a complex task since it involves requirements not previously accounted for modelling conventional systems. The Web environment possesses characteristics such as ease of navigation, hyperdocument structuring, workflow, access and scale facilities, usability, compatibility and interoperability, safety and cultural aspects amongst others.

The present work proposes that modelling of information systems in the Internet incorporate the existing characteristics of modelling workflow management systems to handle structured hyperdocuments.

The notation used is an extension to the UML (Unified Modelling Language) notation, which consists in new stereotypes for hyperdocument modelling and incorporated workflow in information systems. The extension proposes use of the extended diagram of activities to model the activity workflow which will handle the hyperdocuments involved in the domain of a Web application. The hyperdocuments will be modelled using diagrams of document structure, which are derived from the UML class diagrams. The navigation of documents is designed using the surfing context diagrams and access restriction associated to the use cases. Modelling is finalised by adding to the document structure all the dynamic, static and link characteristics which are modelled through the extended diagrams, as far as the behaviour and static diagrams of the UML.

The modelling procedure follows the UML notation, being suggested that, in a first step, the modelling of a preliminary document structuring. In sequence, using the diagram of use cases, the activity flow is modelled by using the extended diagram of activities. After the research procedures associated to documents and activities related to the application domain, the surfing contexts and access restrictions of the use cases are modelled to evince the link components necessary to the navigation amongst documents.

The use of the proposed extension to the UML notation, which accounts for workflow aspects, hyperdocument structuring and its navigation, will lead to less changes of the project development and software implementation to account for the characteristics of the implementation tools for this new environment.

Keywords: Web information systems, UML, workflow, hyperdocument, metamodel, modelling.

1 Introdução

Com a revolução da comunicação e da troca de informações através da *Internet*, surge um novo enfoque para o desenvolvimento de sistemas. São padrões abertos para desenvolvimento, necessidades de novos mecanismos de segurança e novos paradigmas nas implementações de *software*. [HIL 97].

Antes da existência do ambiente *Web*, os problemas que afligiam o desenvolvimento de *software* eram caracterizados por uma série de perspectivas, tais como: estimativas de prazo e custos, produtividade de desenvolvimento e qualidade do *software*. [PRE 95].

O ambiente *Web* tornou a construção de aplicações ainda mais complexa. Há novos elementos que devem ser observados como: gerenciamento de documentos em múltiplas mídias, mecanismos de navegação sobre informações estruturadas, *links* de materiais relacionados, segurança de dados, autoria e criação de documentos, suporte para regras, responsabilidades e controles de acessos, *Workflow* entre usuários e administradores de sistemas. [BAL 98].

Estes novos conceitos associados às aplicações *Web*, originaram novas classificações de aplicações dentro do processo de desenvolvimento de *software*, compreendidas como:

- informacionais (jornais *online*);
- as interativas (jogos *online*);
- as transacionais (comércio eletrônico);
- *Workflow* (execução de processos);
- os ambientes de trabalho colaborativos (ferramentas de projetos colaborativas);
- e as comunidades *online* (os espaços de Chats e Portais). [GIN 2001].

Além destas novas características agregadas ao processo de desenvolvimento de *software*, houve um aumento significativo na demanda destas aplicações, causando mudanças na forma de se analisar e modelar sistemas de informação. *Sites* como das Olimpíadas de *Sidney* em 2000, Olimpíadas de inverno de *Nagano* e jogos de *Wimbledon* em 1998, foram recordistas em número de acessos por minuto. [GIN 2001]. Para que isto aconteça, a modelagem de aplicação na *Web* vai além das formas convencionais de se fazer análise, levantamento de requisitos, modelagem e projeto de sistemas.

O desenvolvimento de software para *Web* deve garantir uma dinâmica acelerada, observar o crescimento progressivo, as mudanças rápidas dos requisitos, dos conteúdos e das suas funcionalidades durante todo o ciclo de vida. [GIN 2001]. Aspectos como navegabilidade, interoperabilidade, segurança, integridade, *performance*, situações culturais, morais e legais também devem ser consideradas relevantes em todas as etapas da análise e projeto, o que pressupõe mudanças em técnicas e padrões de modelagens.

A concepção de *software* na *Web* utiliza predominantemente o conceito de documentos. Os documentos são modelados de forma diferente das telas *GUI* (*Graphics User Interface*) e de relatórios convencionais.

A engenharia de software não pode ocupar-se somente com questões funcionais e transações sobre banco de dados, mas deve tratar aspectos como engenharia de hipermídia e hipertexto, interação homem-computador, desenvolvimento de interface, indexação de informações, gerenciamento de projetos e projetos gráficos. [GIN 2001].

O desenvolvimento de aplicações para *Web* está em evidência entre fábricas de *software*, consultores e pesquisadores. Aplicações em desuso, projetos em atraso, orçamentos excedidos e funcionalidades não abordadas são alguns aspectos que tornam estas questões preocupantes. Assim como a engenharia de *software* tem o foco no aprimoramento de técnicas de modelagem, para que o tempo de manutenção e erros de implementação sejam minimizados, a engenharia de *software* para *Web*, tem buscado formas de análise e projetos capazes de diminuir a defasagem encontrada entre modelagem e implementação de aplicações *Web*.

Pesquisadores, como Gosechka e Schranz, estão trabalhando na elaboração de metodologias baseadas em orientação a objetos, normalmente utilizando-se a UML (*Unified Modeling Language*), conceitos de CORBA (*Common Object Request Broker Architecture*) e XML (*Extensible Markup Language*) para criação de projetos *Web*. [GIN 2001].

As características de navegação, modelagem e estruturação de hiperdocumentos, componentes dinâmicos e sistemas de gerência de *workflow* manipulando estes hiperdocumentos, diferenciam sistemas de informação na Internet de sistemas de informação convencionais. Uma metodologia de desenvolvimento de sistemas para *Internet* deve ser capaz de contribuir para reforçar o entendimento dos documentos envolvidos no processo, a navegabilidade e as atividades entre os mesmos.

O presente trabalho considera dois enfoques importantes na modelagem: hiperdocumentos e *workflow*. Observando-se que toda aplicação *Web* se caracteriza pela manipulação de hiperdocumentos através do acionamento de tarefas automatizadas, a proposta é que toda aplicação *Web* pode ser vista como um sistema de gerência de *workflow* manipulação de hiperdocumentos.

Atualmente na fase de análise e projeto de sistemas, conceitos de implementação para ambiente *Web* não são considerados. A proposta deste trabalho fundamenta-se na idéia de trazer para o início do ciclo de vida de sistemas todos os aspectos importantes para um bom desempenho e aplicabilidade dos sistemas para *Internet*.

A proposta de modelagem aplicada neste trabalho é a utilização da UML (*Unified Modelling Language*). A UML fornece uma padronização de modelagem de sistemas e possibilidade da extensão de modelos já existentes através do conceito de estereótipos. Permite que a implementação seja feita em qualquer ambiente, inclusive *Web*. Uma contribuição deste trabalho encontra-se nas características dos estudos feitos sobre hiperdocumentos e gerência de *workflow* implementados através dos estereótipos utilizados nos diagramas de classes, de colaboração e nos diagramas de atividades.

Considerando-se que toda aplicação *Web* manipula hiperdocumentos, e que tais hiperdocumentos podem ser acionados por tarefas automatizadas ou não de um *workflow*, a proposta é que toda aplicação *Web* seja vista como um sistema de gerência de *workflow* que manipula hiperdocumentos.

Estes conceitos foram traduzidos em forma de extensão à modelagem de sistemas de informação utilizando-se UML (*Unified Modelling Language*). UML contém uma forma de extensão (estereótipos), capaz de permitir a modelagem destes conceitos para sistemas na *Web*. Os estereótipos (palavras-chaves, símbolos ou outras notações), possibilitam estender a UML para que se adapte aos aspectos focados neste trabalho. Um estereótipo é uma especialização de um elemento da UML, com a definição de uma nova semântica. [RUM 99]. A modelagem de *workflow* e a modelagem de documentos estruturados, podem ser tratados como estereótipos, desde a elicitação de requisitos e análise.

A proposta de extensão à UML para modelagem de sistemas de informação na *Internet* foi validada através de um estudo de caso de um sistema de informação para o Poder Judiciário. Este sistema administra e controla processos judiciais desde a entrada no cartório de distribuição até a sentença final atribuída pelo juiz.

1.1 Objetivos

Este trabalho tem por objetivo principal a aplicação de mecanismos de extensão à notação UML para que a modelagem de sistemas de informação na *Internet* possa incorporar conceitos de gerência de *workflow* manipulando hiperdocumentos.

1.1.1 Objetivos Específicos

- Dar continuidade à proposta de modelagem de aplicações *Web* de Flávio Lima [LIM 2000]., tendo como foco principal a abordagem de *workflow* manipulando hiperdocumentos.
- Incorporar características existentes na modelagem de sistemas de gerência de *workflow* à modelagem de sistemas de informação, tornando-os sistemas de informação ativos. Sistemas de informação ativos agregam em sua concepção atividades automatizadas, sem a necessidade de acionamento de todas as funções por intervenção humana.
- Utilizar diagramas de atividades para modelar o *workflow* de atividades manipulando hiperdocumentos envolvidos no domínio de uma aplicação *Web*.
- Possibilitar a modelagem de várias visões sobre hiperdocumentos através de diagramas de estrutura de documentos, que são uma extensão do diagrama de classes da UML.
- Permitir o detalhamento da modelagem sobre a mesma notação à medida em que se percorre as diversas fases de desenvolvimento.

1.1.2 Resultados e Contribuições da Proposta

- Aplicação de características de ferramentas de implementação para ambientes *Web*, à modelagem de sistemas, através do desenho de diagramas de estruturação de documentos e modelagem de *workflow*, elaborando um modelo de documentos com aspectos de hiperdocumento dinâmicos, estáticos e de ligação. A modelagem de hiperdocumentos é feita através dos diagramas de estrutura de documentos. A modelagem de *workflow* é feita através de diagramas de atividade estendido.
- Possibilitar a modelagem de um sistema de informação capaz de abordar a utilização de um sistema de gerência de *workflow*, fazendo com que todos casos de uso sejam mapeados para o diagrama de atividades possibilitando maior produtividade na etapas de projeto e implementação da aplicação *Web*.
- Modelagem capaz de integrar e avaliar aspectos de navegação, estruturação de documentos e manipulação de atividades sobre hiperdocumentos. Esta modelagem é desenhada através de diagramas de contextos de navegação, restrições de acessos de casos de uso e diagramas de estrutura de documentos.
- Modelagem possibilita a diminuição na falha de projetos desenvolvidos para *Web*, uma vez que as características da nova interface são abordadas desde o início do projeto.

- Modelando fluxo de atividades existente dentro dos cenários envolvidos na modelagem da aplicação e prevendo a navegabilidade entre os hiperdocumentos;
- Trabalhando o comportamento dinâmico da aplicação através do diagrama de atividades e outros diagramas comportamentais da UML.

1.2 Organização do Trabalho

A dissertação está organizada da seguinte maneira:

- Capítulo 2 – Sistema de Gerência de *Workflow*, aborda conceitos, definições e técnicas de modelagem de *workflow*. Define os elementos importantes para modelagem de *workflow*.
- Capítulo 3 - Documentos Estruturados e Modelagem de Documentos, define conceitos de documentos estruturados, padrão “*Composite*”, hiperdocumentos, formatos HTML, XML e define conceitos necessários para modelagem de hiperdocumentos.
- Capítulo 4 – Manipulação de Documentos por Sistema de Gestão de *Workflow* na *Internet*. Descreve uma possível integração entre os sistemas de *workflow* e hiperdocumentos na *Internet*. Aborda quais características serão adotadas para o mapeamento da modelagem de sistemas de informação na *Internet*.
- Capítulo 5 – Extensão da UML – Uma Proposta para Modelagem de Sistemas de Informação na *Internet*, descreve cada diagrama que foi estendido, bem como relaciona todos os estereótipos adotados.
- Capítulo 6 - Estudo de Caso, que utiliza diagramas estendidos com um exemplo sobre um sistema de informação para um Tribunal de Justiça. Relatando toda tramitação de documentos junto aos processos nas várias varas, cartório de distribuição e entre oficiais de justiça. Um exemplo que contém um número grande de atores e documentos sendo manipulados pelo sistema de informação na Internet.
- Capítulo 7 – Conclusão, resume os principais aspectos adotados na extensão proposta, relatando os benefícios de se pensar na nova interface desde o início do projeto.

2 Workflow

A tecnologia de *workflow* permite o controle eficaz e rápido de processos de negócio das organizações. Permite a análise, modelagem, implementação e revisão dos processos organizacionais, reduzindo tempos de execução, tempo de respostas e custos.

O conceito de *workflow* surgiu a partir da noção de processos de sistemas de manufatura e de automação de escritórios. Um *workflow* é definido pelo conjunto de atividades processadas ao mesmo tempo ou não, com uma possível especificação de controle e fluxo de dados entre atividades relacionadas. Pode também descrever tarefas de processos de negócio dentro de um nível conceitual necessário para compreensão, avaliação e reengenharia de um processo de negócio. [GEO 95]. *Workflow* também pode ser conceituado como “a automação total ou parcial de um processo, durante a qual documentos, informações e tarefas são trocadas entre os participantes do processo.”.[WFM 98]

As tarefas de um *workflow* podem ser executadas por um ou mais sistemas de informação, agentes humanos ou pela combinação de ambos. O *workflow* define a ordem de execução e as condições pelas quais cada tarefa é iniciada, sendo capaz de representar a sincronização das tarefas e o fluxo de informações. Outros conceitos relacionados a *workflow* são os conceitos de reengenharia e automatização de processos de negócio e de informação. [BOR 2000].

Para padronização e desenvolvimento de terminologias relacionadas a *workflow* foi criada uma entidade denominada *Workflow Management Coalition – WfMC*. A WfMC foi organizada em 1993 com missão de promover a área de *workflow* através da divulgação da tecnologia e do desenvolvimento de padrões de interoperabilidade de sistemas de *workflow*. [WFM 98]

A próxima seção apresenta detalhes sobre os conceitos envolvidos dentro de um *workflow* com base na WfMC e a representação gráfica através do uso de metamodelos.

2.1 Metamodelo da definição de um workflow

A compreensão de todas as técnicas de modelagens, bem como suas comparações são apresentadas nesta pesquisa através do uso de metamodelos.

Um metamodelo pode ser definido por modelo do modelo, ou seja, é criado para auxiliar no estudo e compreensão das propriedades de uma determinada notação ou

metodologia [PAS 94], além de ser adequado aos estudos comparativos entre metodologias.

A figura 2.1 apresenta os principais conceitos para a definição de processos, relacionamentos e atributos de um sistema de *workflow* segundo a WfMC.

Para WfMC, um processo de *workflow* contém uma descrição textual das informações, data de criação, autor, parâmetros do sistema, prioridades de execução, limites de tempo para checagens, atores notificados e atributos de simulação. .[WFM 98]

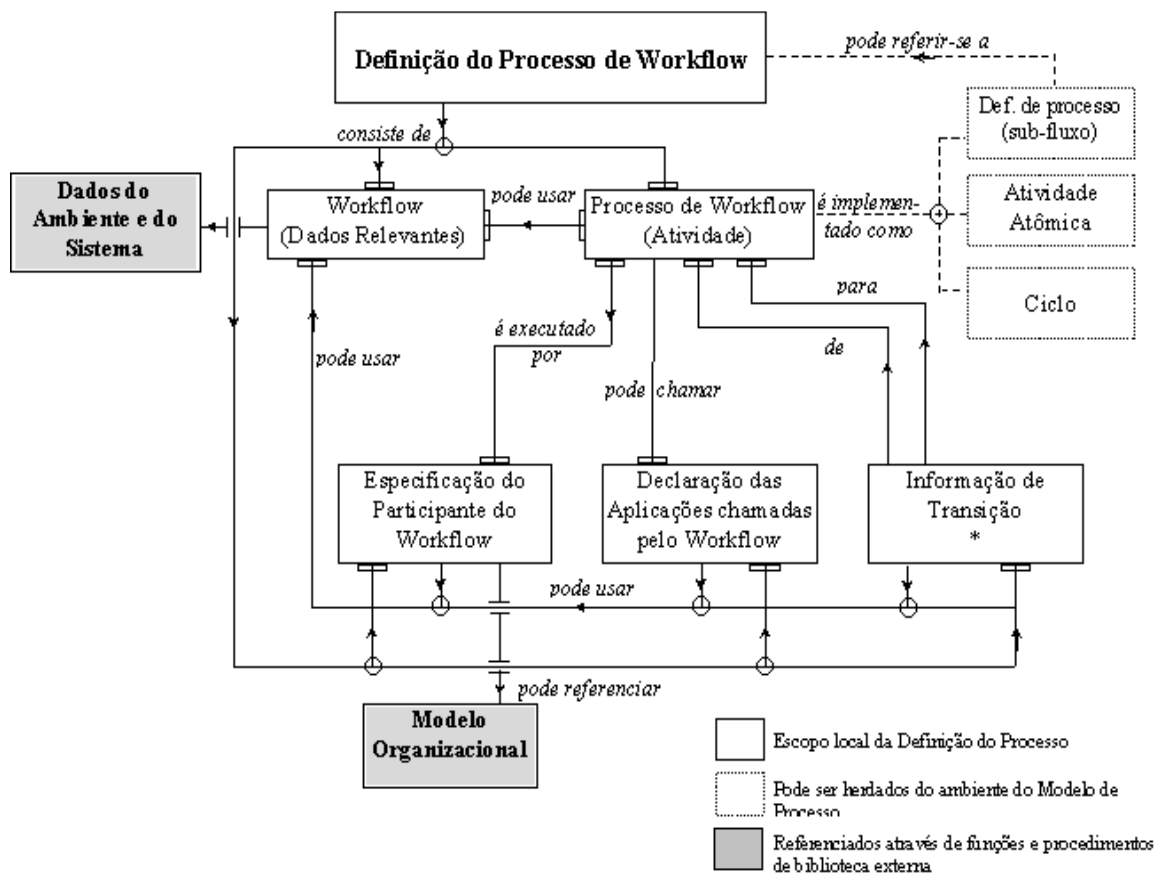


FIGURA 2.1 - Metamodelo da definição de um *workflow* (WfMC, 2004).

Um processo de *workflow* ainda agrupa uma ou mais atividades, cada uma compreendendo uma unidade de tarefa do processo. Todo sistema de *workflow* trabalha com listas de tarefas, e cada atividade representa o trabalho que será executado por uma combinação de recursos e/ou aplicações. Uma atividade pode ser do tipo atômica (normal) ou composta (contém uma série de execuções de uma definição de processo específico – um sub-fluxo). Também pode ser especificada como um ciclo (quando é constituída por várias execuções repetidas). .[WFM 98]

As atividades estão relacionadas a outras através de condições de controle de ciclos. Cada transição possui três propriedades elementares, atividade de origem, atividade de destino e a condição para que a transição seja executada. A transição de uma atividade para outra pode ser condicional ou não-condicional. Transições podem gerar operações sequenciais ou paralelas. Um conjunto de atividades está conectado ao ciclo controlador por transições de ativação e terminação. .[WFM 98]

Os participantes da declaração do *workflow* descrevem os recursos que agem sobre as atividades envolvidas na definição do processo.

Todo *workflow* está inserido em um cenário organizacional, sobre o qual ocorrem avaliações de expressões mais complexas.

Dentro deste cenário, há declarações das aplicações chamadas pelo *workflow*. Estas aplicações suportam o serviço de *workflow* e processam as associações das atividades, bem como suas identificações.

Estes conceitos descrevem o ambiente de *workflow* com base na WfMC. Neste ambiente encontram-se elementos utilizados para modelagem e implementação de *workflow*. A seção 2.2 irá detalhar cada um destes elementos, os quais servirão de base para os estudos referentes aos conceitos de *workflow* que serão incorporados em uma modelagem de sistemas para *Web*.

2.2 Conceitos relacionados em um workflow segundo a WfMC

A WfMC é uma organização que objetiva o desenvolvimento de padrões para a estruturação de uma terminologia para ferramentas de gerência de *workflow*. Sua base encontra-se na busca da interoperabilidade entre produtos de *workflow* e aplicações de tecnologia de informação como *e-mail* e gerenciamento de documentos. .[WFM 95]

O documento WfMC-TC-1011 - *Workflow Management Coalition: Terminology & Glossary* .[WFM 95].

contém definições e especificações técnicas sobre sistemas de gerência de *workflow*. Com base neste documento, a seguir são elucidados vários conceitos básicos sobre *workflow*.

2.2.1 Processo

Processo ou processo de negócio, é o conjunto de procedimentos ou atividades relacionados com o objetivo de realizarem uma meta dentro de um contexto organizacional. Dentro deste contexto organizacional há papéis funcionais e relacionamentos. Outros sinônimos são rede de atividades, redes de Petri, modelos ou partes de instruções.

2.2.2 Participante

O participante de um *workflow* ou ator é um recurso que executa o trabalho representado por uma instância de atividade. Os recursos podem ser humano, equipamento, uma regra ou uma unidade organizacional. Outros sinônimos utilizados são ator, agente ou usuário.

2.2.3 Papel

O papel é representado por um conjunto de atores que contém atributos, qualificações ou habilidades que os tornem aptos à execução de uma atividade.

2.2.4 Atividade

Atividade é um conjunto de eventos que ocorrem sob a responsabilidade de um ator. Esta definição permite que uma atividade seja realizada por muitos atores, contanto que um ator seja responsável pela atividade. Uma atividade requer um recurso humano ou um equipamento para execução. Conhecida também por etapa, nó, tarefa, elemento de trabalho, elemento de processo, operação ou instrução.

2.2.5 Instância

Uma instância representa a associação ou inclusão de dados a um determinado estágio de um processo ou atividade. Cada instância representa uma linha de execução de um processo ou de uma atividade, que pode ser controlada independentemente de seu estado interno ou externo.

2.2.6 Tarefa

Uma tarefa representa o trabalho realizado por um participante numa instância de processo dentro do contexto do *workflow*. Normalmente uma atividade gera uma ou mais tarefas, que juntas constituem uma atividade a ser realizada por um usuário. Outros sinônimos utilizados são: trabalho, objeto de trabalho, elemento ou tarefa.

2.2.7 Lista de tarefas

Uma lista de tarefas está associada a um participante do *workflow*, ou a um grupo de participantes de um *workflow*, que podem compartilhar de uma lista comum. Os elementos de uma lista de tarefas constituem a *interface* entre a ferramenta de *workflow* e os participantes. Também conhecido por *To-Do List* e lista de trabalho.

2.2.8 Deadline

Deadline é baseado no tempo de espera que cada regra de atividade ou lista de tarefas precisa para completar seus objetivos.

2.2.9 Sincronismo

Sincronismo é a representação formal da interação das atividades através dos fluxos de controle de um *workflow*. O sincronismo estabelece a dependência entre as atividades e especifica quais tarefas devem ser executadas em paralelo, ou quais podem ser prorrogadas até a conclusão de outra atividade. O sincronismo possui cinco tipos de formas: seqüencial, paralelo, convergente (*join*), divergente (*fork*) e condicional. Também pode ser definido como processamento concorrente ou processamento paralelo.

2.2.10 Evento

Evento é a ocorrência de uma condição particular (pode ser interna ou externa ao sistema de gerência de *workflow*), que causa uma ou mais ações dentro do sistema de gerência de *workflow*. Conhecido também por ação, sinal ou notificação.

2.2.11 Transição

É o ponto de um determinado momento de execução de uma instância de processo. Após este momento, o controle passa para outra atividade da mesma instância do processo. Pode ser considerado também como navegação.

2.2.12 Gatilho (*Trigger*)

Um gatilho permite o disparo de uma atividade a partir de um evento. Ele pode ser visto como uma regra que é avaliada em função da ocorrência de um evento.

Muitos pesquisadores e desenvolvedores de ferramentas de sistemas de gerência de *workflow*, em associação à WfMC, têm proposto e se preocupado com a criação de técnicas de modelagem. Esta busca pela modelagem e implantação de sistemas de gerência de *workflow* se deve a duas necessidades básicas: a conectividade entre sistemas heterogêneos, autônomos, distribuídos e a integração entre aplicações. [GEO 95].

Na próxima seção serão analisadas técnicas de modelagem de *workflow*. Os elementos identificados nas várias técnicas serão utilizados na composição de uma modelagem de *workflow* baseada na UML.

2.3 Técnicas de modelagem de workflow e o modelo de referência da WfMC

Com o objetivo de identificação de outros elementos importantes para uma proposta de extensão ao diagrama de atividades da UML (*Unified Modeling Language*), foram analisados o Modelo de Gatilhos, o Modelo ICN (*Information Control Net*), o Modelo Ceri/Casati e o Modelo de Redes de Petri, no sentido de complementar a idéia da WfMC.

Em tais modelos citados, nota-se que tarefas são visualizadas como uma seqüência de atividades subdivididas em pequenas tarefas. Cada tarefa é compreendida como um conjunto de entradas para produção de um conjunto de saídas. [GEO 95].

Para cada técnica são apresentadas as características mais relevantes, comentários sobre a composição da técnica (seus elementos) e a representação gráfica da modelagem.

2.3.1 Modelo de Referência da WfMC

A WfMC propôs um modelo de interoperabilidade por reconhecer que há vários produtos de gerenciamento de fluxo de trabalho com características em comum. Desta forma, a idéia é permitir que através do uso de padrões possa haver interoperabilidade entre vários produtos do mercado. [WFM 95]

Neste direcionamento a WfMC estabeleceu a identificação de áreas funcionais para o desenvolvimento de especificações apropriadas à implementação de produtos de *workflow*.

Os sistemas de gerência de *workflow* – WfMS, são sistemas que permitem definição, criação e gerência da execução de *workflow* através do uso de software, executando em um ou mais mecanismos de *workflow*, o qual é capaz de interpretar a definição do processo, interagir com os participantes do *workflow* e, quando necessário, invocar ferramentas e aplicações de sistemas de informação.”.[WFM 95].

O modelo WfMC foi concebido para permitir que diferentes ferramentas de modelagem de *workflow* e sistemas de gerência de *workflow* (WfMS) pudessem trocar informações entre si.

A interoperabilidade entre os produtos é obtida através de dois elementos, definidos na norma:

- Um metamodelo contendo as informações que devem ser armazenadas sobre o *workflow*.
- Uma linguagem padrão para interoperabilidade, a WPDL.

O modelo WfMC e a linguagem padrão de interoperabilidade WPDL estão descritas nas figuras 2.2 e 2.3.

A WPDL (*Workflow Process Definition Language*) é uma linguagem que descreve um sistema de *workflow* como um conjunto de caracteres ASCII, utilizando chaves para especificação de objetos, atributos e relacionamentos, e variáveis para especificação de nomes e valores. Ela oferece um número mínimo de entidades pré-definidas, um número mínimo de relações pré-definidas entre as entidades, um número de atributos pré-definidos, atributos adicionais genéricos, relações adicionais adversas e objetos de dados genéricos adicionais.

Possui também duas abordagens para uma definição comum do processo: APIs em tempo de construção (*Build-Time APIs*) para criação de objetos, seus atributos e relacionamentos dos participantes em um *workflow*; e definição de uma linguagem comum para descrição do processo do *workflow*.

Com a opção de linguagem comum para descrição, a WPDL formaliza um padrão de troca comum que suporta a transferência de definições dos processos do *workflow* entre diferentes produtos de *workflow*, permitindo um método comum para acessar e descrever definições de *workflow*. [NIC 98].

Após o levantamento e descrição de cada técnica de modelagem de *workflow*, bem como dos conceitos da WfMC, a próxima seção trará uma análise comparativa entre as modelagens estudadas.

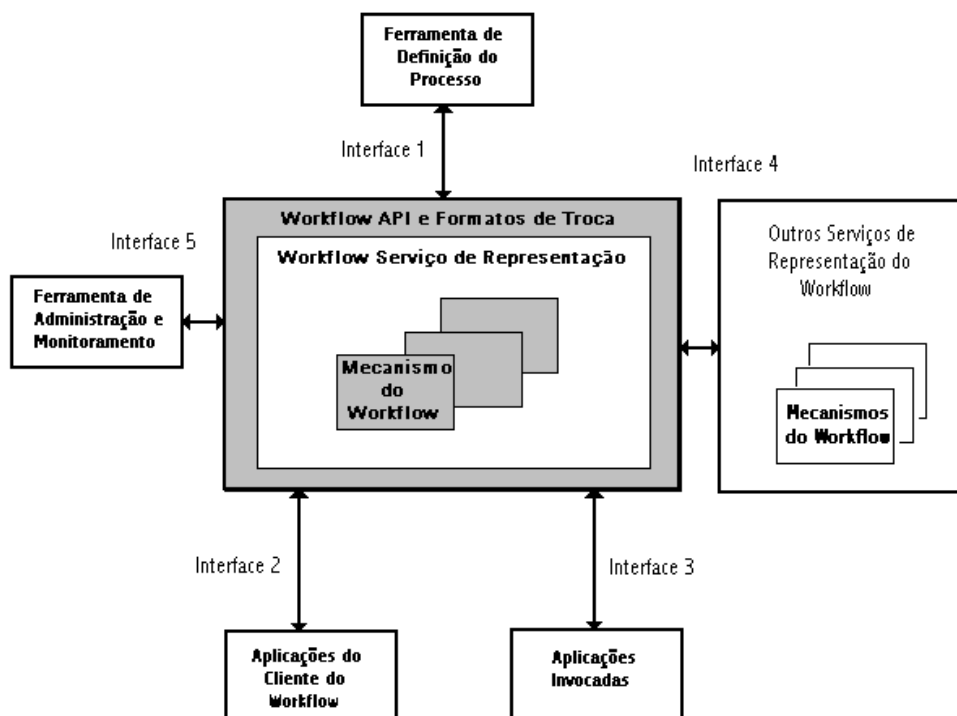


FIGURA 2.2 -Modelo de Referência da WfMC – Componentes e Interfaces .[WFM 95]

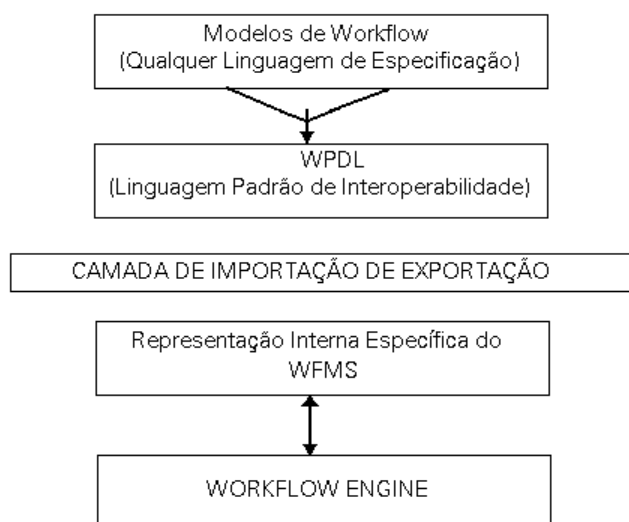


FIGURA 2.3- Esquema do modelo de interoperabilidade da WfMC .[WFM 95]

2.3.2 Modelo de Gatilhos

A técnica de modelagem por gatilhos é resultado de um trabalho de campo desenvolvido pela equipe de Stef Joosten [JOO 95]. Este trabalho consistiu de uma análise em diversas corporações que utilizavam sistemas de *workflow*.

Este modelo descreve o comportamento dinâmico de um sistema através de gatilhos. É considerado um modelo formal e seu nível mais baixo de abstração pode ser convertido para redes de Petri.

O processo de análise de um *workflow* consiste em entrevistas que permitam determinar papéis a serem representados e atividades executadas por estes papéis. As entrevistas estão centradas na identificação das atividades, papéis e gatilhos envolvidos no processo.

1. Definições do modelo de gatilhos

Esta técnica foi definida dentro de uma visão que utiliza conceitos de fácil compreensão, para qual encontram-se os seguintes elementos:

- **Atividade:** uma atividade é basicamente um conjunto de blocos de tarefas de um *workflow*. Toda atividade consiste de eventos.
- **Ator:** qualquer elemento externo que interage com um sistema de *workflow* é denominado ator. Um ator pode ser uma pessoa, uma organização, outro sistema ou um equipamento.
- **Sincronismo Causalidade:** Ligação que permite a execução de uma atividade sucessora, somente com o término da atividade precedente.
- **Sincronismo Fork:** Ligação entre várias atividades que permite a execução das atividades sucessoras após o término de uma atividade precedente.
- **Sincronismo Join:** Ligação entre atividades que exige a execução de várias atividades precedentes para que posteriormente seja feita a execução de uma atividade sucessora.
- **Evento:** um evento é algo que acontece em algum ponto de tempo em uma transição, podendo modificar o estado de um objeto.
- **Evento Pré-condição:** evento que possui uma cláusula que condiciona a execução da próxima atividade.
- **Evento Pós-condição:** evento que possui uma cláusula que promove a execução de uma condição após a execução de uma atividade.
- **Gatilho:** Um gatilho permite o disparo de uma atividade a partir de um evento. Ele pode ser visto como uma regra que é avaliada em função da ocorrência de um evento.

- **Objeto:** pode ser definido com a reunião de regras, procedimentos e informações em um só elemento. É qualquer coisa que possa ser visto, tocado, ou de alguma forma sentido.
- **Papel:** é a participação de um determinado ator dentro de um sistema de *workflow*. Um ator pode estar envolvido em atividades de diferentes tipos, isto pode gerar papéis diferentes para o mesmo ator.
- **Processo:** processo pode ser conceituado como uma coleção de atividades. O conceito de processo é usado para estabelecer a hierarquia em sub-processos.
- **Workflow:** o conceito de *workflow* refere-se a uma coleção de atividades relacionadas entre si por gatilhos disparados por eventos externos.

Uma distinção importante entre atividades e eventos, segundo Joosten [JOO 95], é que uma atividade pode ser associada a uma ligação de tempo, ao passo que um evento ocorre num instante específico de tempo.

Este modelo de gatilhos considera que um evento pode conter uma pré-condição ou uma pós-condição para uma atividade. As atividades se relacionam com outras por gatilhos que são disparados por eventos externos.

2.Representação do modelo de gatilhos

Em um modelo de gatilhos, uma atividade é representada por um retângulo, contendo o nome da atividade. Uma seta aponta para uma atividade que é disparada por eventos. Estes eventos resultam na execução da atividade no outro extremo da linha. O modelo é visualizado em colunas, e cada coluna contém as atividades associadas a um determinado papel. A figura 2.4 é mostra um modelo de gatilhos.

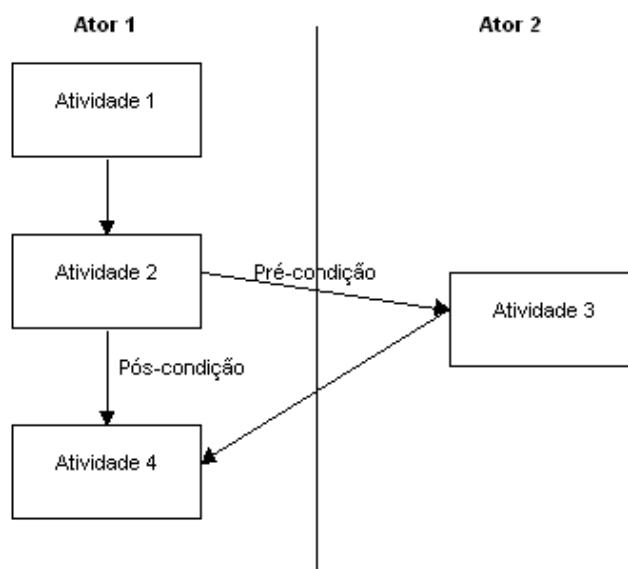


FIGURA 2.4 - Representação gráfica do Modelo de Gatilhos [AMA 97].

2.3.3 ICN – Information Control Net

Este modelo foi idealizado para descrever atividades realizadas em escritórios, sendo uma das primeiras metas dos estudos em automação de escritórios. A fundamentação teórica para este modelo de *workflow* foi proposta pelo Dr. Clarence Ellis na década de 70. [AMA 97] [ELL 80].

O modelo define um escritório como um conjunto de procedimentos relacionados. Cada procedimento consiste em um conjunto de atividades ligadas por restrições de precedência. Para que uma atividade seja executada, o modelo requisita informações provenientes de repositórios, como arquivos e formulários. A idéia desta rede é capturar noções de procedimentos, atividades, precedência e repositórios de forma gráfica.

3. Definições do modelo ICN

O modelo ICN utiliza os seguintes conceitos em seu modelo:

- **Ator:** qualquer elemento externo que interage com um sistema de *workflow* é denominado ator. Um ator pode ser uma pessoa, uma organização, outro sistema ou um equipamento.
- **Atividade:** uma atividade é basicamente um conjunto de blocos de tarefas de um *workflow*. Toda atividade consiste de eventos.
- **Workflow:** o conceito de *workflow* refere-se a uma coleção de procedimentos relacionados para automatizar um escritório.
- **Procedimento:** um procedimento consiste num conjunto de atividades ligadas por elementos denominados restrições de precedência.
- **Documento (Repositório):** são locais de armazenamento como arquivos ou formulários.
- **Paralelismo (nodo de paralelismo):** indica que todas as atividades sucessoras devem ser executadas em paralelo.
- **Evento Pré-condição (nodo de decisão):** indicam que somente uma das atividades sucessoras será executada.

4. Representação do modelo ICN

O diagrama de uma ICN utiliza círculos para representar atividades e quadrados para repositórios, como mostra a figura 2.5. Uma linha ininterrupta de uma atividade “A” para uma atividade “B” representa um arco de precedência e indica que a atividade “B” só pode ser realizada após o término da atividade “A”. Os arcos podem conter nodos sobre si que podem assumir uma decisão (OR), representados por pequenos círculos

preenchidos, ou paralelismo. Nodos de decisão indicam que somente uma atividade sucessora é executada, enquanto que no paralelo, todas as atividades sucessoras são executadas. Linhas pontilhadas das atividades para os repositórios especificam alteração de repositório, enquanto linhas pontilhadas dos repositórios para as atividades indicam leitura do repositório.

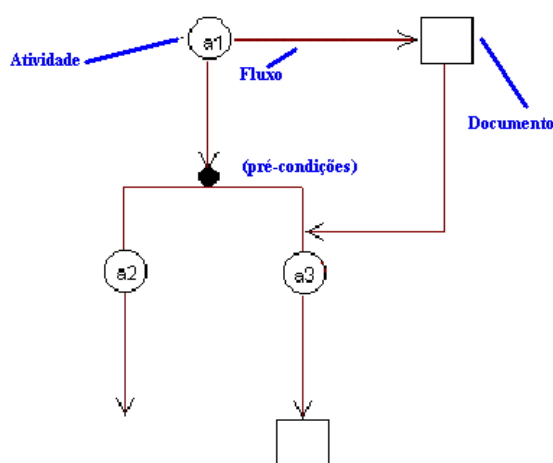


FIGURA 2.5 - Representação gráfica do modelo ICN [AMA 97].

2.3.4 Modelo de Ceri/Casati

O modelo proposto por Casati, Ceri, Pernici e Pozzi [CAS 99], consiste em um conjunto de tarefas com seus relacionamentos. Além de propiciar interação entre um *workflow* e um banco de dados.

As atividades envolvem a execução coordenada de múltiplas tarefas desempenhadas por diferentes entidades processadoras. [CAS 99]. A especificação de um *workflow* envolve a descrição de cada tarefa e das entidades processadoras habilitadas pela execução. Estes aspectos são relevantes para controlar, coordenar a execução e demonstrar a relação entre tarefas.

Este modelo utiliza a combinação de duas linguagens de especificação de *workflow* (WFDL), uma textual e outra gráfica. Estas linguagens são responsáveis por combinar a especificação de *workflows*, com acessos a bancos de dados externos [CAS 99], sendo este último fator, visto como aspecto diferencial neste modelo.

A linguagem de definição textual é composta por símbolos e textos, sendo responsável por especificar e descrever não apenas os mecanismos que são usados para ativar e terminar as tarefas (tanto em situações normais quanto excepcionais) como também especificar detalhadamente cada tarefa a ser executada.

2.3.4.1 Definições do modelo Ceri/Casati

Atividade: uma atividade pode ser entendida como um conjunto de tarefas de um determinado *workflow*.

Ator: entende-se por ator uma pessoa, organização, sistema ou equipamento que interage com um sistema de *workflow*.

Criação de execução: ou instância, representa a associação ou inclusão de dados a um determinado estágio de uma atividade.

Descrição da tarefa: especifica o propósito da atividade.

Evento Pré-condição: evento que possui uma cláusula que condiciona a execução da próxima atividade.

Evento Pós-condição: evento que possui uma cláusula que promove a execução de uma condição após a execução de uma atividade.

Exceção: descreve atitudes que deve ser tomadas na ocorrência de eventos anormais. A exceção também pode restringir a execução de um evento através do tempo e do estado com base nos dados, propiciando término ou cancelamento de uma atividade específica.

Finalização de execução: instância que representa o término de uma atividade.

Multi-atividade: define um conjunto de atividades que realizam exatamente as mesmas operações em paralelo, porém com atores distintos.

Nome da tarefa: nome que representa a identificação de uma tarefa.

Símbolos de início e término: representam o início e o final de uma instância de *workflow*.

Sincronismo: estabelece a dependência entre as atividades e especifica quais tarefas serão executadas em paralelo, seqüencialmente ou concorrentemente.

Super-atividade: representa a decomposição de um *workflow* complexo.

Tarefa: é um elemento de uma atividade. Uma atividade é composta por um conjunto de tarefas.

2.3.4.2 Representação do modelo Ceri/Casati

A figura 2.6 demonstra graficamente a representação da modelagem de Ceri/Casati, bem como a indicação de todos seus elementos.

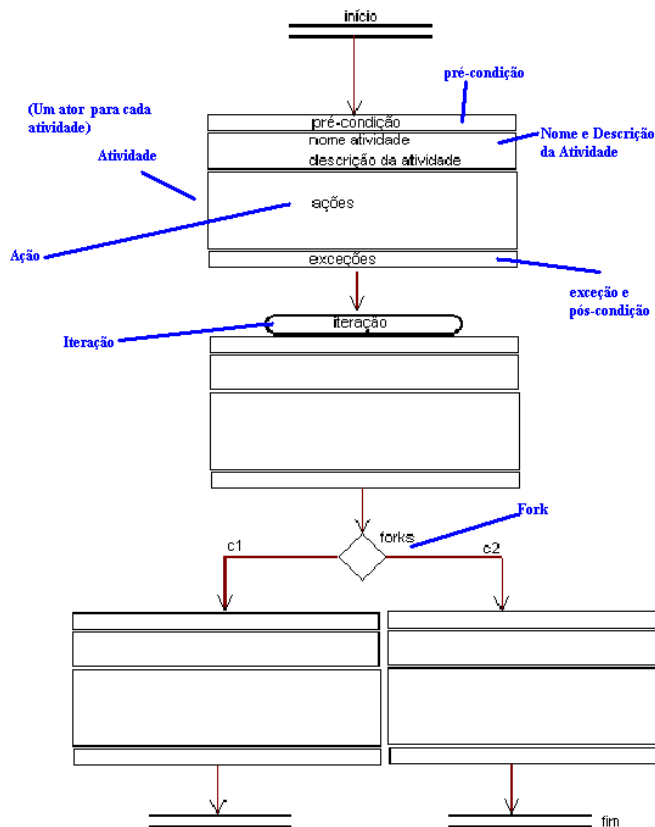


FIGURA 2.6 - Representação gráfica do modelo Ceri/Casati [AMA 97].

2.3.5 Redes de Petri

As redes de Petri são uma ferramenta gráfica e matemática para modelagem de vários tipos de sistemas, as quais apresentam um bom nível de abstração e comparação com outros modelos gráficos. É possível modelar sistemas paralelos, concorrentes, assíncronos e não-determinísticos.

É um modelo do tipo estado-evento, para o qual se tem um evento com pré-condições que permitem a ocorrência e pós-condições decorrentes, que podem ser tornar pré-condições de eventos posteriores.

A aplicação de redes de Petri, como dito anteriormente, é bastante variada, sendo utilizada para modelagem de redes de computadores e protocolos de comunicação, sistemas operacionais, programação paralela, e para qualquer área em que a concorrência seja um fator preponderante. Além da concorrência, o comportamento dinâmico de um sistema também pode ser modelado com redes de Petri.

A grande solidez e o volume do estudo em redes de Petri, tem como resultando inúmeras técnicas de análise extremamente úteis para modelagem de *workflow*. Ex:

técnicas gerais para provar propriedades sobre a rede: *deadlock*, violação de restrições de integridade; cálculo de métricas de *performance*: tempos de resposta, tempos de espera, gargalos; detecção de tarefas que nunca serão executadas; garantia que a instância de *workflow* terá um término em um tempo finito; garantia que, quando a instância termina, os *tokens* referentes à instância são removidos, não havendo outra informação sobre a instância em nenhum *nodo* da rede.

Existe um conjunto de primitivas de *workflow* definidas pela WfMC, que podem ser facilmente modeladas através de Redes de Petri. [AMA 97].

As redes de Petri são constituídas por dois tipos de componentes: um ativo denominado transição e outro passivo denominado lugar. Os lugares correspondem às variáveis de estado e às transições das ações realizadas por um sistema. Estes elementos constituem os vértices do grafo associado às redes de Petri. Os arcos são responsáveis por interligarem lugares às transições e correspondem à relação entre as condições verdadeiras que possibilita a execução das ações. Os arcos que interligam as transições aos lugares representam a relação entre as ações e as condições que se tornam verdadeiras com a execução das ações.

2.3.5.1 Definições do modelo de Redes de Petri

Uma rede de Petri pode ser representada por um grafo orientado com marcação definida por:

- **Atividade:** é um conjunto de tarefas ou de eventos que ocorrem sob a responsabilidade de um ator.
- **Estado (círculo):** modela condição ou estado de um agente e corresponde às condições que devem ser certificadas para os eventos ocorrerem.
- **Evento Pós-condição:** evento que possui uma cláusula que promove a execução de uma condição após a execução de uma atividade.
- **Evento Pré-condição:** evento que possui uma cláusula que condiciona a execução da próxima atividade.
- **Evento:** um evento é algo que acontece em algum ponto de tempo em uma transição, podendo modificar o estado de um objeto.
- **Fluxo (Arco orientado):** interliga um lugar a uma transição ou vice-versa, encadeando condições e eventos.
- **Marca ou Ficha:** representa um recurso disponível ao agente. A evolução da marcação permite modelar o comportamento dinâmico do sistema.
- **Papel:** papel determina o que a atividade deve desempenhar dentro de um sistema de *workflow*. Pode ser desempenhado por um ou mais atores.
- **Sincronismo Causalidade:** Ligação que permite a execução de uma atividade sucessora, somente com o término da atividade precedente.
- **Sincronismo Fork:** Ligação entre várias atividades que permite a execução das atividades sucessoras após o término de uma atividade precedente.

- **Sincronismo Join:** Ligação entre atividades que exige a execução de várias atividades precedentes para que posteriormente seja feita a execução de uma atividade sucessora.
- **Sincronismo:** estabelece a dependência entre as atividades e especifica quais tarefas serão executadas em paralelo, seqüencialmente ou concorrentemente.
- **Transição (retângulo ou barra):** Modela um evento como início de uma operação que corresponde aos eventos que caracterizam as mudanças de estado do sistema.

A figura 2.7 apresenta a representação gráfica do modelo de Redes de Petri. “As Redes de Petri são uma formalização da concorrência e da sincronização de sistemas com atividade distribuída sem utilização de qualquer noção de tempo global [RUM 99].

5.Representação do modelo de Redes de Petri

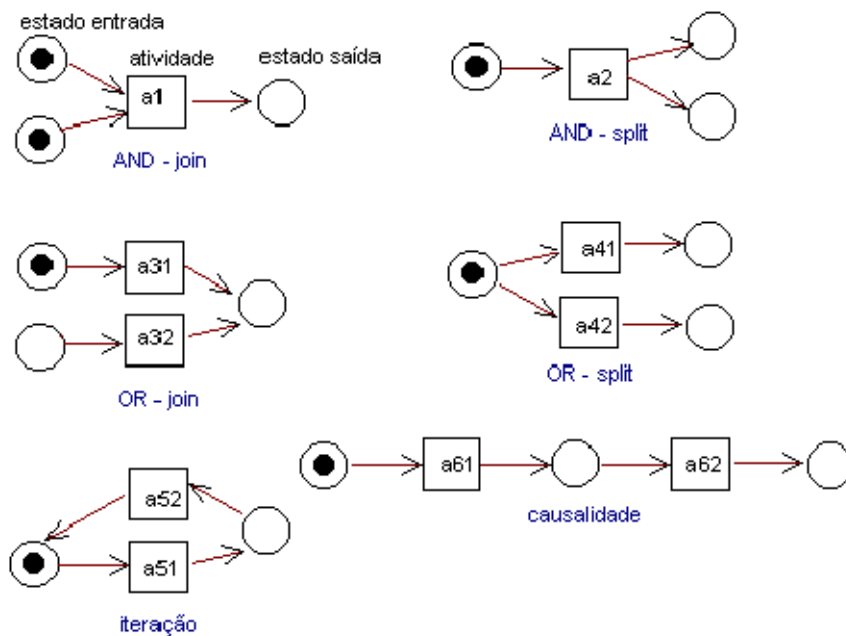


FIGURA 2.7 - Representação gráfica do modelo de Redes de Petri [AMA 97].

2.4 Oracle Workflow Builder

Além das técnicas de modelagem de *workflow* já abordadas, serão apresentados conceitos referentes à ferramenta comercial *Oracle Workflow Builder*. Optou-se por esta ferramenta pela facilidade de acesso que a pesquisadora tem em ambiente profissional.

Oracle Workflow Builder, da *Oracle* é uma ferramenta comercial bastante difundida e utiliza-se dos conceitos descritos na WfMC, bem como de outras técnicas abordadas neste capítulo.

6. Definições do modelo da ferramenta da Oracle

Segundo os manuais de referência da ferramenta *Oracle Workflow Builder*, um *workflow* pode ser projetado graficamente contendo os seguintes elementos:

- **Atividade:** uma atividade é uma unidade de tarefa que contribui para completa realização de um processo. Uma atividade pode ser uma notificação, uma função ou um processo. Uma atividade notificação envia uma mensagem para um usuário do *workflow*. Uma *atividade* função chama uma *stored procedure* PL/SQL ou um programa externo que automatize esta função. Uma atividade processo é o modelo de processo de *workflow* que inclui uma *atividade* de outro processo para representar sub-processos.
- **Ator:** é uma pessoa, sistema ou equipamento que exerce um determinado papel dentro do sistema de *workflow*.
- **Atributo :** *Data, Document, Form, Lookup, number, role, text, URL (Universal Resource Locator)*.
- **Condição:** é um fator que determina a execução ou não execução de uma atividade.
- **Documento:** coleção de dados que são utilizados dentro das instâncias dos processos.
- **Evento:** um evento é algo que acontece em algum ponto de tempo em uma transição, podendo modificar o estado de um objeto.
- **Exceção:** descreve atitudes que deve ser tomadas na ocorrência de eventos anormais. A exceção também pode restringir a execução de um evento através do tempo e do estado com base nos dados, propiciando término ou cancelamento de uma atividade específica.
- **Fluxo de informações:** controle de movimentação exercido sobre os documentos.
- **Função:** função de atividade é uma chamada externa a um programa ou então uma chamada a uma *procedure* do *Oracle*, escrita em PL/SQL.
- **Lookup type:** um *lookup type* é uma lista de valores estática que pode ser referenciada pelas atividades, tipos de itens, mensagens ou atributos de atividades. Enquanto um atributo de mensagem pode referenciar um *lookup type* como resposta a uma lista de possíveis execuções de notificações.
- **Mensagem:** quando se define uma mensagem, o sistema de *workflow* fica aguardando que haja uma resposta com um determinado parâmetro para continuação das próximas atividades.
- **Notificação:** é um mecanismo que informa ao usuário sobre uma ação que deve ser tomada para que o *workflow* possa continuar.
- **Papel:** *Oracle workflow* armazena num diretório de serviço do banco de dados todos os papéis relacionados ao *workflow*.

- **Processo:** um processo de atividades representa uma coleção de atividades num relacionamento específico. Quando um processo de atividade é contido em outro processo, chama-se sub-processo. Em outras palavras, atividades de um processo podem ser processos de si próprios.
- **Regra:** definem quais informações devem transitar pelo fluxo de trabalho e sob que condições.

2.5 Elementos encontrados na modelagem de workflow

Na tabela 2.1 encontram-se listados em ordem alfabética, todos os elementos encontrados nas quatro técnicas analisadas, na WfMC e na ferramenta *Oracle Workflow Builder*.

Estes elementos que participam das várias técnicas estudadas, estão representados no metamodelo que utiliza a notação do diagrama de classes da UML na Figura 2.8.

2.6 Comparação entre as técnicas e definições de Workflow

Um dos resultados obtidos através dos estudos feitos sobre técnicas de modelagem de *workflow* é a representação tabular de todos os elementos encontrados nas várias técnicas. Na tabela 2.1, estes elementos são referenciados na primeira coluna, fazendo-se então um apontamento para quais técnicas os utilizam.

TABELA 2.1 - Elementos encontrados na modelagem de *workflow*

Elemento/Técnica	Técnicas de Modelagem de Workflow , WfMC e Oracle Workflow Builder					
	Gatilho	ICN	Casati	Petri	WfMC	Oracle
Ação/Transição						
Atividade						
Ator/Participante						
Fluxo/Sincronismo/Conexão						
Sincronismo Causalidade						
Sincronismo Fork						
Sincronismo Join						
Condição						
Gatilho de tempo/Deadline/dependência de tempo						

Descrição de Tarefas						
Documento						
Estado						
Evento						
Evento Pós-condição						
Evento Pré-condição						
Exceção						
Finalização de Execução						
Função						
Gatilho/Dependência entre atividades						
Instância/Criação de Execução						
Lista de Tarefas						
Lista de valores						
Marca ou ficha						
Mensagem						
Multi-atividade						
Nome da tarefa						
Notificação						
Objeto						
Papel						
Paralelismo						
Procedimento						
Processo						
Regra						
Símbolos de Início e Fim						
Sincronismo						
Super-atividade						
Tarefa						
Workflow						

Cada elemento contido na tabela 2.1 é definido a seguir:

- **Ação:** ver Transição.
- **Atividade:** é um conjunto de eventos que ocorrem sob a responsabilidade de um ator. Esta definição permite que uma atividade seja realizada por muitos

atores, contanto que um ator seja responsável pela atividade. Uma atividade requer um recurso humano ou um equipamento para execução. Conhecida também por etapa, nó, tarefa, elemento de trabalho, elemento de processo, operação ou instrução. Uma atividade é uma unidade de tarefa que contribui para completa realização de um processo. Uma atividade pode ser uma notificação, uma função ou um processo. Uma atividade notificação envia uma mensagem para um usuário do *workflow*. Uma *atividade* função chama uma *store procedure* PL/SQL ou um programa externo que automatize esta função. Uma atividade processo é o modelo de processo de *workflow* que inclui uma *atividade* de outro processo para representar sub-processos. (Conceito da WfMC com Oracle).

- **Ator (Participante):** O participante de um *workflow* ou ator é um recurso que executa o trabalho representado por uma instância de atividade. Os recursos podem ser humano, equipamento, uma regra ou uma unidade organizacional. Outros sinônimos utilizados são ator, agente ou usuário.
- **Atributo:** *Data, Document, Form, Lookup, number, role, text, URL (Universal Resource Locator).* (Oracle)
- **Condição:** é um fator que determina a execução ou não execução de uma atividade.
- **Criação de execução:** ou instância, representa a associação ou inclusão de dados a um determinado estágio de uma atividade.
- **Deadline** é baseado no tempo de espera que cada regra de atividade ou lista de tarefas precisa para completar seus objetivos.
- **Descrição da tarefa:** especifica o propósito da atividade.
- **Documento (Repositório):** são locais de armazenamento como arquivos ou formulários. É uma coleção de dados que são utilizados dentro das instâncias dos processos.
- **Estado:** modela a condição ou estado de um agente e corresponde às condições que devem ser certificadas para os eventos ocorrerem.
- **Evento:** é a ocorrência de uma condição particular (pode ser interna ou externa ao sistema de gerência de *workflow*), que causa uma ou mais ações dentro do sistema de gerência de *workflow*. Conhecido também por ação, sinal ou notificação.
- **Evento Pós-condição:** evento que possui uma cláusula que promove a execução de uma condição após a execução de uma atividade.
- **Evento Pré-condição (nodo de decisão):** indicam que somente uma das atividades sucessoras será executada.
- **Exceção:** descreve atitudes que deve ser tomadas na ocorrência de eventos anormais. A exceção também pode restringir a execução de um evento através do tempo e do estado com base nos dados, propiciando término ou cancelamento de uma atividade específica.
- **Finalização de execução:** instância que representa o término de uma atividade.
- **Fluxo (Arco orientado) (Conexão):** interliga um lugar a uma transição ou vice-versa, encadeando condições e eventos.
- **Função (atividade tipo função):** função de atividade é uma chamada externa a um programa ou então uma chamada a uma *procedure* do Oracle, escrita em PL/SQL.

- **Gatilho (*Trigger*):** Um gatilho permite o disparo de uma atividade a partir de um evento. Ele pode ser visto como uma regra que é avaliada em função da ocorrência de um evento.
- **Instância:** representa a associação ou inclusão de dados a um determinado estágio de um processo ou atividade. Cada instância representa uma linha de execução de um processo ou de uma atividade, que pode ser controlada independentemente de seu estado interno ou externo.
- **Lista de tarefas:** está associada a um participante do *workflow*, ou a um grupo de participantes de um *workflow*, que podem compartilhar de uma lista comum. Os elementos de uma lista de tarefas constituem a *interface* entre a ferramenta de *workflow* e os participantes. Também conhecido por *To-Do List* e lista de trabalho.
- **Lista de valores:** um *lookup type* é uma lista de valores estática que pode ser referenciada pelas atividades, tipos de itens, mensagens ou atributos de atividades. Enquanto um atributo de mensagem pode referenciar um *lookup type* como resposta a uma lista de possíveis execuções de notificações.
- **Marca ou Ficha:** representa um recurso disponível ao agente. A evolução da marcação permite modelar o comportamento dinâmico do sistema.
- **Mensagem:** quando se define uma mensagem, o sistema de *workflow* fica aguardando que haja uma resposta com um determinado parâmetro para continuação das próximas atividades.
- **Multi-atividade:** define um conjunto de atividades que realizam exatamente as mesmas operações em paralelo, porém com atores distintos.
- **Nome da tarefa:** nome que representa a identificação de uma tarefa.
- **Notificação (atividade tipo notificação):** é um mecanismo que informa ao usuário sobre uma ação que deve ser tomada para que o *workflow* possa continuar.
- **Objeto:** pode ser definido com a reunião de regras, procedimentos e informações em um só elemento.
- **Papel:** é representado por um conjunto de atores que contém atributos, qualificações ou habilidades que os tornem aptos à execução de uma atividade.
- **Paralelismo (nodo de paralelismo):** indica que todas as atividades sucessoras devem ser executadas em paralelo.
- **Procedimento:** um procedimento consiste num conjunto de atividades ligadas por elementos denominados restrições de precedência.
- **Processo ou processo de negócio:** é o conjunto de procedimentos ou atividades relacionados com o objetivo de realizarem uma meta dentro de um contexto organizacional. Dentro deste contexto organizacional há papéis funcionais e relacionamentos. Outros sinônimos são rede de atividades, redes de Petri, modelos ou partes de instruções. Processo pode ser conceituado como uma coleção de atividades. O conceito de processo é usado para estabelecer a hierarquia em subprocessos.
- **Regra:** define quais informações devem transitar pelo fluxo de trabalho e sob que condições.
- **Símbolos de início e término:** representam o início e o final de uma instância de *workflow*.

- **Sincronismo Causalidade:** Ligação que permite a execução de uma atividade sucessora, somente com o término da atividade precedente.
- **Sincronismo Fork:** Ligação entre várias atividades que permite a execução das atividades sucessoras após o término de uma atividade precedente.
- **Sincronismo Join:** Ligação entre atividades que exige a execução de várias atividades precedentes para que posteriormente seja feita a execução de uma atividade sucessora.
- **Sincronismo:** é a representação formal da interação das atividades através dos fluxos de controle de um *workflow*. O sincronismo estabelece a dependência entre as atividades e especifica quais tarefas devem ser executadas em paralelo, ou quais podem ser prorrogadas até a conclusão de outra atividade. O sincronismo possui cinco tipos de formas: seqüencial, paralelo, convergente (*join*), divergente (*fork*) e condicional. Também pode ser definido como processamento concorrente ou processamento paralelo.
- **Super-atividade:** representa a decomposição de um *workflow* complexo.
- **Tarefa:** uma tarefa representa o trabalho realizado por um participante numa instância de processo dentro do contexto do *workflow*. Normalmente uma atividade gera uma ou mais tarefas, que juntas constituem uma atividade a ser realizada por um usuário. Outros sinônimos utilizados são: trabalho, objeto de trabalho, elemento ou tarefa.
- **Transição:** é o ponto de um determinado momento de execução de uma instância de processo. Após este momento, o controle passa para outra atividade da mesma instância do processo. Pode ser considerado também como navegação. Modela um evento como início de uma operação que corresponde aos eventos que caracterizam as mudanças de estado do sistema.
- **Workflow:** o conceito de *workflow* refere-se a uma coleção de atividades relacionadas entre si por gatilhos disparados por eventos externos. Também pode ser conceituado como uma coleção de procedimentos relacionados para automatizar um escritório.

Como auxílio à compreensão dos conceitos encontrados na modelagem de *workflow*, foi elaborado um metamodelo (figura 2.8) apresentando todos os elementos contidos nas várias formas de modelagem de *workflow*. Este metamodelo tem a finalidade de representar a importância de cada elemento contido num modelo de *workflow*, que servirá como base para a proposta de extensão da UML no capítulo 5. Ele foi desenhado como um diagrama de classes, permitindo que seja compreendida a relação entre os vários elementos, aqui representados como classes.

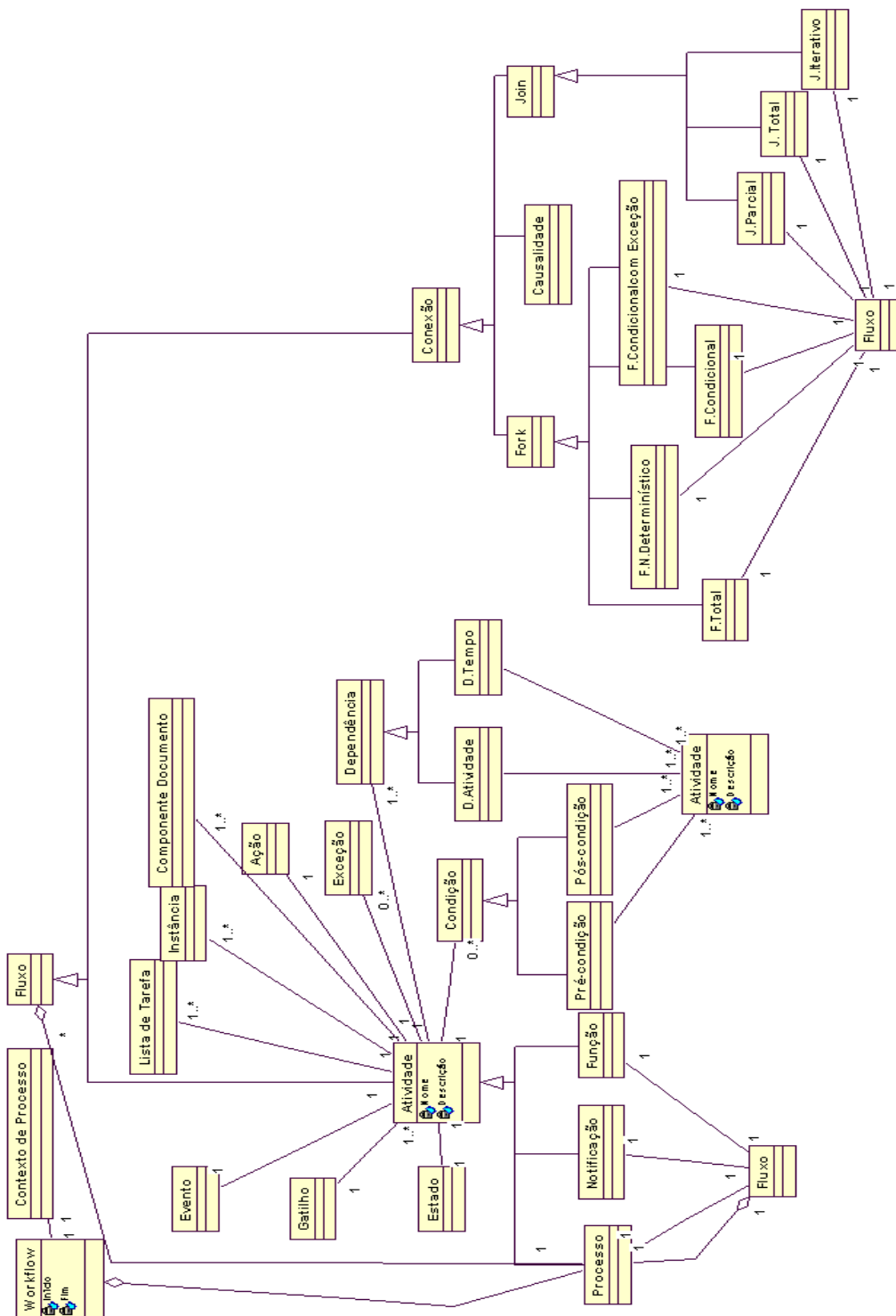


FIGURA 2.8 - Metamodelo de definições de *workflow* segundo as modelagens e padronizações pesquisadas

2.6.1 Entendendo o metamodelo:

Em um sistema de *workflow*, tem-se o próprio *workflow* que tem por atributos início, contexto do processo e fim. O contexto de processo estabelece as fronteiras com os demais sistemas e atores envolvidos no processo.

Cada *workflow* tem seus processos definidos através da classe processo. Todo processo tem agregado a si fluxos, e tais fluxos podem ser categorizados em fluxos de convergente, causalidade e divergente. Ao término de cada fluxo, existem atividades, as quais são definidas com um nome e uma descrição.

Uma atividade pode ser entendida como uma atividade processo, de notificação ou de função. Uma atividade processo é uma atividade que por sua vez será posteriormente detalhada, uma atividade de notificação tem por finalidade apontar alguma situação através de mensagens, e-mails e outras formas de comunicação com os atores envolvidos. Uma atividade função será responsável pela execução de uma rotina automatizada dentro do processo de *workflow*.

Toda atividade está relacionada a um evento e pode ser disparada por um gatilho (que dependa de outra atividade ou de um determinado tempo). As atividades podem ser instanciadas, ter relacionamento com documentos, fazerem parte de uma lista de tarefas e estão sempre associadas a um estado. Além de que toda atividade pode conter exceções no seu modo de execução.

2.7 Considerações sobre modelagem de workflow

O objetivo deste trabalho encontra-se na modelagem comportamental. A proposta do trabalho de Flávio Lima delimitou questões referente às modelagens estrutural e navegacional dos hiperdocumentos.

O estudo das técnicas de modelagem de *workflow*, do padrão da WfMC e de uma ferramenta comercial possibilitaram a avaliação de todos os elementos utilizados por estas. As características encontradas foram incluídas em uma linguagem de modelagem OO através de estereótipos, a UML. [ERI 98] *in apud* [LIM 2000].

3 Documentos estruturados e modelagem de documentos

A *Web* trouxe consigo uma nova visão para modelagem de sistemas de informação, por utilizar principalmente dados não estruturados. São recursos de banco de dados heterogêneos, coleções de estruturas livres, coleções de documentos, conceitos de colaboração e distribuição de documentos, que dão suporte a uma nova forma de se pensar sistemas. [MOR 1998].

Durante muito tempo, a modelagem de sistemas de informação preocupou-se com as funcionalidades e a elaboração de um banco de dados. Isto é visto claramente nas técnicas de modelagem adotadas pela análise estruturada e essencial de sistemas. Os problemas do desenvolvimento concentravam-se na melhoria de padronização e otimização de rotinas. Havia uma demanda exagerada em manutenção corretiva e implementativa de sistemas.

A evolução e o crescimento da *Web* impactaram diretamente em conceitos de modelagem de sistemas de informação. O projeto de sistemas de informação para *Internet* envolve além de funcionalidades e estruturação de banco de dados, conceitos relacionados a documentos, *workflow*, dados livres e dados não estruturados.

Dentro desta nova visão, este capítulo elucida conceitos referentes à modelagem de documentos, os quais já foram avaliados e experimentados em profundidade na dissertação de mestrado de Flávio Azevedo de Lima. [LIM 2000].

3.1 Documentos Estruturados

Documento é uma combinação de elementos gráficos básicos como caracteres, linhas, polígonos entre outras formas. Tais elementos compõem o conteúdo total de informações de um documento. [GAM 95].

Um documento suporta três características básicas, que são:

- Estrutura física, contendo um conjunto de texto e gráficos dentro de linhas, colunas e tabelas;
- Geração e apresentação visual;

- Mapeamento das posições de exibição dos elementos internos da representação. [GAM 95].

Além destas características, há uma questão importante definida por composição recursiva. A composição recursiva é uma técnica de representação da estrutura hierárquica das informações. A composição recursiva auxilia na produção de documentos com elementos gráficos simplificados.

Este novo conceito é melhor discutido na próxima seção que trabalha conceitos do padrão *Composite*.

3.1.1 Padrão Composição (“*Composite Pattern*”)

A composição recursiva não é aplicada somente para estruturação de documentos, mas pode-se representar qualquer estrutura hierárquica complexa. O padrão *composite* captura a essência da composição recursiva dentro da orientação a objetos. [GAM 95].

A idéia básica do padrão *composite* encontra-se na estruturação hierárquica do tipo “todo-parte”, no qual objetos são visualizados dentro de uma estrutura em árvore. A chave para o padrão *composite* é baseada numa classe abstrata, que representa tanto primitivas quanto conteúdos.

A figura 3.1., demonstra graficamente o padrão *composite*, através de um metamodelo baseado em diagrama de classes da UML.

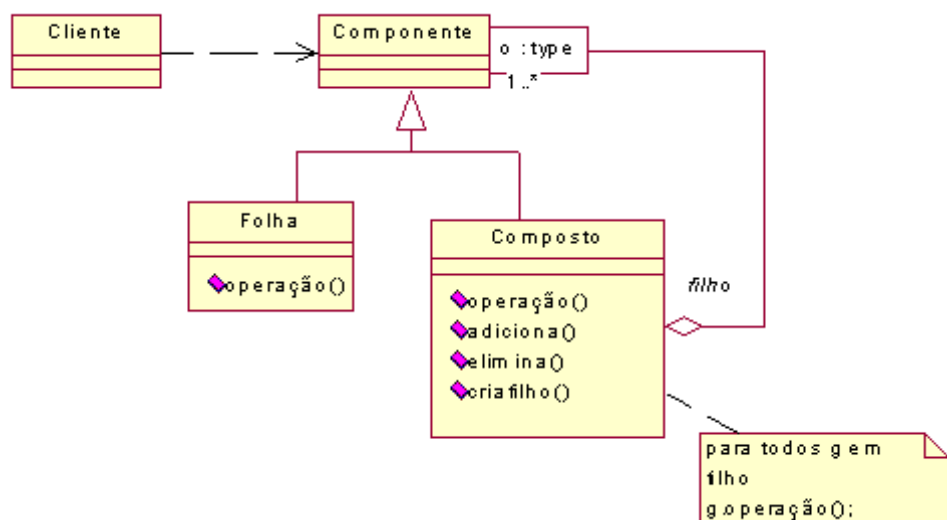


FIGURA 3.1 - Padrão Composite [GAM 95]

Este modelo é definido através da tabela 3.1, conceituando cada participante do metamodelo proposto por Gama.

Tanto documentos estruturados quanto hiperdocumentos abordam conceitos relacionados a textos, imagens, figuras, gráficos, *links*, componentes ativos e navegação.

As informações encontradas em documentos não estruturados (do ponto de vista de desenvolvimento de sistemas), possuem um formato proprietário de armazenamento, impossibilitando a reutilização de informações em outras aplicações. Isto gera uma rápida desatualização dos conteúdos dificultando a migração de tais informações para sistemas mais modernos. [LIM 2000]

A proposta então é o estudo de conceitos que envolvam hiperdocumentos construídos em *HTML (Hypertext Markup Language)* ou *XML (Extensible Markup Language)*. Tais hiperdocumentos são incrementados através da inserção de componentes ativos como *applets*, *scripts*, *CGI's*, *servelets* e *links* de navegação.

TABELA 3.1 - Participantes do metamodelo proposto por Gama [GAM 95]

Participante	Descrição
Componente	<ul style="list-style-type: none"> ▪ Declara a interface dos objetos na composição ▪ Implementa comportamento padrão para interfaces comuns para todas as classes ▪ Declara uma interface para acesso e gerenciamento de todos componentes filho ▪ Define uma interface para acesso a componentes pais dentro da estrutura recursiva, e implementa-a de forma apropriada
Folha	<ul style="list-style-type: none"> ▪ Representa uma folha dentro da composição ▪ Define o comportamento para os objetos primitivos da composição
Composto	<ul style="list-style-type: none"> ▪ Define o comportamento para todos componentes que têm filhos ▪ Amazena os componentes filhos ▪ Implementa operações apresentadas nos componentes filhos dentro da interface <i>Composite</i>
Cliente	<ul style="list-style-type: none"> ▪ Manipula os objetos dentro da composição através da interface <i>Composite</i>

Nas próximas seções serão abordados conceitos referentes à estrutura de hiperdocumentos e as técnicas de modelagens existentes.

3.2 Hiperdocumentos

A principal característica de um hipertexto encontra-se na forma de comunicação com o usuário (ser humano). A *interface* deve ser de fácil assimilação e não provocar dúvidas na sua utilização. As páginas precisam conter informações organizadas, dentro

do limite humano de observação, ter uma aparência agradável, seja na exposição das informações ou nas cores utilizadas.

A origem do termo hipertexto é da década de 60, segundo Bush *in apud* [NEM 95], o qual diz que a mente humana funciona por associações, por isto o significado de texto não linear.

Para Ted Nelson, em [NEM 95], hipertexto é uma combinação de texto em linguagem natural, com características computacionais para saltos interativos, ou apresentações dinâmicas de um texto não-linear, que por sua vez não pode ser impresso de forma convencional numa página.

As características de hipertexto agregadas a capacidade multimídia, ou seja, além de texto, outras mídias como imagem (em movimento ou estática), som, animação e gráficos, deram então origem ao termo hiperdocumento. [NEM 95].

Para compreensão de um hiperdocumento, a melhor alternativa é a divisão de seus elementos em unidades. Estas unidades precisam ser associadas através de conexões. Cada unidade é definida então como “nó” e cada conexão como “elo”. O relacionamento dos elos e dos nós necessitam de um ponto de apoio, definido como âncora.

Surge então a idéia de navegabilidade, a qual permite que o usuário possa visualizar todas as páginas de um hiperdocumento. A navegabilidade define que o usuário tem que estar de alguma forma orientado de onde partiu e para onde quer chegar ou voltar. [W3C 2000]. A navegabilidade além de definir pontos de partida e chegada ao usuário, deve oferecer segurança às informações preenchidas e manter um controle eficiente da navegação, impedindo que usuários não autorizados naveguem sem restrições.

A próxima seção desenvolverá um estudo sobre modelagens de hiperdocumentos.

3.3 Modelagem de Hiperdocumentos

Para promoção da interoperabilidade e administração de fóruns de discussão, foi fundado um comitê internacional denominado W3C (*World Wide Web Consortium*), que pode ser encontrado no site www.w3.org [W3C 2000]. Este consórcio *World Wide Web* foi criado em outubro de 1994, logo após Tim Berners-Lee ter inventando a *Web* no *Massachusetts Institute of Technology, Laboratory for Computer Science* [MIT/LCS]. A idéia principal encontrava-se no início de desenvolvimento de protocolos comuns à WWW, permitindo assim a evolução e garantia da disseminação de padrões *Web*. O W3C tem hoje mais de 400 membros ao redor do mundo e tem ganho reconhecimento

internacional pela sua crescente contribuição para a *Web*. Os principais pontos na missão do W3C são:

- tornar a *Web* acessível a todos os continentes, independente de recursos de hardware, cultura, língua ou tecnologia;
- desenvolver um ambiente de software que permita um melhor uso dos recursos disponibilizados pela *Web*;
- conduzir o desenvolvimento na *Web* para que questões legais, comerciais e sociais sejam respeitadas, aumentando assim o seu uso.

Dentro das responsabilidades que a W3C possui, encontra-se a padronização de modelagem de hiperdocumentos. Esta modelagem pode ser representada através da DTD “*Document Type Definition*”, da “*Office Document Architecture*” - ODA e da “*Standard Generalized Markup Language*” SGML.[W3C 2000].

3.3.1 O modelo DTD

Um DTD define regras para a especificação de uma classe de documentos, tais como:

- Tipos de elementos existentes no documento;
- Atributos destes elementos;
- Instâncias destes elementos hierarquicamente relacionados.

A estrutura especificada em um DTD, segundo sua definição no padrão SGML, propõe que a estrutura lógica de um documento é apenas descrita, não sendo fornecida informação alguma sobre a semântica da apresentação do documento.

3.3.2 O modelo ODA

A ODA faz a modelagem de documentos de forma hierárquica e orientada a objetos. Objetos são os atributos dos documentos. A modelagem separa a estrutura do documento de seu conteúdo, fazendo uma modelagem para o documento e outra para a arquitetura do conteúdo. A estrutura da modelagem ODA é hierárquica e composta por uma estrutura lógica, na qual o documento é subdividido em itens que tenham significado para o usuário, e por uma estrutura de apresentação, a qual representa a visualização do documento. A comunicação de documentos modelados em ODA é feita através dos formatos ODIF – “*Office Document Interchange Formatted*” e ODL – “*Office Document Language*”. [W3C 2000]. Esta modelagem ODA é utilizada para projetos de hiperdocumentos de pequeno porte.

3.3.3 A SGML

A SGML é uma linguagem genérica para notação de instruções de processamento de textos ou simplesmente “*markup*”. Também faz a modelagem dos documentos em formato de árvore como em ODA, porém inclui uma semântica para sua apresentação. Os itens lógicos são chamados de elementos, e cada elemento é delimitado por marcas ou “*tags*” de início e fim. O conteúdo do elemento (existem elementos dentro de elementos) também é delimitado por marcas. O usuário pode criar atributos não definidos em SGML. [W3C 2000].

Os formatos HTML (*HyperText Markup Language*) e XML (*Extensible Markup Language*) tiveram sua origem na SGML. Estes serão descritos no próximo tópico.

3.3.4 Os formatos : HTML e XML

O formato HTML é altamente difundido no ambiente WWW da Internet. Este permite a definição de ligações entre as páginas ou “*homepages*”, tornando o ambiente WWW um hiperdocumento globalizado e distribuído. HTML não é um formato proprietário, apenas possui extensões proprietárias a alguns “*browsers*”. O HTML foi e é atualmente a linguagem de marcação de textos mais difundida, porém a XML (*Extensible Markup Language*), que também é uma linguagem de representação de hiperdocumentos, vem ganhando crescente adesão. [MOR 1998].

Todo documento é constituído por páginas. Um hiperdocumento pode ser comparado a cada seção que é aberta no “*browser*”.

Um documento HTML possui componentes especializados em estáticos, de ligação e dinâmicos.

Os componentes estáticos caracterizam o corpo do documento, formado por cabeçalho, declarações e corpo. O corpo engloba texto, imagem, gráfico, animação, “*frames*” e formulário.

Os componentes de ligação (“*links*”) são responsáveis pelos elos de ligação entre documentos. Todo “*link*” tem um destino (estático ou dinâmico) e é acompanhado por uma âncora.

Os componentes dinâmicos nada mais são que programas, os quais se classificam em *Applets*, *Servelets*, CGI's, ASP, Scripts, *ActiveX*, *ORB*, acesso a banco de dados e acessos a sistemas legados. No caso de *Applets*, normalmente são escritos em Java, e quanto aos CGI's, estes possuem uma gama enorme de linguagens que suportam os conceitos de comunicação entre o servidor *Web* e o servidor de banco de dados. Como exemplos de linguagem para escrita de CGI's encontram-se: C++, *Perl*, *Delphi*. [W3C 2000].

XML é uma metalinguagem definida como um subconjunto de SGML, cujo objetivo é fornecer os benefícios não existentes em HTML e ser mais fácil de utilizar do que SGML. Em XML, é possível a criação de elementos próprios de acordo com a aplicação que está sendo modelada, dando importância ao conteúdo e à estrutura da informação, sem se preocupar com a apresentação. Para que um corretor de validação XML verifique se um documento está correto ou não, ele processa inicialmente seu DTD correspondente, para verificar a estrutura do documento. Há vários corretores de validação (*parsers*) disponíveis gratuitamente no WWW, os quais verificam (análise léxica) e validam (análise sintática) documentos XML de acordo com seu DTD associado. Além desses, há também corretores que não exigem a presença de um DTD e validam somente seus elementos.

Outros benefícios disponíveis no WWW são os editores de documentos XML. Eles permitem a criação simplificada de documentos XML (com ou sem o DTD), possibilitando inserir e eliminar elementos, propriedades e informações associadas, mudar parte da estrutura para outro local e validar o documento. Exemplos de editores para documentos XML são o *Microsoft Notepad XML* e o XED. (marca registrada).

Um documento XML não é responsável pela apresentação das informações, mas pelo conteúdo a ser apresentado. Desta forma um documento XML necessita de *parsers* específicos ou linguagens apropriadas para associação dos conteúdos aos estilos dos documentos em XML. [W3C 2000].

3.4 Outros padrões de modelagem de documentos

O trabalho de dissertação de Flávio Lima teve ênfase em questões de modelagem de documentos. Além das técnicas e padrões de modelagens de hiperdocumento, houve uma preocupação quanto ao estudo sobre gerência de documentos. Conforme Lima [LIM 2000], gerência de documentos é uma área de pesquisa de grande importância e pouco difundida. Mais de 80% da informação das empresas encontra-se em meio eletrônico num formato de documento. A falta de uma política de gerenciamento de documentos leva ao mau aproveitamento dessa grande e valiosa porção de informação de uma empresa.

O problema de uma gerência eficaz de documentos encontra-se em informações expressas em forma não estruturada, armazenamento em formato proprietário, impossibilidade de reutilização de informações em outras aplicações, informações desatualizadas e dificuldade de migração para sistemas mais modernos. [LIM 2000].

Para elucidar algumas características sobre gerência de documentos, conforme Flávio Lima, são listadas e comentadas ferramentas e projetos envolvendo esta área.

3.4.1 EDMS 98

EDMS 98 é um produto comercial para gerência de documentos desenvolvido pela Documentum Inc. Esta ferramenta é especializada em gerência de documentos compostos com estrutura hierárquica. Possui uma arquitetura de três camadas suportando vários bancos de dados, sendo que seus documentos podem ser configurados em XML. Utiliza-se de ferramentas de autoria bastante difundidas como *Microsoft Word e WordPerfect*, bem como de ferramentas de autoria para XML com capacidade de manipulação de elementos em nível mais baixo do que arquivos. [LIM 2000].

As principais características do EDMS 98 são: [LIM 2000]:

- definição de documentos compostos;
- automação do ciclo de vida de documentos organizacionais, incluindo criação/captura, alteração, roteamento, publicação e remoção;
- repositório único para armazenamento, busca e acesso de informação, com capacidade de indexação e perfis;
- integridade de documentos, possibilitando diferentes visões de acordo com o usuário;
- reutilização de documentos, com montagem de novos documentos a partir de outros já existentes;
- disponibilidade de informação em múltiplas plataformas;
- capacidade de manipulação de grandes quantidades de documentos;
- gerência de relacionamentos e *workflow*;
- integração com outras aplicações da organização, como *Microsoft Office, SAP R/3, Lotus Notes* e outras.

3.4.2 GDOC

GDOC é uma ferramenta para armazenamento e autoria de documentos em ambiente de *Internet*. [LIM 2000]. Combina transações sobre banco de dados relacional e HTML com *applets Java* para visualização dos documentos. Este modelo suporta a manipulação de documentos estruturados de forma similar aos modelos ODA e SGML.

O Modelo de Documentos utilizado no GDOC [LIM 2000], é estruturado em três etapas: modelo de esquema, modelo de instâncias e modelo de apresentação. Para a criação de um documento é necessário que o usuário se concentre na edição do conteúdo do documento. [LIM 2000]. Além disto, o GDOC suporta modelagem de *workflow*, relacionando apenas um documento, oferecendo elementos de modelagem para tarefas, decisões e mensagens, e permite que apenas um autor modifique um documento.

3.4.3 OHypA

O metamodelo OHypA modela vários aspectos de documentos estruturados e hipertextos. Baseado no padrão ODA/ODIF (*Office Document Architecture / Office Document Interchange Format*), tem por objetivo auxiliar na modelagem de sistemas de gerência de documentos estruturados com características multimídia. [LIM 2000].

A criação dos documentos exige que se conheça todos os elementos importantes para que a metodologia possa ser utilizada adequadamente.

3.4.4 PSM

O PSM é uma metodologia de desenvolvimento para *Web*, baseada em RMM (técnica de modelagem de sistemas hipermídia integrados com bancos de dados). Na metodologia PSM cada documento pode ter diferentes componentes, o que impossibilita a criação de uma estrutura genérica ou até mesmo um modelo do tipo E-R (entidade-relacionamento).[LIM 2000].

Esta metodologia apresenta sete estágios para o desenvolvimento conforme [LIM 2000]: arquitetura da informação, projeto de interface de usuário e navegação, criação e autoria de conteúdo, gerência de *workflow* e documentos, publicação, revisão de documentos e manipulação de *links*, busca e recuperação.

3.4.5 TSIMMIS/OEM

TSIMMIS é um projeto cujo objetivo é o desenvolvimento de ferramentas para integração de fontes heterogêneas de informação, tanto para dados estruturados quanto para dados não estruturados. [LIM 2000].

Este modelo possui dois elementos importantes: tradutores e mediadores. Os tradutores convertem a lógica dos dados para um modelo comum, enquanto que os mediadores refinam a informação vinda de uma ou mais fontes, processando os resultados enviados pelos tradutores. [LIM 2000].

Este projeto é baseado em arquivos em HTML, exigindo conhecimento da estrutura de *tags* de uma página para definição de *scripts* que percorrem um documento HTML.

3.5 Características importantes para modelagem de documentos

As ferramentas e metodologias anteriormente descritas foram alvo de pesquisa de Flávio Lima em [LIM 2000], no qual são citados pontos importantes para modelagem de documentos, como:

- Documentos virtuais (documentos gerados no momento da consulta do usuário);
- Automação do ciclo de vida de documentos organizacionais (criação, alteração, roteamento, publicação e remoção);
- Repositório único de busca, com capacidade de indexação;
- Possibilidade de diferentes visões de acordo com o usuário;
- Reutilização de documentos, com base em documentos já existentes (reuso de componentes especializados);
- Disponibilidade em várias plataformas;
- Integração com outras ferramentas;
- Modelagem em três partes (esquema, instâncias e apresentação de documentos);
- Arquitetura de conteúdo em três tipos: texto, imagem e gráfico;
- Mecanismos de navegação consistentes;
- Manipulação de grande quantidade de informação não estruturada;
- Controle de versões;
- Representação Orientação a objetos para documentos;
- Documentos ativos e passivos;
- Documentos estruturados compostos por diferentes tipos de componentes;
- Documentos dinâmicos (componentes executáveis);
- Relacionamento entre documentos;
- Portabilidade;
- Ferramentas de busca e recuperação.

3.6 Proposta de metodologia de desenvolvimento de sistemas de informação para Web

A proposta de Flávio Lima para uma metodologia de desenvolvimento em ambiente *Web*, consiste em: [LIM 2000]:

- **“Modelagem de Documentos:** páginas HTML ou XML são vistas como documentos estruturados que contêm componentes estáticos (texto, imagens, vídeo), dinâmicos (*applets* Java, formulários) e de ligação, podendo existir fisicamente na forma de um arquivo ou ser gerados dinamicamente (documentos virtuais). Esses documentos também podem estar envolvidos em um *workflow* organizacional.
- **Modelagem de Bancos de Dados:** dados podem residir tanto em documentos quanto em banco de dados, aumentando e propiciando maior segurança, acesso compartilhado e melhor desempenho.
- **Modelagem Navegacional:** a navegação sobre os documentos do sistema são uma característica inerente de WIS e constituem a interface utilizada pelo usuário, disparando e/ou interagindo com transações e funções. Os caminhos de navegação, ou seja, a estrutura navegacional do sistema, devem ser especificados, bem como os relacionamentos (por exemplo, um documento pode depender de dados de outro documento) existentes entre diferentes tipos de documentos.
- **Modelagem de Transações:** o usuário percebe a funcionalidade do sistema através da navegação sobre seus documentos. É através da navegação que funções são disparadas e acessadas. Portanto, para especificar uma transação, é necessário descrever os documentos envolvidos, caminhos de navegação, classes de usuários, eventos, dados e resultados.
- **Modelagem de Componentes Executáveis:** um WIS é composto de uma série de componentes executáveis interagindo entre si. Cada componente é um conjunto de classes, com atributos e métodos, com uma colaboração bem definida. Cada um desses componentes deve ser inteiramente especificado. Além disso, em um WIS, estes componentes podem residir em servidores ou em clientes.”

Desta forma, Flávio Lima chega à conclusão de que uma metodologia de desenvolvimento para *Web* deve ser capaz de integrar modelagem de documentos, de domínio da aplicação, de banco de dados e de *workflow*.

3.7 Documentos e a Web

Como nos estudos sobre *workflow*, a proposta deste trabalho é estender à modelagem de sistemas de informação na *Internet*, além dos conceitos de *workflow*, conceitos de documentos estruturados. A *Web* disponibiliza funções de gerenciamento de documentos, pois a infraestrutura da *Web* trabalha com uma coleção de objetos.

Para maior compreensão dos conceitos estudados sobre documentos e hiperdocumentos, foi elaborado um metamodelo que encontra-se na figura 3.2. Este metamodelo apresenta elementos que compõem um hiperdocumento denominado componente. Pode-se então dizer que um hiperdocumento pode ser composto por

estáticos, de ligação e dinâmicos. O componente estático representa o corpo do hiperdocumento, composto por texto, imagem, gráfico dentro de uma condição estática. O componente de ligação, possui a função de navegação entre os hiperdocumentos, tendo uma âncora apontando para um componente estático. Os componentes dinâmicos geralmente estão associados ao formulário de um componente estático. Estes têm a função de executar uma operação, seja manipulando banco de dados, arquivos, cálculos, e retornando o resultado sob a forma de componente estático.

Este metamodelo tem a finalidade de representar a importância de cada elemento contido num modelo de estrutura de hiperdocumento, que serve como base para a proposta de extensão da UML do capítulo 5.

A visão de manipulação e modelagem de documentos e hiperdocumentos elucidada conceitos importantes para modelagem de sistemas de informação na *Web*. Uma vez que a *interface* com o usuário é através de páginas *Web*, trazer conceitos de modelagem de hiperdocumentos para o projeto de sistemas é de fundamental importância.

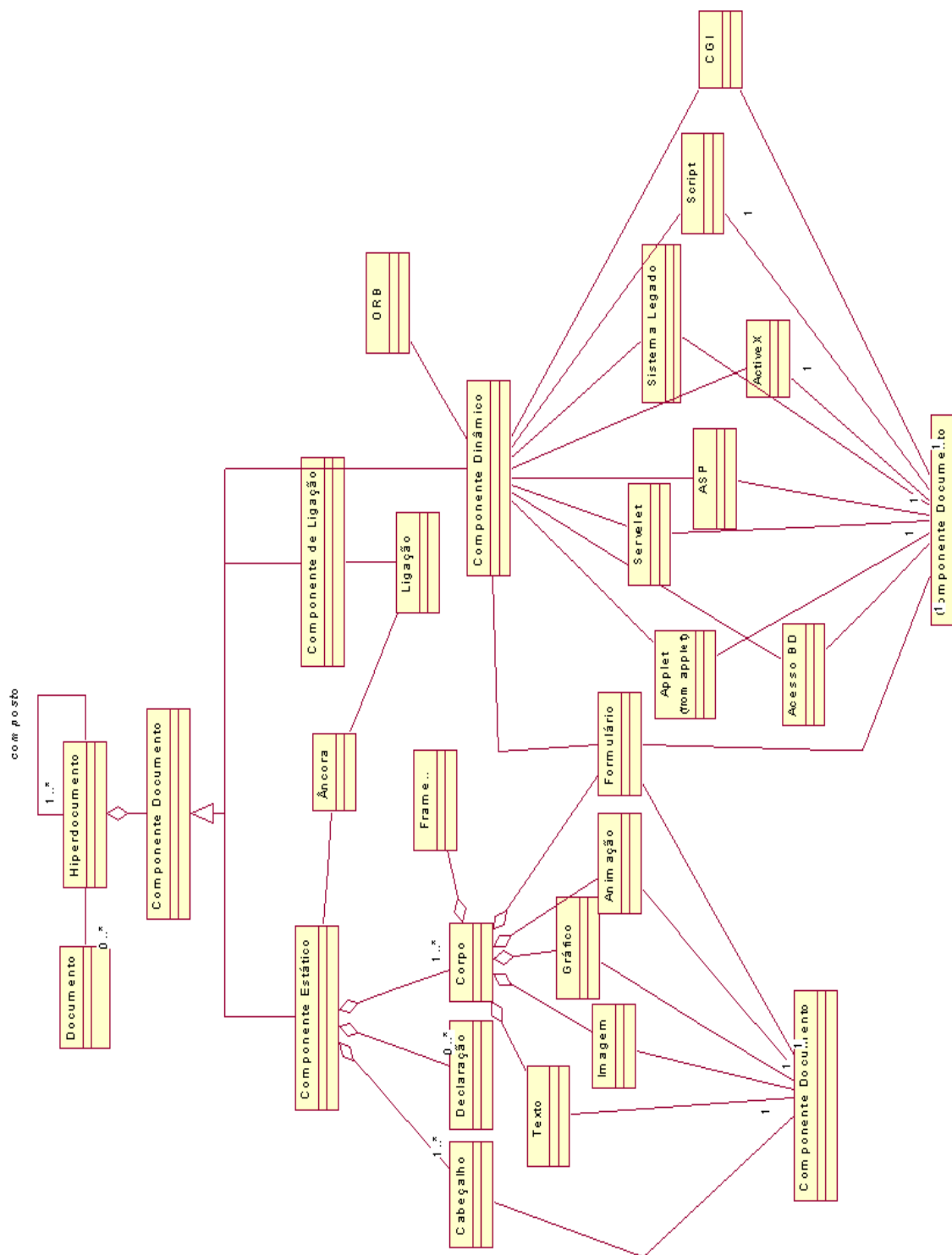


FIGURA 3.2 - Metamodelo com as definições dos elementos necessários para modelagem de hiperdocumentos

4 Modelagem de Workflow com Diagrama de Atividades da UML

O capítulo dois desta pesquisa buscou conceitos sobre *workflow*, técnicas de modelagem, o padrão WfMC e uma ferramenta comercial de gerência de *workflow* com o objetivo de listar e comparar elementos e conceitos sobre como modelar a dinâmica dos processos de negócio. O objetivo deste levantamento é a validação e proposta de uma modelagem dinâmica sobre hiperdocumentos. O capítulo três estabeleceu um relato sobre a área de gerência de documentos e modelagem de hiperdocumentos, elucidando a necessidade de uma modelagem estrutural e de navegabilidade para sistemas de informação na *Web*.

O mapeamento dos conceitos para metamodelos, facilitou a compreensão dos elementos envolvidos nas várias técnicas. Propiciou uma forma sistemática para listar e estender o modelo do diagrama de atividades da UML, conforme será exposto nas próximas seções.

4.1 Modelagem orientada a objetos padrão – UML (Unified Modeling Language) e as necessidades de extensão

O padrão de modelagem de sistemas orientado a objetos mais utilizado atualmente é UML (*Unified Modeling Language*), [FOW 97]. A implementação baseada nesta modelagem produz sistemas com arquitetura de três camadas: camada 1 - *interface* de usuário, camada 2 - lógica de domínio da aplicação e camada 3 - banco de dados, associadas ou não a aplicações legadas e com critérios bem definidos para construção das classes. Isto faz com que os sistemas sejam bem estruturados, eficientes e portáteis.[RUM 99]. Apesar deste padrão ser bastante abrangente em termos de conceitos, a modelagem de hiperdocumentos e de *workflow* não é incorporado à UML de forma explícita.

A extensão proposta utiliza as características de modelagem de hiperdocumentos e de *workflow* para compor as modelagens estruturais, de navegação e dinâmicas para sistemas de informação na *Web*. Sabendo-se que toda interação com o usuário se dá através de hiperdocumentos, a proposta é de uma modelagem estrutural dos documentos juntamente com os requisitos de navegabilidade, observando-se a proposta da W3C. Ou seja, agregando características na forma de comunicação com o usuário, na facilidade de assimilação, na clareza dos limites de observação humana, na utilização adequada de som, animação e principalmente, na idéia de navegabilidade dirigida, definindo pontos de partida e de chegada/retorno ao usuário.

Um projeto modelado em UML pode ser implementado em qualquer plataforma, inclusive na *Web*, porém os hiperdocumentos, que são a *interface* natural com o usuário, não são abordados com todas as suas características necessárias. Isto causa a perda das abordagens de hipermídia, navegabilidade e aspectos dinâmicos. A proposta básica da UML não prevê a modelagem sob a ótica da *Web*, por ser uma padronização aberta para quaisquer tipos de implementações de sistemas.

A idéia central encontra-se na modelagem de sistemas de informação capazes de incorporar conceitos de *workflow* que manipulem hiperdocumentos desde o início da modelagem. E que estes sistemas de *workflow* façam parte do sistema de informação em desenvolvimento. Na próxima seção serão demonstradas as novas características agregadas ao diagrama de atividades da UML.

4.2 Características agregadas ao diagrama de atividades da UML segundo técnicas de modelagem de workflow

4.2.1 O diagrama de atividades da UML

A UML foi criada a partir da cooperação de técnicas e metodologias de Ivar Jacobson, Grady Booch e James Rumbaugh. [RUM 99] Um diagrama de atividades inclui características de modelagem de estados (SDL), diagramas de eventos de Jim Odell e redes de Petri. Este diagrama é muito interessante quando se tem a necessidade de conexão com sistemas de *workflow* ou descrições de comportamento de processamento paralelo [RUM 99].

Uma atividade pode ser executada por um ser humano ou por um computador e pode ser mapeada para um método de uma classe. [FOW 97].

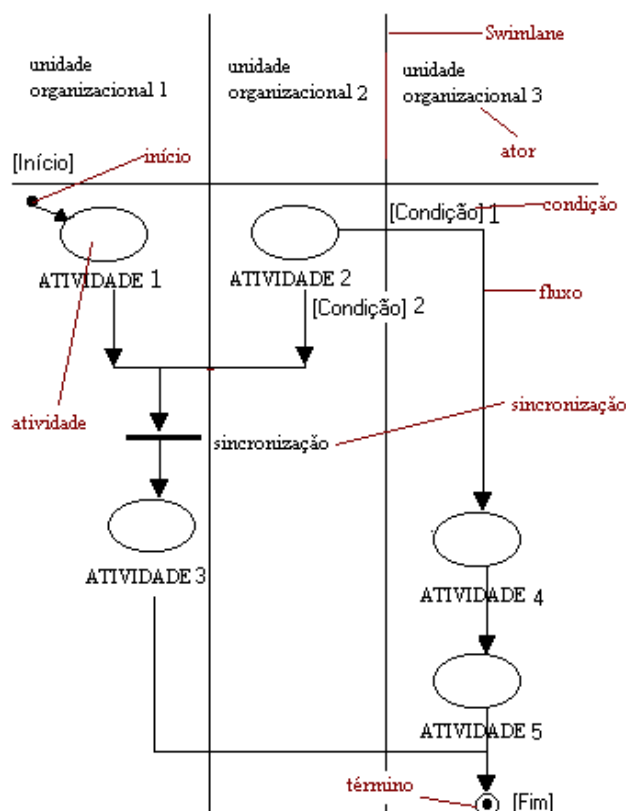


FIGURA 4.1 - *Swimlanes* - diagrama de atividades da UML

O diagrama de atividades é essencial para a modelagem dos fluxos de atividades dos processos de negócio, principalmente quando as atividades necessitam ser executadas em paralelo. Os casos de uso mais complexos são especificados através de um diagrama de atividades. [RUM 99].

As atividades, dentro de um diagrama de atividades, são separadas em regiões denominadas *Swimlanes* (raias). Um *Swimlane* representa uma unidade organizacional desenhada em um diagrama de atividades. [RUM 99] A representação gráfica dos *Swimlanes* pode ser visualizada na figura 4.1. As unidades organizacionais são separadas por linhas verticais (raias).

O diagrama de atividades, conforme a figura 4.1, é composto pelos seguintes elementos:

- **Atividade:** pode ser definida pelo conjunto de eventos ocorridos sob a responsabilidade de um ator.
- **Evento:** é o momento de transição entre uma atividade antecessora e uma atividade sucessora.
- **Ator:** pessoa ou área de negócio pela execução de uma atividade.
- **Fluxo:** define a conexão entre as atividades envolvidas no diagrama.
- **Condição:** define uma regra para que seja executada uma determinada atividade.

- **Dependência entre atividades:** validação que existe para que uma atividade possa ser executada após o término de outra atividade.
- **Swimlanes:** separação por unidade organizacional contendo o diagrama de atividades.

Para suporte à extensão da UML foram avaliadas as técnicas de modelagem de *workflow* estudadas no capítulo 2, bem como a proposta de interoperabilidade da WfMC e os elementos que encontrados na ferramenta de *workflow* da *Oracle*.

4.2.2 Elementos de modelagem de *workflow* que não se encontram no diagrama de atividades da UML

A tabela 2.1 do segundo capítulo desta pesquisa determina os elementos encontrados nas técnicas de Gatilho, ICN, Ceri/Casati, redes de Petri, o padrão WfMC e os elementos da ferramenta de *workflow* da *Oracle*. À tabela 4.1 foram inclusos os elementos encontrados no diagrama de atividades da UML.

Os elementos assinalados com o símbolo (✖) na tabela 4.1, são os elementos que não foram encontrados no diagrama de atividades. Neste comparativo notou-se que o diagrama de atividades da UML possuía os seguintes conceitos:

- Ação;
- Atividade;
- Ator;
- Conexões causalidade, fork e join;
- Gatilho de atividades;
- Estado;
- Evento;
- Fluxo;
- Papel;
- Pré e pós-condições;
- Processo.

TABELA 4.1 - Comparativo de elementos encontrados na modelagem de *workflow*, padronização e no diagrama de atividades da UML

Técnicas de Modelagem de Workflow , WfMC, Oracle Workflow e Diagrama de Atividades da UML							
Elemento/Técnica	Gatilho	ICN	Ceri/Casati	Redes Petri	de WfMC	Oracle	Diagrama de Atividades
Ação			✓				✓
Atividade	✓	✓	✓	✓	✓	✓	✓
Ator	✓				✓	✓	✓
Componente Documento		✓				✓	✗
Conexão – Causalidade			✓		✓		✓
Conexão Fork			✓		✓		✓
Conexão Join			✓		✓	✓	✓
Contexto de Processo							✗
Dependência de tempos					✓		✗
Dependência entre atividades e tempo			✓			✓	✓
Estado				✓	✓	✓	✓
Evento	✓				✓		✓
Exceção			✓			✓	✗
Fluxo		✓			✓		✓
Função						✓	✗
Gatilho	✓				✓	✓	✗
Notificação						✓	✗
Papel	✓				✓		✓
Pós-condição	✓				✓	✓	✓
Pré-condição	✓	✓	✓		✓	✓	✓
Processo	✓					✓	✓
Workflow	✓						✗

Os elementos estereotipados, ou seja, não abordados pelo diagrama de atividades, conforme a tabela 4.2, são:

- Documento;
- Gatilho de tempo;

- Exceção;
- Atividades desmembradas em Notificação, Processo e Função.

Com estas evidências, procedeu-se então à extensão do diagrama de atividades da UML, conforme mostra a figura 4.2.

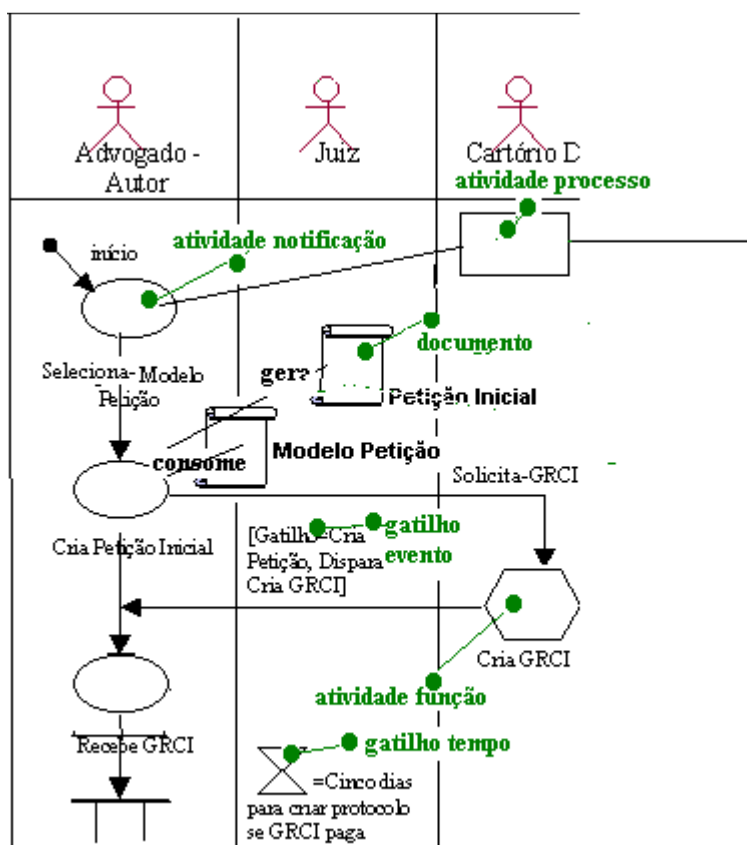


FIGURA 4.2 - Swimlanes - diagrama de atividades da UML estendido

Desta forma, o diagrama de atividades da UML incorpora todos os elementos essenciais para uma modelagem comportamental adequada aos sistemas de informação na *Web*.

4.3 Considerações sobre a extensão da modelagem de workflow

Segundo Marjanovic em [MAR 2000] e Hagen em [HAG 2000], as vantagens e benefícios da utilização de ferramentas de automação de *workflow* são valiosas para área de sistemas de informação, apesar de que estas têm tratado um sistema de

informação apenas como um recurso para o *workflow*. Diante deste fato, este trabalho iniciou o estudo e a tentativa de trazer o sistema de *workflow* para dentro do sistema de informação na *Web*, fazendo com que o sistema de informação seja tratado como um sistema de *workflow* que manipule hiperdocumentos.

Em linhas gerais, esta pesquisa demonstra a necessidade da modelagem comportamental relacionada à modelagem estrutural e navegacional dos hiperdocumentos que fazem o *front-end* com o usuário no ambiente *Web*. Salienta a importância de uma construção de aspectos dinâmicos através de um projeto que trate a estruturação de hiperdocumentos combinados ao processo de *workflow* mapeado.

O próximo capítulo apresenta a proposta integral da modelagem, incluindo extensões aprimoradas do trabalho de dissertação de Flávio Lima, no que se refere ao caso de modelagem de hiperdocumentos incorporada à modelagem de sistemas de informação na *Internet*.

5 A Integração da Visão Estrutural dos Hiperdocumentos com a Visão Dinâmica de Workflow como Mecanismo de Modelagem de Sistemas de Informação na Internet

Os sistemas de informação baseados em conceitos *Web*, vêm sofrendo mudanças tecnológicas e de interação humano-computador com o decorrer do tempo. Isto afeta diretamente aspectos relativos ao levantamento de requisitos, seus conteúdos e suas funcionalidades durante todo processo de ciclo de vida. Novas especificações ocorrem freqüentemente, tornando o desenvolvimento algo ininterrupto. [GIN 2001].

Diante desta evolução, há muitas linhas de pesquisa propondo técnicas e metodologias que permitam a construção de aplicações mais adequadas para o ambiente *Web*. O desenvolvimento de sistemas de informação na *Web* deve prever desde o início da concepção aspectos inerentes : à estrutura de hiperdocumentos, sua navegabilidade, agilidade no acesso simultâneo e contínuo de milhares de usuários, segurança, interoperabilidade, *workflow*, entre outros. [GIN 2001].

O número de aplicações para *Web* tem crescido muito, e por isto a preocupação da elaboração de projetos com características deste novo ambiente. A engenharia de *software* é uma área que busca melhorias contínuas no desenvolvimento de *software*. (PRESSMAN, 1999). Seu principal objetivo é permitir que aplicações tenham maior sucesso na sua implementação, propondo técnicas e metodologias para sua construção.(PRESSMAN, 1999). Nesta mesma linha de raciocínio, a busca por técnicas e metodologias para a construção de aplicações *Web*, tem por foco permitir o sucesso de projetos voltados para *Web*.

A proposta de extensão à notação da UML quer minimizar os problemas relacionados à modelagem de hiperdocumentos e *workflow* em aplicações *Web*. O hiperdocumento pode ser modelado como classes de componentes estáticos, dinâmicos e de ligação. Toda a construção de hiperdocumentos que são utilizados dentro do domínio da aplicação é prevista desde o início da concepção do *software*. Também são considerados aspectos de visões e navegabilidade possíveis para cada ator envolvido no processo de negócio. O *workflow*, modelado através do diagrama de atividades da UML, permite que durante o desenho dos casos de uso, seja projetado em paralelo ou intercaladamente, todo o fluxo de atividades que manipula hiperdocumentos desenhados por um diagrama de classes específico para documentos.

A possibilidade de se considerar as visões de documentos dentro dos cenários do domínio da aplicação e a manipulação dos documentos pelas atividades automatizadas,

garantem que o desenvolvimento de uma aplicação na *Web* contenha requisitos deste ambiente adicionados na fase inicial do projeto.

Considerando-se aplicações *Web* que manipulem hiperdocumentos, de várias estruturas e que tais hiperdocumentos são acionados por tarefas automatizadas de *workflow*, a proposta é que toda aplicação *Web* possa ser vista como um sistema de gerência de *workflow* que manipula hiperdocumentos. Para isto, a modelagem para desenvolvimento de sistemas, deve ser capaz de contribuir para reforçar o entendimento dos hiperdocumentos, sua navegabilidade e a manipulação executada pelo *workflow* destes hiperdocumentos.

A proposta de modelagem será descrita neste capítulo, demonstrando os modelos de representação da UML estendida para a manipulação de *workflow* sobre hiperdocumentos. Esta extensão está baseada na continuidade dos estudos feitos por Flávio de Lima em sua dissertação de mestrado, [LIM 2000], cujo principal enfoque foi gerência de documentos. Além de revalidar modelos já propostos por Lima, o foco de concentração está na modelagem dos aspectos comportamentais de sistemas de informação na *Web*.

5.1 A UML (Unified Modeling Language)

Dentre as técnicas e modelagens de sistemas baseadas em objetos, a UML – *Unified Modeling Language*, ganhou um espaço de destaque na Engenharia de *Software*. Como notação para orientação a objetos, a UML atualmente é a mais difundida, tendo como objetivo principal ser utilizada para qualquer tipo de desenvolvimento de sistemas e em todas as fases de desenvolvimento, segundo Eriksson [ERI 98] in apud [LIM 2000]. A UML é uma linguagem-padrão para a elaboração da estrutura de projetos de *software* e pode ser empregada para visualização gráfica, especificação, construção e documentação de sistemas complexos. [BOO 99].

UML é uma linguagem tida como uma parte de um método de desenvolvimento de *software*. Uma linguagem de modelagem pode ser considerada uma linguagem cujo vocabulário e regras buscam a representação conceitual e física de sistemas. A UML é constituída por três blocos de construção: itens; relacionamentos e diagramas. [BOO 99].

Como a extensão proposta está relacionada ao bloco dos itens, é interessante observá-los para maior compreensão da extensão proposta. Os itens estão divididos em estruturais, comportamentais, agrupamentos e anotacionais.

Os itens estruturais são como substantivos, utilizados para modelar a parte estática do projeto. Representam elementos conceituais físicos e são divididos em sete

tipos: Classes; Interface; Colaborações; Casos de Uso; Classes ativas; Componentes e nós. Os itens comportamentais são como os verbos de um modelo, representando comportamentos no tempo e espaço. São dois tipos de itens comportamentais: iteração e máquina de estados. Semanticamente, tais elementos são conectados a vários elementos estruturais, classes principais, colaborações e objetos. Os itens de agrupamento são partes organizacionais dos modelos de UML, que podem ser decompostos em pacotes. E finalmente, itens anotacionais são partes explicativas para descrever, esclarecer e fazer alguma observação sobre qualquer elemento dos modelos. [BOO 99].

Para extensão proposta, também foram estudados os mecanismos de estensibilidade da UML. UML fornece uma linguagem-padrão para especificação e projetos de sistemas, porém não é fechada para expressar outras realidades pertinentes a qualquer domínio de aplicação. Dentro de seu metamodelo conceitual, ela prevê que a modelagem possa ser ampliada, de maneira controlada, através de elementos que incluam as seguintes características: Estereótipos, valores atribuídos e restrições. Um estereótipo amplia o vocabulário da UML, permitindo a criação de novos tipos de blocos de construção, derivados dos já existentes.[BOO 99].

Estes mecanismos de extensão permitem a definição e ampliação da UML para adaptação a novas tecnologias de *software*, principalmente no que se refere ao surgimento de linguagens de programação distribuídas para *Internet*. [BOO 99].

De acordo com as regras de extensão da UML, a mesma permite a inclusão de novos blocos de construção, modificação dos blocos existentes e alteração da semântica. Isto pode ser feito de forma controlada, permanecendo a extensão fiel ao propósito da UML, ou seja, comunicar as informações de forma clara.

A extensão proposta modificou o diagrama de atividades da UML, incluiu os diagramas de documentos, restrições de acesso em casos de uso e contextos de navegação de Lima [LIM 2000]. Na lista abaixo encontram-se todos os diagramas da UML, destacando-se e comentando-se sobre os diagramas estendidos:

- Diagrama de Casos de Uso;
- Diagrama de Atividades : utilizado em paralelo à construção dos casos de uso. Cada caso de uso se torna uma atividade. Isto facilita a compreensão do *workflow* envolvido no domínio do problema que está sendo modelado. Para cada cenário, fica mais claro quais atividades envolvidas entre os vários atores e os papéis que cada um tem a desempenhar;
- Diagrama de Documentos Preliminar :corresponde ao conjunto de documentos (sejam hiperdocumentos, e-mails, etc.) que será manipulado pelo *workflow* modelado através do diagrama de atividades;
- Diagrama de Restrições de Acesso em Casos de Uso para Documentos: especifica formalmente através de quais contextos cada ator poderá acessar os documentos envolvidos na aplicação;

- Diagrama de Contextos de Navegação : detalha a navegação relevante entre os documentos envolvidos na aplicação;
- Diagrama de classes;
- Diagrama de objetos;
- Diagrama de casos de uso;
- Diagrama de seqüências;
- Diagrama de colaborações;
- Diagrama de Documentos Final corresponde ao conjunto de documentos (sejam hiperdocumentos, e-mails, etc.) que será manipulado pelo workflow modelado através do diagrama de atividades, além de todas as características de navegação e componentes dinâmicos que serão trazidos dos diagramas anteriores);
- Diagrama de grafos de estados;
- Diagrama de componentes;
- Diagrama de implantação.

Para uma visualização macro das extensões propostas, a próxima seção irá tratar do ciclo de vida da construção de *software*, a qual expressará graficamente a nova abordagem na forma de modelagem.

5.2 A proposta de ciclo de vida de sistemas de informação na Internet

O ciclo de vida de sistemas de informação baseados na notação UML, normalmente tem por início a fase de análise de requisitos. Neste momento, são criados os cenários, os casos de usos, podendo-se utilizar os cartões CRC (*Classes, Responsibilities and Collaborators*). São levantadas todas as necessidades de desenvolvimento de um sistema de informação, baseando-se em entrevistas com pessoas que trabalham com o negócio em questão, bem como em documentos manipulados pelas pessoas envolvidas. [RUM 99]

O próximo passo é a fase de projeto orientado a objetos, que possui cinco grupos de diagramas para serem utilizados: diagrama de classes, diagrama de atividades, diagrama de estados, diagrama de seqüência e diagrama de objetos. Os diagramas de classes e objetos representam uma situação estática do sistema, enquanto os demais representam uma condição dinâmica do sistema. [RUM 99]

Feita a análise do sistema, a próxima etapa é a execução do projeto do sistema, auxiliado pelos diagramas de classe, de seqüência, de componentes, de pacotes. Após o projeto, finalmente vem a implementação, codificação e teste do sistema, auxiliados pelo modelo entidade-relacionamento (no caso de implementação em banco de dados relacional), projeto de telas, relatórios e as funcionalidades dos programas. [RUM 99]

Todo processo de levantamento de requisitos envolve o desenho dos cenários, do *workflow* de atividades e dos documentos manipulados pelo *workflow*. Neste momento,

pode-se utilizar os diagramas de casos de uso, o diagrama de atividades e o diagrama de documentos. O diagrama de documentos é utilizado para visualização dos documentos envolvidos no domínio do problema. O diagrama de documentos é um diagrama que mostra o conjunto de documentos, suas composições e seus relacionamentos. O diagrama de atividade, em conjunto com os dois diagramas anteriormente citados, caracteriza todo fluxo de atividades envolvido no domínio do problema. Cada caso de uso, torna-se uma atividade dentro deste diagrama. A medida que o diagrama de atividades vai sendo desenhado, pode-se também definir documentos ainda não projetados no diagrama de documentos.

Durante o projeto destes três diagramas, são enumerados todos os documentos preliminares necessários à modelagem da aplicação. Isto é ilustrado na figura 5.1.

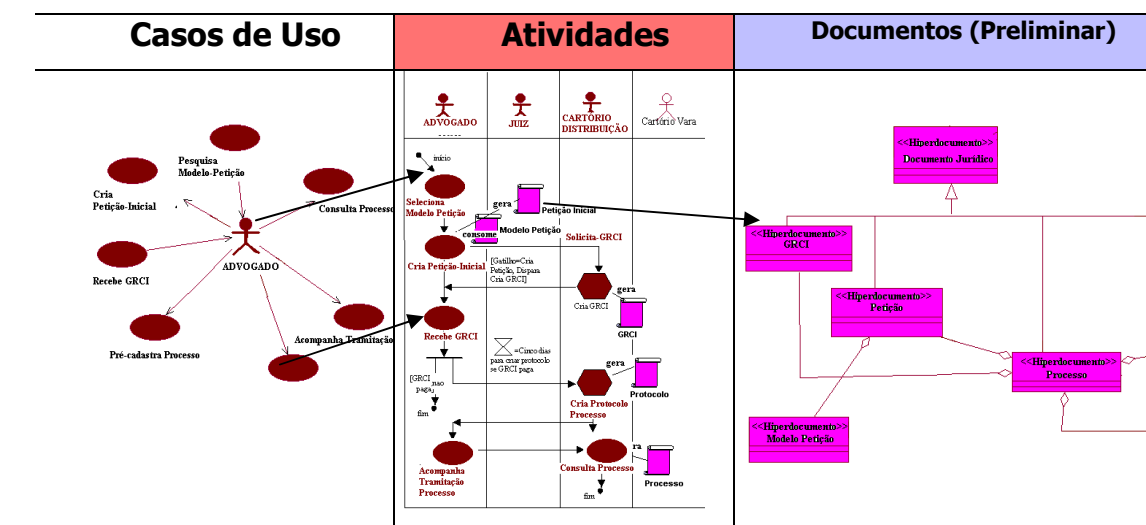
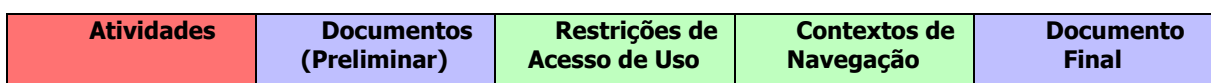


FIGURA 5.2 - Relacionamentos entre o diagrama de casos de uso com o diagrama de atividades estendido e diagrama de documentos

Na modelagem da estrutura dos documentos, os documentos são refinados em três componentes: estáticos, dinâmicos e de ligação. Os componentes estáticos são *frames*, texto, cabeçalho, etc., que caracterizam a interface com o usuário. Os componentes de ligação são responsáveis pela navegabilidade entre os hiperdocumentos envolvidos na aplicação. Os componentes de ligação são modelados através dos diagramas de contextos de navegação e diagrama de restrições em acessos de casos de uso para documentos. Finalmente, os componentes dinâmicos são responsáveis pela execução e funcionalidades do sistema de informação na *Web*. Ou seja, estas funções poderão gerar dinamicamente outros hiperdocumentos, manipular banco de dados, integrar sistemas legados e sistemas de gerência de *workflow*. A figura 5.2 ilustra este refinamento na modelagem dos documentos.



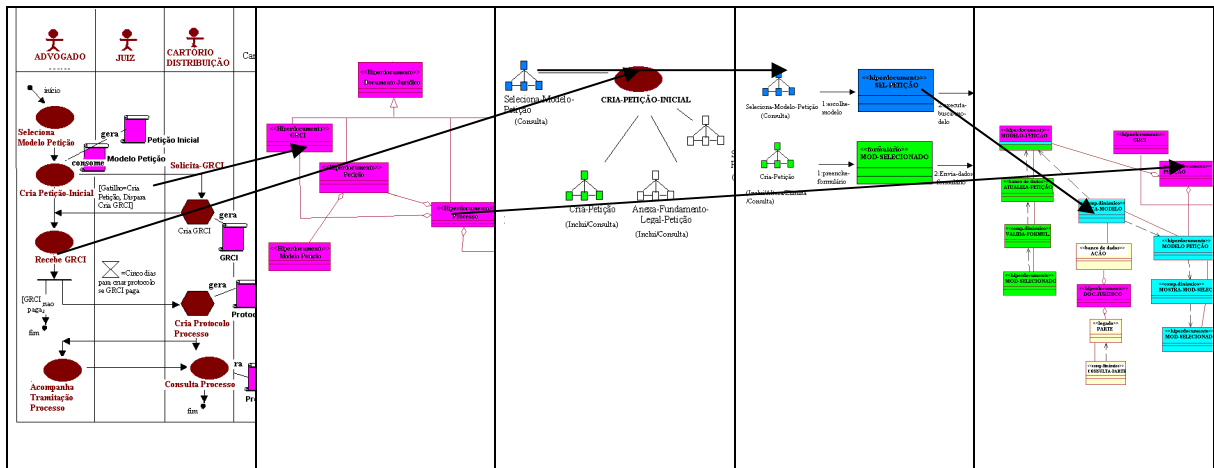


FIGURA 5.3 - Relacionamentos entre o diagrama de atividades estendido, diagrama de documentos, restrições de acesso de caso de uso e contextos de navegação.

A navegação sobre os documentos envolvidos no diagrama de atividades é desenhada sobre os diagramas de restrições de acesso em casos de uso para documentos e sobre o diagrama de contextos de navegação. Cada atividade envolvida no diagrama de atividades é refinada em restrições de acesso de casos. Na sequência, cada restrição de acesso é associada a um diagrama de contextos de navegação. Dentro deste diagrama de contexto de navegação, são relacionados os documentos do diagrama de estrutura de documentos com seus *links* de navegação. Além do relacionamento de documentos já existentes, este diagrama poderá adicionar novos documentos que são visualizados dentro de cada restrição de acessos de casos de uso.

Ao final da modelagem destes diagramas, o diagrama de documentos torna-se mais completo, considerando então todos componentes necessários a uma implementação mais rápida e adequada a ferramentas de desenvolvimento para *Web*.

Na próxima seção, serão detalhadas todas as propostas de extensão à notação UML, que suportarão conceitos inerentes às tecnologias *Web*.

5.3 A UML e as extensões propostas

5.3.1 Diagrama de estrutura de documentos preliminar

A notação do diagrama de estrutura de documentos é uma extensão ao diagrama de classes da UML, e modela componentes de hiperdocumentos, conforme o metamodelo apresentado figura 3.2 do capítulo 3. Ele define a semântica de documentos estruturados, que são parte do repositório de dados organizacionais e implementam a interface gráfica do sistema em desenvolvimento. [LIM 2000].

O diagrama de estrutura de documentos é utilizado para capturar toda estrutura dos hiperdocumentos do sistema em desenvolvimento, que representarão grande parte da *interface* com o usuário. A separação de conceitos comportamentais, navegacionais e projetos de *interface*, maximizam o reuso no projeto e na implementação, além de facilitar a implementação, tornando-a mais ágil. [SCH 2001].

Ainda utilizando o metamodelo da figura 3.2 do capítulo 3, observa-se que a composição de um hiperdocumento apresenta a seguinte constituição:

- Hiperdocumentos, que segundo [NEM 95], é um hipertexto com multimídia (som, imagem, gráfico, animação).
 - Documentos, que para [GAM 95] são definidos como uma combinação de elementos gráficos básicos. (Estes documentos podem ser dos mais variados tipos, como textos, planilhas, apresentações, *postscripts*);
 - Componentes (partes elementares que entram na composição dos hiperdocumentos)
 - Estáticos (Frame, Formulário, Cabeçalho, Declaração, Corpo, Texto, Imagem, Gráfico, Animação);
 - Dinâmicos (*Applets*, *Servelets*, CGI's, ASP, Scripts, Acessos a Banco de Dados, Acessos a Sistemas Legados, *ActiveX*, ORB e PHP);
 - Elemento de Ligação (Âncora e *Links* de Navegação).

O metamodelo da figura 3.2 do capítulo 3 ilustra os relacionamentos e agregações que um hiperdocumento pode ter no início da modelagem. Um hiperdocumento pode ser composto por vários hiperdocumentos, agregando-se componentes dinâmicos, elementos de ligação e estáticos. A descrição e comentários sobre cada um destes componentes são apresentados no esquema de diagrama de estrutura de documentos preliminar na figura 5.3.

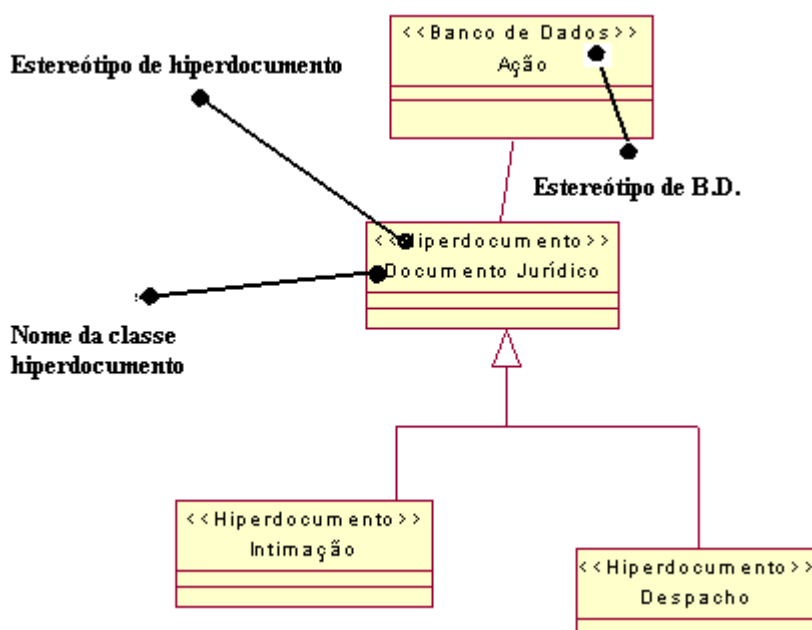


FIGURA 5.4 - Exemplificação da utilização de um diagrama de estrutura de documentos preliminar

A notação de estrutura de documentos pode ser usada para refinamento dos documentos de um sistema ao longo da modelagem. Na fase de elaboração de requisitos gera-se um modelo preliminar, contendo uma visualização macro dos documentos envolvidos. No diagrama de estrutura de documentos final, obtém-se além dos documentos levantados na análise de requisitos, outros documentos intermediários possíveis de serem visualizados com auxílio dos diagramas que se seguem nesta proposta. A estrutura do hiperdocumento também é modelada com mais detalhes, à medida que a modelagem do *workflow* indicar novos documentos ou componentes no modelo de documentos.

O diagrama de estrutura de documentos preliminar não possui ainda uma visualização completa de todos os documentos que serão utilizados pelo sistema de informação. Isto porque nesta fase, ainda se está fazendo o levantamento de necessidades e desenhando-se as possíveis estruturas de hiperdocumentos e documentos estruturados. Neste momento ainda não são conhecidas características dinâmicas de forma mais refinada dos hiperdocumentos. Isto é possível após a modelagem dos contextos de navegação e dos diagramas comportamentais do sistema.

O modelo de estrutura de documento contém características dinâmicas necessárias à implementação de cada hiperdocumento. Nesta visão, o diagrama de estrutura final fornece informações sobre a natureza da implementação de cada componente como: *applet*, *servelet*, CGI, entre outros.

A figura 5.3 demonstra graficamente a utilização do diagrama de estrutura de documentos preliminar, que no capítulo 6 é melhor explorada na figura 6.3, a qual refere-se ao estudo de caso de um sistema de informação para o Poder Judiciário.

Como extensão do diagrama de classes, utilizou-se os próprios ícones apresentados pela UML para caracterizar estereótipos de classes (<<estereótipo>>). A proposta de Flávio Lima [LIM 2000] aborda a concepção de novos ícones para representação dos estereótipos do diagrama de estrutura de documentos, o que acaba por prejudicar o entendimento dos modelos.

A próxima seção trata da modelagem do diagrama de atividades, que é responsável pela especificação do fluxo de atividades dentro do domínio da aplicação.

5.3.2 Diagrama de atividades

A notação do diagrama de atividades modela as características dinâmicas de um sistema de informação. Na modelagem de sistemas de informações convencionais, normalmente é utilizado para modelar processos de negócio que possuam paralelismo ou sejam muito complexos. [BOO 99]. Após o estudo comparativo de técnicas de modelagem de *workflow*, *WfMC* e *Oracle Workflow Builder*, foram relacionados elementos importantes para modelagem de *workflow* que não fazem parte do diagrama de atividades da UML. Tais elementos foram agregados à notação do diagrama de atividades, conforme o metamodelo apresentando figura 2.7 do capítulo 2.

O diagrama de atividades da notação UML, é composto pelos seguintes elementos: [WFM 2000].

- **Atividade:** pode ser definida pelo conjunto de eventos ocorridos sob a responsabilidade de um ator.
- **Evento:** é o momento de transição entre uma atividade antecessora e uma atividade sucessora.
- **Ator:** pessoa ou máquina responsável pela execução de uma atividade.
- **Fluxo:** define a conexão entre as atividades envolvidas no diagrama.
- **Condição:** define uma regra para que seja executada uma determinada atividade.
- **Dependência** entre atividades: validação que existe para que uma atividade possa ser executada após o término de outra atividade.
- **Raia:** separação de atividades por atores num diagrama de atividades.”, conforme relacionado na capítulo 4, na seção 4.3.1.

De acordo com a análise comparativa do capítulo 4, foram enumerados os seguintes elementos para estender a notação do diagrama de atividades da UML:

- **Gatilho:** este proporciona o disparo de uma atividade por um evento. Equivale a uma regra disparada em ocorrência de um evento. Este gatilho pode ainda ter dependência em relação ao tempo.
- **Componente Documento:** representa os hiperdocumentos manipulados pelo *workflow*, utilizados geralmente como *interface*.
- **Atividade Processo:** representa um sub-processo dentro do processo que está sendo modelado.
- **Atividade Função:** a atividade função se refere a uma chamada de um programa externo ou então uma chamada PL/SQL no caso de *Oracle*.
- **Atividade Notificação:** a atividade notificação se refere a uma chamada de intervenção humana. Normalmente deverá haver uma decisão para continuidade da execução das próximas atividades.

Para Mike Andersen [AND 99], a criação de padrões de interoperabilidade para o gerenciamento de *workflow* é algo importante para integração de sistemas inter-empresariais, e para isto, os elementos anteriormente descritos: gatilho, documento, gatilho (evento e tempo) e atividade (processo, função e notificação), foram enumerados como elementos fundamentais para extensão à notação do diagrama de atividades da UML.

O gatilho, incorporado ao diagrama de atividades, possibilita a inclusão de regras que disparam as atividades para um determinado evento. Muito útil para uma automação completa de atividades, pois elimina em muito a intervenção humana sobre o início e execução de atividades.

O documento, conforme toda importância estudada no capítulo 3, é o meio de *interface* entre o sistema de informação e o usuário. O diagrama de atividade estará relacionando a atividade analisada ao documento estruturado no diagrama de estrutura de documentos. O nome atribuído ao documento no diagrama de atividades corresponde ao nome dado no diagrama de estrutura de documentos. O diagrama de estrutura de documentos modela a estrutura de cada documento ao passo que o diagrama de atividades, modela como os documentos serão manipulados pelo *workflow*.

O gatilho com dependência em relação ao tempo, se constitui num item importante quando uma ou várias atividades dependem apenas de um contexto de horas ou dias para ser executadas. Do mesmo modo que os gatilhos, a dependência em relação ao tempo evitará também a intervenção humana, possibilitando uma condição de execução de atividade sem falha de operação.

A criação de sub-tipos de atividades é interessante do ponto de vista de implementação, visto que normalmente as atividades notificação dão origem a hiperdocumentos (gerados dinamicamente) com formulários de entradas ou de saídas. Ou podem acionar ferramentas de *workflow*, *e-mail*, etc. A atividade notificação é

representada pela elipse do diagrama de atividades. A atividade função é a chamada a um programa externo ou a uma *stored procedure* escrita em PL/SQL por exemplo. Esta atividade é representada graficamente por um hexágono. As atividades processo são representadas graficamente por um retângulo, e representam subprocessos dentro do processo que está sendo modelado no diagrama de atividades estendido.

A figura 5.4 apresenta um modelo de representação de diagrama de atividades estendido. Cada atividade relacionada no *Swimlane*, corresponde a um caso de uso do diagrama de casos de uso. O diagrama a partir deste momento lista a ordem e regras de execução de cada caso de uso, bem como a manipulação de documentos feito pelos mesmos.

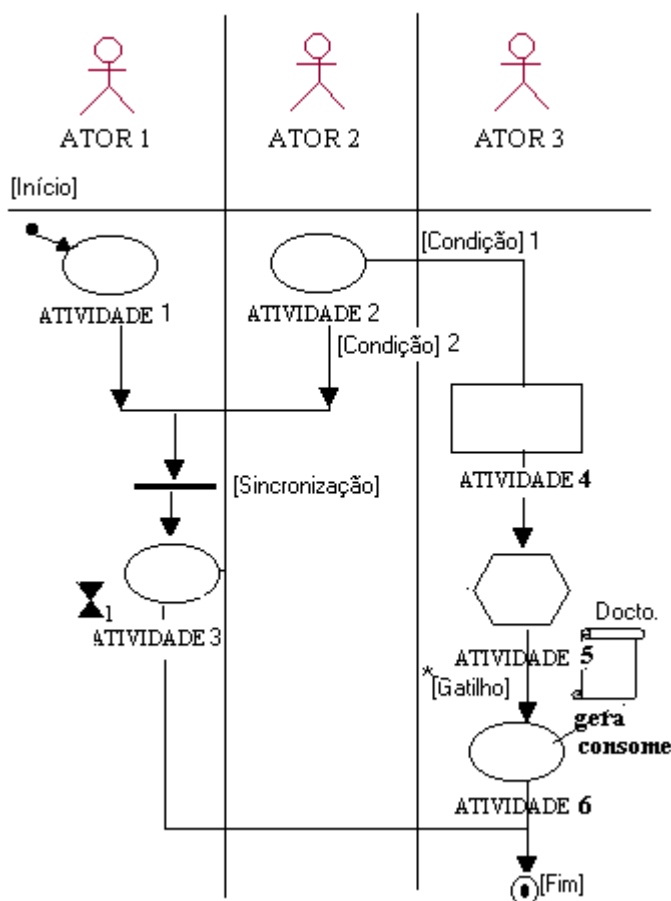


FIGURA 5.5 - Esquema do diagrama de atividades estendido

A figura 5.5 demonstra a utilização do diagrama de atividade referente ao estudo de caso de um sistema de informação para o Poder Judiciário, que é explorado no capítulo 6 na figura 6.4. Através deste exemplo, observam-se os novos conceitos agregados ao diagrama de atividades, melhorando o entendimento do fluxo de atividades entre os vários atores, além de refinar aspectos comportamentais do diagrama de estrutura de documento final.

A implementação do sistema será como na notação da UML. Os diagramas propostos por este capítulo, agregam uma nova extensão à UML no momento de descrição dos casos de uso, principalmente relacionando-se aspectos de navegação, visões de documentos e estrutura de documentos. Os diagramas abaixo relacionados são utilizados para decomposição de cada atividade:

- Modelo de classes persistentes; [RUM 99].
- Diagrama de seqüência para traduzir os cenários; [RUM 99].
- Diagrama de restrições de acesso em casos de uso, conforme item 5.3.3 do capítulo 5;
- Diagrama de contextos de navegação, conforme item 5.3.4 do capítulo 5. Este diagrama substitui o diagrama de colaboração, pois consegue demonstrar a colaboração entre as classes de estruturas de documentos e demonstrar seus links de navegação;
- Detalhamento das classes de controle, transação.[RUM 99].

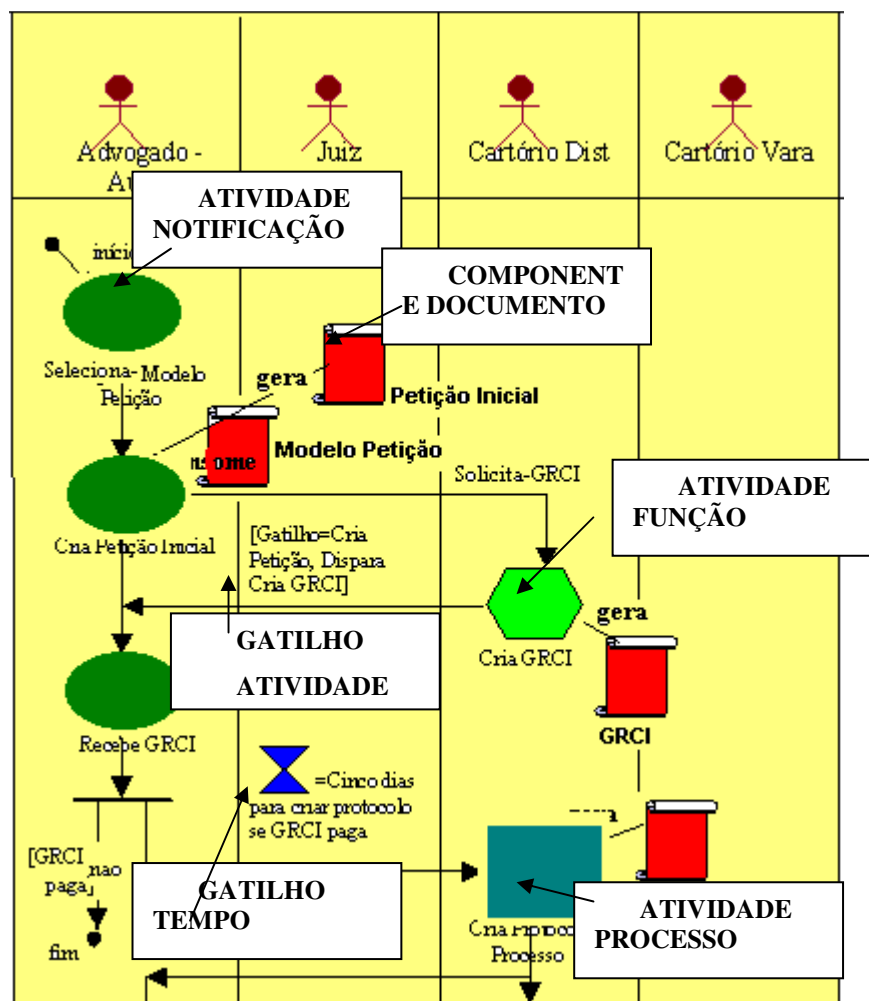


FIGURA 5.6 - Exemplificação do diagrama de atividades

As atividades poderão ser implementadas em alguma linguagem de programação *Web*, ou então disparar processos de *workflow* a partir de uma ferramenta de gerência de *workflow*.

5.3.3 Diagrama de restrições de acesso em casos de uso para documentos

O diagrama de restrições de acesso em Casos de Uso, conforme Lima em [LIM 2000], “especifica formalmente através de quais contextos cada ator poderá acessar os documentos envolvidos”.

Os contextos tendem a designar os acessos que cada ator poderá ter, porém especificando-se que sua visão será dentro de uma das permissões: Consultar, Alterar, Eliminar e Incluir. Conforme figura 5.7.

A idéia de restrições de casos de uso foi elaborada por Flávio [LIM 2000], mas sua notação foi alterada, contendo permissões de manipulação sobre cada caso de uso. Isto porque o diagrama, possui uma semântica muito similar ao diagrama de casos de uso.

Nesta nova concepção de diagramas de contextos de casos de uso, as permissões trarão a conotação de qual ator poderá incluir, modificar, eliminar ou consultar algo no sistema através do hiperdocumento focado.

Tais contextos são de extrema importância, quando se quer demonstrar que atores podem manipular o mesmo caso de uso, porém com visões diferenciadas.

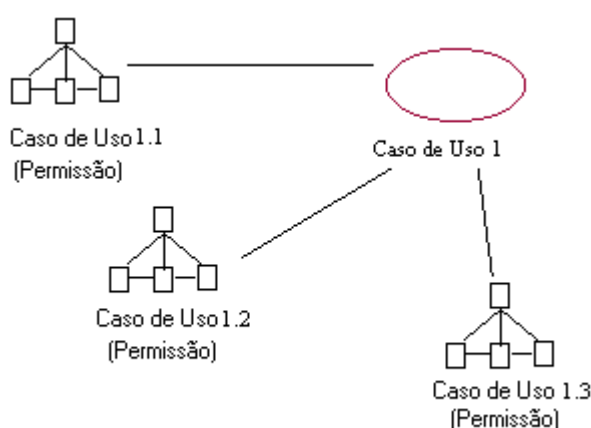


FIGURA 5.7 - Esquema do diagrama de restrições de acesso em casos de uso para documentos

A figura 5.8 permite modelar várias visões sobre os documentos que serão manipulados pelo advogado ao criar uma petição para iniciar um processo dentro do Poder Judiciário. Ainda, exemplifica os estereótipos agregados ao diagrama de restrições em casos de usos para documentos. A elipse representa o caso de uso (proveniente do diagrama de casos de uso), sendo detalhado em restrições de acesso, representadas graficamente pelo ícone representando uma hierarquia de quadrados. Os relacionamentos entre casos de uso e restrições de casos de uso é uma reta simples. As permissões são representadas de forma descritiva, logo abaixo do nome de cada restrição de caso de uso.

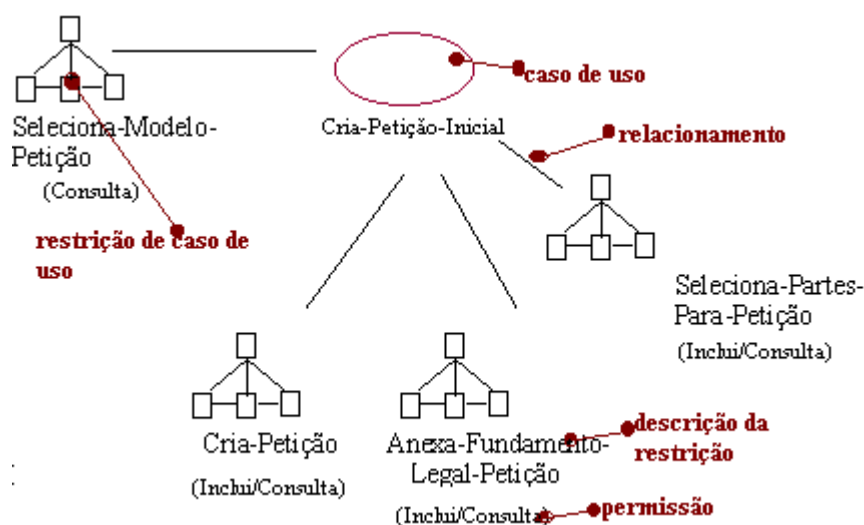


FIGURA 5.8 - Exemplificação do diagrama de restrições de acesso em casos de uso para documentos

5.3.4 Diagrama de contextos de navegação

Este diagrama modela aspectos importantes ao sistema: a estrutura estática de caminhos de navegação e “links” detalhados entre documentos relevantes, segundo Lima em [LIM 2000].

Flávio propôs este modelo em dois níveis: estrutura de navegação e um refinamento detalhado de *links*.

A reformulação deste diagrama utiliza apenas um diagrama. Conforme a figura 5.8, os seguintes elementos foram adotados:

- Restrição de Caso de Uso (Permissão), o qual é proveniente do diagrama de restrições de casos de uso;

- Tipo componente, que identifica se o componente em questão é um formulário, página de entrada, botão, âncora, link, componente executável, acesso a banco de dados ou acesso a sistemas legados;
- Nome do componente: que descreve o nome do tipo de elemento de documento citado anteriormente.
- Ordem de navegação: indica o sentido e ordem da navegação;

Esta fase inicia um processo de maior detalhamento sobre todos os componentes dinâmicos, estáticos e de ligação de cada documento.

Estes refinamentos são agregadas posteriormente ao diagrama de estrutura de documento final, tornando cada hiperdocumento mais completo, prevendo vários elementos possíveis e necessários ao projeto e à implementação.

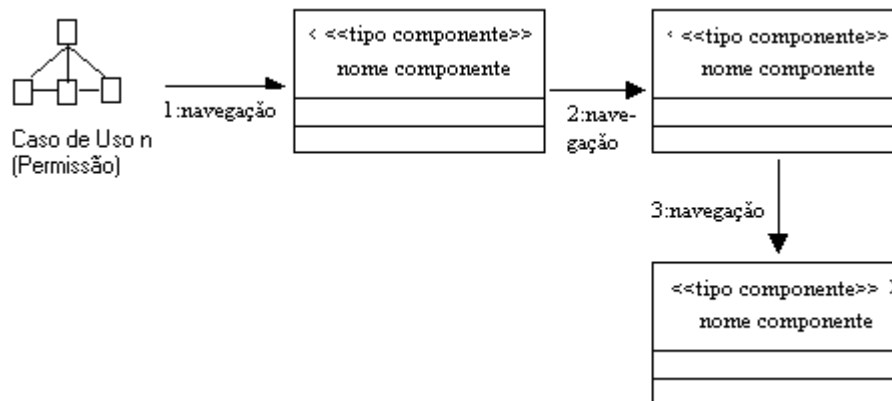


FIGURA 5.9 - Esquema do diagrama de contextos de navegação

A descrição de cada elemento que compõe este diagrama, pode ser observado na figura 5.9. Assim como no diagrama de restrições de casos de uso, o ícone hierarquia de quadrados representa a restrição de acesso em caso de uso. As permissões do contexto vêm descritas logo após o ícone de restrição de acesso em caso de uso. A navegação é descrita abaixo de uma seta indicando a ordem em que ocorrerá a navegação. A navegação será entre hiperdocumentos, e cada hiperdocumento que aparecerá neste modelo deverá ser desenhado no diagrama de estrutura de documentos final, não tendo necessariamente aparecido no diagrama de estrutura de documentos preliminar. A navegação terá uma numeração indicando a ordem de navegação. A representação gráfica de hiperdocumento é a mesma utilizada no diagrama de estrutura de documentos.

Ao final deste diagrama, os *links* de navegação foram modelados e mapeados para o diagrama de estrutura de documentos final.

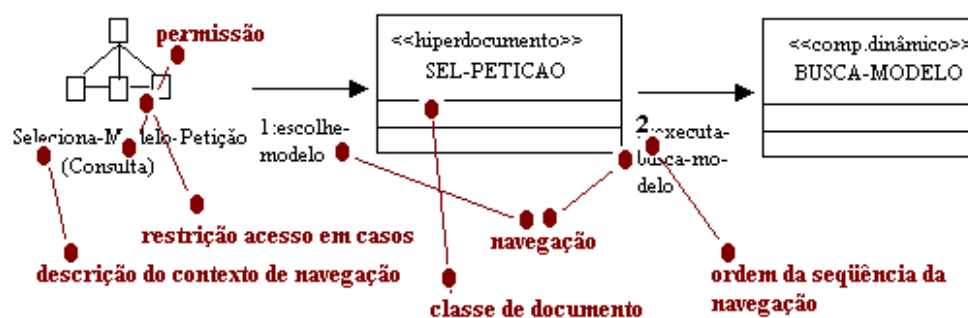


FIGURA 5.10 - Exemplificação do diagrama de contextos de navegação

Para um melhor refinamento de interações, relacionamentos e troca de mensagens entre os objetos e seus vínculos, pode-se complementar a modelagem com diagramas de seqüência da UML. Quando o enfoque é a representação dos contextos de navegação, o diagrama de colaboração pode ser substituído pelo diagrama de contextos de navegação.

5.3.5 Diagrama de classes persistentes e banco de dados

Nesta fase modela-se o banco de dados, os sistemas legados e tabelas de indexação de documentos correspondentes [LIM 2000]. O modelo baseia-se no diagrama de classes, que é o principal modelo da UML. Melhores informações sobre os demais diagramas poderão ser encontrados na documentação UML 1.3 no endereço <http://www.rational.com/uml/resources/documentation/index.jsp>.

Ele descreve o modelo estático do sistema, além de caracterizar os tipos de objetos no sistema e seus relacionamentos. [RUM 99]. Um exemplo de diagrama de sistemas legado pode ser visualizado na figura 6.7, o qual modela um sistema que administra antecedentes criminais das partes envolvidas num processo jurídico. O mesmo pode estar escrito em qualquer plataforma e linguagem. Um exemplo de diagrama de classes persistentes pode ser analisado na figura 6.8.

Os sistemas legados são representados pelo estereótipo <<legado>>, as tabelas de indexação por <<indexação>> e as entidades do banco de dados como <<entidade>>. As demais classes representadas como classes normais do diagrama de classes.

5.3.6 Diagrama de classes de programação

O diagrama de classes de programação define os componentes dinâmicos que serão adicionados ao diagrama de estrutura de documentos final. Para uma modelagem mais

precisa, será necessário o conhecimento sobre a tecnologia de implementação que será adotada. Dentre as tecnologias pode-se destacar *applets*, *servelets*, CGI's, linguagens scripts, ASP, *ActiveX*, entre outros.

É importante salientar que se codificar cada classe de programação, as mesmas deverão observar as seqüências propostas no diagrama de atividades. Ou seja, cada *applet*, script, CGI, deverá ter anexo à sua lógica, as considerações de tempo, seqüência, sincronização, entre outras inerentes ao diagrama de atividades.

Este modelo também adota a notação do diagrama de classes da UML. Como exemplo pode ser encontrado na figura 6.9, caracterizando o tipo de implementação que será feito para cada componente dinâmico extraído do modelo de contextos de navegação.

5.3.7 Diagrama de documento final

Este modelo tem início no diagrama de estrutura de documentos preliminar, e é complementado com os diagramas de atividades, restrições de acesso em casos de uso, contextos de navegação e diagrama de classes de programação.

A figura 5.11 demonstra graficamente um diagrama de estrutura de documento final.

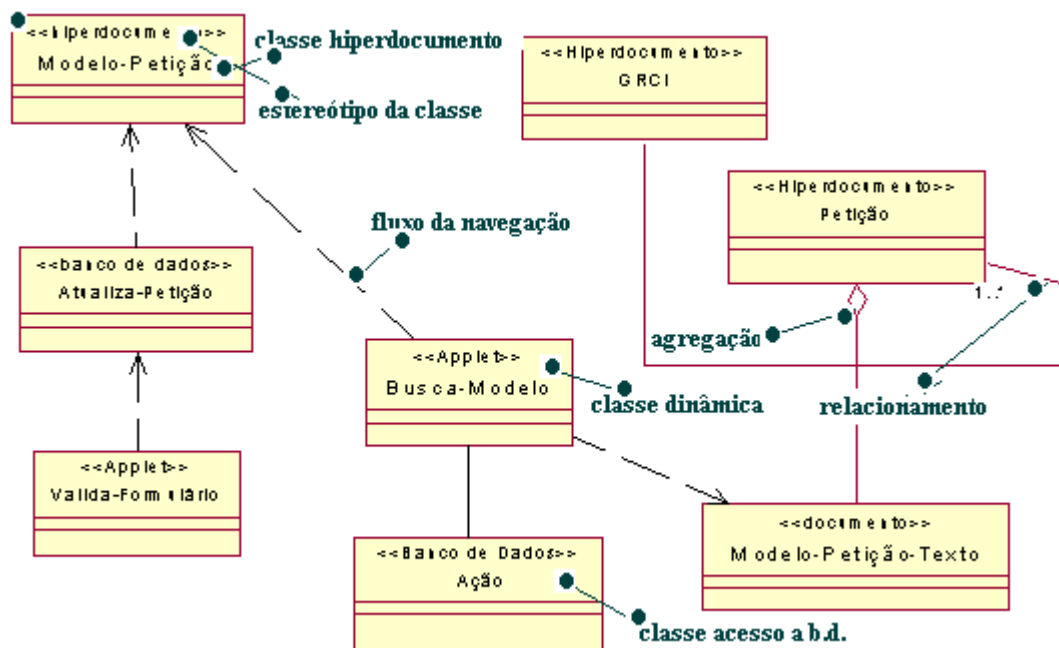


FIGURA 5.11 - Exemplificação do diagrama de estrutura de documento final

A notação do diagrama de estrutura de documento final do caso de uso do capítulo 6, pode ser vista na figura 6.10. Além dos símbolos de associação, agregação e herança do diagrama de classes, é incorporado a esta notação o símbolo de fluxos de navegação sobre os hiperdocumentos.

5.4 Lista de estereótipos adotados com a extensão da UML

5.4.1 Estereótipos do diagrama de estrutura de documento (preliminar e final)

Abaixo segue um descritivo de todos os estereótipos encontrados no diagrama de estrutura de documentos:

- a) **<<hiperdocumento>>** : representa uma classe do tipo hiperdocumento que representa geralmente a *interface* com o usuário do sistema de informação na Internet. Hiperdocumentos são escritos em HTML ou XML. De forma rápida possuem *layout* flexível, suportam componentes executáveis, *links* de navegação para outros hiperdocumentos e são visualizados através de navegadores como por exemplo Netscape e Explorer, segundo [LIM 2000].
- b) **<<componente estático>>** : corresponde a alguma parte do corpo do hiperdocumento e que tenha comportamento estático, como por exemplo cabeçalho, frame, declaração, imagem.
- c) **<<componente dinâmico>>** : todo componente que gerará a execução de alguma tarefa, atividade ou função. Mais adiante são enumerados os tipos de componentes dinâmicos. Representa *interface* para elementos executáveis como CGI ou applet.
- d) **<<componente de ligação>>** : representa um elo de ligação entre dois hiperdocumentos, que podem estar armazenados estaticamente ou ser gerado dinamicamente.
- e) **<<Frame>>** : corresponde a uma divisão visual dentro do corpo do hiperdocumento. Agrupa diferentes hiperdocumentos em uma janela do navegador, conforme [LIM 2000].
- f) **<<Formulário>>** : ambiente dentro do hiperdocumento que servirá de base para execução de um componente dinâmico e conterá informações para serem alimentadas ou visualizadas pelo usuário. Para [LIM 2000], representa grupos de componentes de interface gráfica como campos de texto, botões e rótulos.

- g) <<**Cabeçalho**>> : cabeçalho de um hiperdocumento.
- h) <<**Declaração**>> : são declarações de variáveis de ambiente.
- i) <<**Corpo**>> : local reservado para inclusão de todos os demais componentes estáticos, dinâmicos e de ligação.
- j) <<**Texto**>> : qualquer citação textual dentro do corpo do hiperdocumento.
- k) <<**Imagem**>> : qualquer arquivo digital do tipo bmp, gif, jpg, entre outros que podem estar anexados ao corpo do hiperdocumento.
- l) <<**Gráfico**>> : qualquer arquivo digital que contenha uma representação de imagem vetorial.
- m) <<**Animação**>> : geralmente arquivos do tipo gif que possuem a característica de ficar se movimentando dentro da visualização de um hiperdocumento.
- n) <<**Applet**>> : um tipo de componente dinâmico executado no *browser*.
- o) <<**Servelet**>> : um tipo de componente dinâmico executado no servidor.
- p) <<**CGI**>> : geralmente um componente dinâmico (interpretado ou compilado) que é executado no servidor.
- q) <<**ASP**>> : tipo de componente dinâmico escrito em alguma linguagem de programação como Visual Basic, C++, Delphi.
- r) <<**Banco de Dados**>> : todo acesso que será feito a algum banco de dados.
- s) <<**Legado**>> : acesso e compartilhamento de dados com outros sistemas de informação pré-existentes, que podem estar implementados em modo convencional ou na própria *Web*.
- t) <<**Indexação**>> : indexação de arquivos.
- u) <<**ORB**>> : componente dinâmico responsável pela execução de algum acesso a outros sistemas, sejam legados ou outros sistemas de informação na

Web. Responsável pelas chamadas a outros sistemas. Normalmente utilizado juntamente com o estereótipo <<Legado>>.

- v) <<**ActiveX**>> : tipo de componente dinâmico executado no servidor.
- w) <<**Script**>> : tipo de componente dinâmico agregado ao corpo do hiperdocumento escrito em HTML ou XML.
- x) <<**Âncora**>> : ponto de referência para ligação de um hiperdocumento. Pode ser visto como um estereótipo de associação.
- y) <<**Link**>> : ligação entre hiperdocumentos. Endereçamento.
- z) <<**Entidade**>> : classe do tipo entidade, que dará origem às tabelas de um banco de dados.

5.4.2 Estereótipos do diagrama de atividades

- a) <<**Gatilho Evento**>> : gatilho orientado a evento proporciona o disparo de uma próxima atividade de acordo com uma regra cumprida após a execução de uma atividade antecedente.
- b) <<**Gatilho Tempo**>> : gatilho orientado a tempo proporciona o disparo de uma próxima atividade de acordo com o aguardo de uma determinada situação relacionada a condições de tempo.
- c) <<**Documento**>> : representa os hiperdocumentos manipulados pelo *workflow*, utilizados geralmente como *interface*.
- d) <<**Atividade Função**>> : a atividade função se refere a uma chamada de um programa externo ou então uma chamada PL/SQL no caso de *Oracle*.
- e) <<**Atividade Processo**>> : a atividade processo representa um sub-processo do processo que está sendo modelado através do diagrama de atividades estendido.
- f) <<**Atividade Notificação**>> : a atividade notificação se refere a uma chamada de intervenção humana. Normalmente deverá haver uma decisão para continuidade da execução das próximas atividades.

A notação entre os sinais “<<>>” é utilizada pela UML para representar estereótipos. Uma vez que foi utilizada a ferramenta case *Rational Rose*, decidiu-se em adotar o mesmo padrão para representação dos estereótipos.

5.5 Considerações finais sobre a proposta de extensão

5.5.1 Sobre a proposta da extensão

Na notação WIDe de Flávio de Lima em [LIM 2000], os estereótipos foram representados graficamente por ícones diferentes. Na extensão aqui proposta, foi utilizada a modelagem de diagrama de classes, tratando estereótipos com a própria notação da UML, conforme a figuras 5.1 e 5.9 ilustram. Esta modificação quanto à proposta de Flávio Lima é explicada na seção 5.3.1 deste capítulo. 13- ver base para falar da simplicidade.

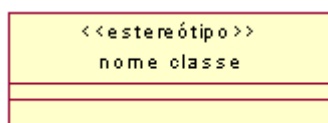


FIGURA 5.12 - Representação de estereótipos em diagramas de classes

Conforme demonstrado graficamente nas figuras 5.1 e 5.2 deste capítulo, o processo de levantamento de requisitos e projeto do sistema, envolvem num primeiro momento a identificação dos casos de uso, dos documentos que serão tratados pela aplicação *Web* e pelo diagrama de atividades estendido, cuja finalidade está no levantamento de todo fluxo de atividades e como estas atividades estarão manipulando os documentos.

Os documentos são compostos por elementos estáticos, dinâmicos e de ligação, fornecendo uma estruturação propícia ao projeto e implementação de uma aplicação *Web*. Isto ao considerar-se que toda aplicação *Web*, tem como uma característica principal o tratamento de hiperdocumentos. E o crescente número de elementos que podem compor um hiperdocumento, é desenhado de acordo com a extensão proposta nos diagrama de estrutura de documentos.

A navegação entre os documentos encontrados nos diagramas de estrutura de documentos e diagrama de atividades é modelada através de dois diagramas: contextos de navegação e restrições de acesso em casos de uso para documentos. Estes dois diagramas, conforme a figura 5.2, modelam as visões que os atores terão sobre os documentos, bem como sua navegabilidade e possíveis componentes dinâmicos que serão mapeados para o diagrama de estrutura de documentos final.

Ao final da modelagem destes diagramas, o diagrama de documentos torna-se mais completo, considerando então todos componentes necessários a uma implementação mais rápida e adequada a ferramentas de implementação na *Web*.

5.5.2 Sobre o *workflow* de atividades e seus tratamentos

No momento em que se comparou técnicas de modelagens de *workflow*, com padrão de interoperabilidade da WfMC e com a ferramenta comercial da Oracle, notou-se que uma atividade pode ser de três tipos:

- **Atividade Processo:** que representa um subprocesso dentro do processo que está sendo modelado.
- **Atividade Função:** a atividade função se refere a uma chamada de um programa externo ou então uma chamada PL/SQL no caso de *Oracle*.
- **Atividade Notificação:** a atividade notificação se refere a uma chamada de intervenção humana. Normalmente deverá haver uma decisão para continuidade da execução das próximas atividades.

Segundo [BOO 99], um diagrama de atividades **mostra** o fluxo de uma atividade para outra. Uma atividade é uma execução em andamento não-atômica em uma máquina de estados. As atividades efetivamente resultam em alguma ação, formada pelas computações executáveis atômicas que resultam em uma mudança de estado do sistema ou o retorno de um valor.

Cada atividade do diagrama de atividades estendido corresponde a um caso de uso dentro dos cenários modelados. A especificação de cada uma de acordo com seus três tipos, permite que tais atividades possam ser projetadas e implementadas de acordo com as ferramentas de automação de *workflow*, ou ferramentas de implementação na *Web*. Sendo uma atividade modelada como atividade função, o projeto e implementação desta pode gerar um componente dinâmico dentro da proposta do diagrama de estrutura de documentos. Uma atividade notificação, pode disparar ferramentas de *workflow*, ferramentas de *e-mail*, ou até mesmo outra atividade função, que por sua vez gerará dinamicamente uma página *Web* para solicitar a intervenção humana.

5.5.3 Considerações sobre a modelagem

Pode-se observar com esta forma de se modelar sistemas de informação na *Internet* que:

- O usuário decide por onde começar a modelagem. Pelo diagrama de atividades ou pelo diagrama de documentos, mas sempre associando atividades a documentos e documentos a atividades;

- Casos de uso correspondem a atividades do *workflow*;
- Casos de uso especificam o domínio da aplicação;
- Atividades estão relacionadas com documentos e casos de uso;
- Casos de uso serão detalhados através de diagramas de sequência ou colaboração, com as classes de implementação ou diagramas de restrições de acesso com contextos de navegação e diagramas de estrutura de documentos.

5.5.4 Regras de consistências do modelo final

Para verificação da consistência do modelo, deve-se obedecer aos critérios abaixo:

- Cada componente de documento tem que estar associado a pelo menos uma atividade;
- Cada atividade deve estar associada a pelo menos um componente de documento;
- Cada componente de documento tem que ter uma atividade que o crie, o use e o destrua. Exceções a essa regra existem, mas devem ser justificadas;
- Cada Caso de Uso deve estar associado a uma Atividade;
- Cada Restrição de acesso deve estar associada a um contexto de navegação;
- Cada Restrição de acesso deve estar associada a uma atividade;
- Cada Restrição de acesso deve estar associada a um contexto de navegação;
- Cada Contexto de navegação deve estar associado ao modelo final de documentos;

5.5.5 Sobre a continuidade da proposta de extensão de Flávio Lima

A proposta de extensão da UML de Flávio Lima tem por objetivo apresentar uma modelagem de estruturação de documentos, navegação e visões dos documentos. Esta pesquisa visa incrementar aspectos dinâmicos à modelagem de sistemas de informação na *Internet*, além de propor mudanças aos diagramas de classes de documentos, restrições de acesso em casos de uso e contextos de navegação.

A modelagem comportamental para sistemas de informação na *Internet* foram associados a conceitos de *workflow*. Um *workflow* auxilia no entendimento de como os documentos são manipulados, em que momento eles são acionados e quais usuários estarão utilizando estes documentos. Além disto, auxilia no momento do levantamento dos componentes dinâmicos e de ligação existentes na estruturação dos documentos. O diagrama de atividades estendido também auxilia no levantamento de documentos ainda não modelados no diagrama de estrutura de documentos.

O diagrama de atividades em conjunto com os diagramas de contexto de navegação e restrições de acessos em casos de uso, relaciona novos documentos

envolvidos na aplicação, e o relacionamento entre os mesmos através de *links* de navegação.

Novamente reforçando as vantagens da proposta, um sistema de informação assim modelado, é capaz de armazenar e atualizar informações, bem como criar padrões de independência entre transações inter-empresariais. Traduz conceitos de hiperdocumentos sendo manipulados pelo *workflow* do negócio.

A experimentação desta proposta encontra-se no capítulo 6, que contém um estudo de caso sobre processos judiciais, demonstrando claramente um sistema de informação que pode ser modelado no ambiente *Web* e que necessita de um controle rigoroso na questão de fluxos de atividades manipulando documentos. No decorrer deste capítulo, à medida que a modelagem é efetuada, os conceitos de extensão propostos estão sendo validados.

6 Experimentação

Para experimentação e refinamento da extensão proposta à notação da UML, foi elaborado um estudo de caso, para que atividades de *workflow* e manipulação de documentos fossem elementos altamente relevantes para um sistema de informação na Internet.

O estudo aborda um ambiente de tribunal de justiça, partindo desde o trabalho de redação de uma petição, até a sentença final de um juiz de direito.

Considera-se um sistema de gestão de documentos jurídicos, no qual juízes, advogados, cartórios e partes possam criar, incluir, consultar, despachar e acompanhar vários documentos. Documentos como: processos, pareceres, intimações, mandados de segurança, laudos, sentenças, etc.

A justiça administra todo conjunto de regras e princípios jurídicos através de processos. Neste sentido, processo então pode ser definido como um conjunto de atos, em sequência preordenada, que devem ser executados para a solução de uma causa. Há vários tipos de processos, de acordo com o sistema jurídico adotado no país. No caso brasileiro, os processos poderão ser do tipo: civil, militar e penal.

O presente estudo de caso, utilizou o entendimento de processos civis para elaboração do estudo de caso. Todo processo civil começa por iniciativa de uma parte. Chama-se parte aos atores envolvidos no processo. As partes poderão ser do tipo réu ou autor. Réu é a parte acusada de algo, enquanto autor é a parte que está acionando a outra parte judicialmente.

As partes necessitam envolver outro ator, denominado advogado. Então o advogado da parte autor inicia o processo através de um documento jurídico chamado petição. A petição descreverá os motivos da ação judicial, mencionando leis e artigos do código civil (também considerado um documento jurídico), bem como motivo da ação. Anexa documentos, provas e encaminha ao Poder Judiciário.

A petição é protocolada e distribuída pelo cartório de distribuição. Este cartório após efetuar cobrança da entrada do processo, de acordo com a ação, distribui o processo ao cartório da vara de interesse. No caso deste estudo de caso, o cartório da vara civil é quem irá receber a petição inicial do advogado, devidamente protocolada.

O cartório da vara estará iniciando alguns procedimentos necessários antes de enviar o processo ao juiz de direito. Dentre estes procedimentos, o cartório poderá estar emitindo mais documentos jurídicos que irão compor o processo. Após o juiz receber o processo, toma algumas decisões, que podem ser desde intimar a parte réu, até solicitar ao cartório da vara para expedição de outros documentos jurídicos, a divulgação em jornais e diários oficiais, agendamento de julgamento, entre outros. Todos os procedimentos, são devidamente registrados nos autos do processo. [CAR 95]

Muitos documentos jurídicos poderão ser encaminhados às partes envolvidas através de outro ator: o oficial de justiça. Este tem por responsabilidade verificar diariamente junto aos cartórios das varas todo expediente que deverá cumprir, além de prestar contas dos documentos que ficou encarregado de entregar.

Finalmente, após todos expedientes executados e audiências proferidas, o processo é encerrado através das decisões do juiz. Todo processo judicial termina com um documento jurídico denominado sentença ou arquivamento do processo. O juiz determinará a sentença que será aplicada às partes envolvidas no caso.

Neste sistema, todas as informações estão estruturadas em documentos interligados, com dados armazenados em tabelas de banco de dados, páginas HTML, XML e arquivos de imagens, filmes, sons, etc.

A gestão de documentos controla quem usa e quem é responsável pelos documentos referidos. O sistema de *workflow*, trata da seqüência de tarefas e ativação de ferramentas, que são adequadas à tramitação dos documentos jurídicos.

Nota-se a importância de uma modelagem orientada a *workflow* e documentos neste estudo de caso, visto que há um volume muito grande de documentos tramitando entre os vários atores envolvidos.

Nas próximas seções estarão sendo validados os diagramas da UML, bem como as extensões propostas.

6.1 Diagrama de casos de uso

Conforme analisado no capítulo 5, o diagrama de casos de uso auxilia no levantamento de requisitos e análise do sistema de informação, demonstrando os cenários do negócio.

Para o cenário Tribunal de Justiça (T.J.), seguem os seguintes atores:

- Juiz de Direito;
- Advogado (Representante apto a defender cada uma das partes);
- Parte (autor e réu);
- Cartório de Distribuição (responsável pela distribuição inicial dos processos às varas de direito);
- Cartório da Vara (Local no qual os processos são encaminhados, julgados, arquivados);
- Oficial de Justiça (responsável pelo encaminhamento de documentos jurídicos às várias partes envolvidas no processo).

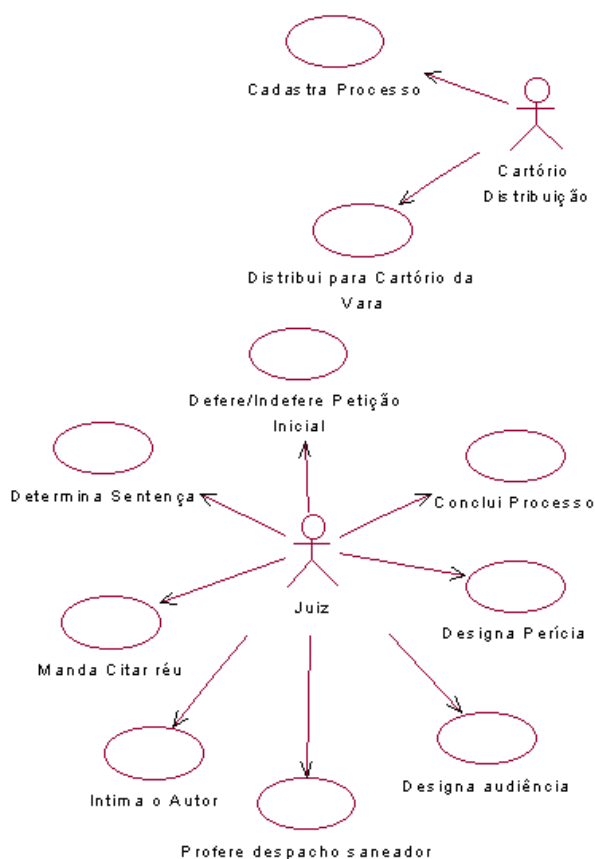


FIGURA 6.1 - Diagrama de casos de uso de um tribunal de justiça(1)

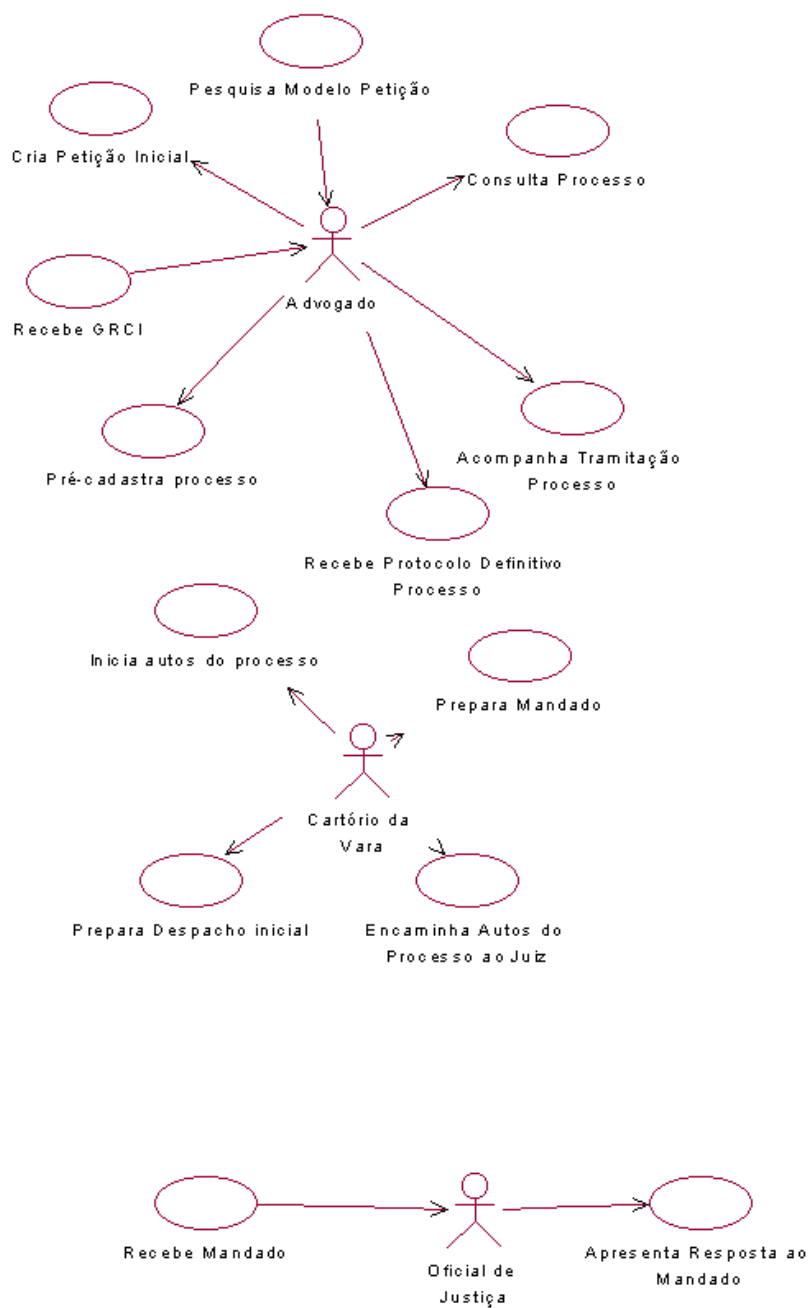


FIGURA 6.2 - Diagrama de casos de uso de um tribunal de justiça(2)

6.2 Diagrama de estrutura de documentos preliminar

O diagrama de estrutura de documentos preliminar define os principais documentos e suas estruturas. A figura 6.3 demonstra os principais documentos envolvidos num ambiente de Tribunal de Justiça.

As classes Lei, Código e suas subclasses são provenientes de documentos textos existentes sob a forma de estereótipos <<documento>>.

A classe Ação é um estereótipo do tipo <<banco de dados>>, no qual o Tribunal de Justiça mantém um catálogo de todos os tipos de ações pré-existentes no meio jurídico.

As partes envolvidas num processo já existem no sistema de informação atual, chamado de sistema legado e associado no diagrama de estrutura de documentos como um estereótipo <<legado>>.

As demais classes são tipo hiperdocumento.

A idéia do sistema é que o advogado do autor inicie o processo através de uma petição inicial que terá seu modelo dentro do próprio sistema de informação do Tribunal de Justiça.

Na seqüência, há uma parte do diagrama de atividades, demonstrando todos os procedimentos iniciais da entrada de um processo no T.J..

6.3 Diagrama de atividades

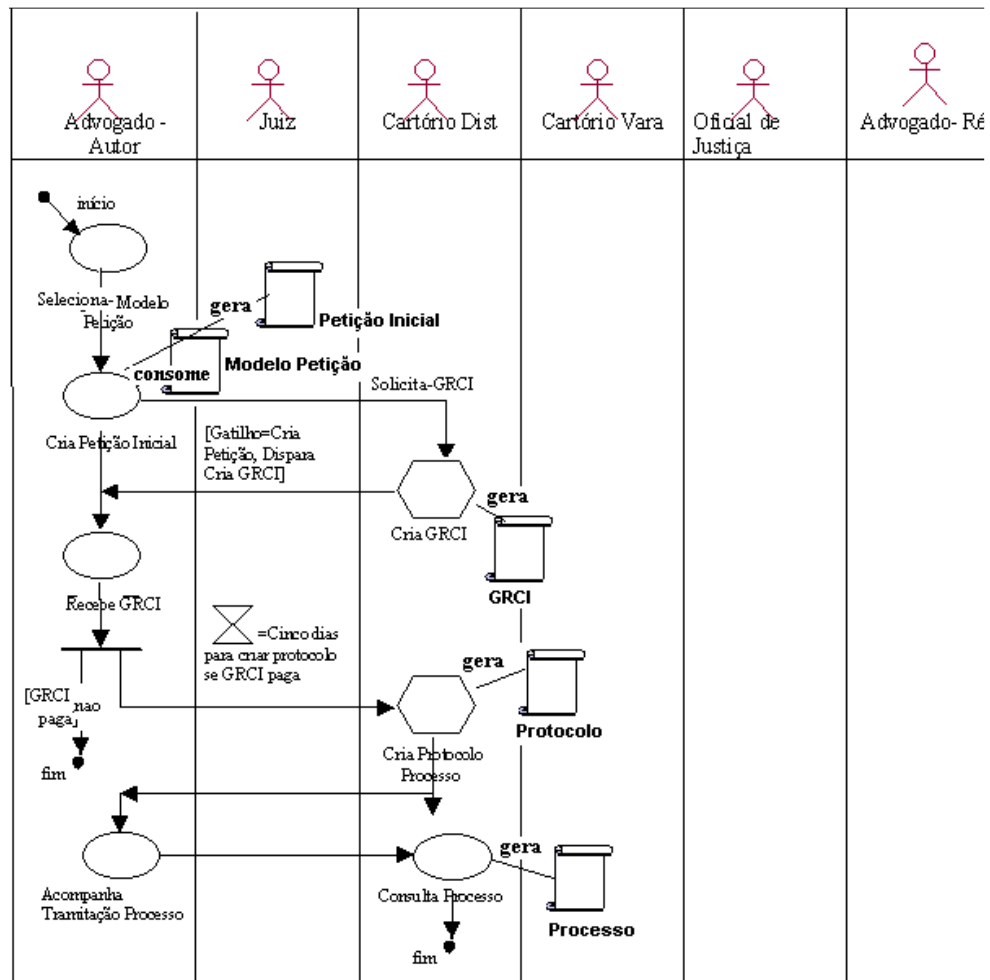


FIGURA 6.4 - Diagrama de atividades par ao início de um processo no tribunal de justiça

Para auxiliar no entendimento dos casos de uso e documentos manipulados pelo sistema de informação, o diagrama de atividades modela os casos de uso relacionados ao ator advogado.

O advogado pesquisa um modelo de petição no sistema de informação do Tribunal de Justiça. A partir do modelo encontrado, cria sua petição inicial no sistema. Após ter criado a petição, automaticamente o sistema dispara uma guia de recolhimento das custas iniciais, a GRCI, que deverá ser paga no banco. Caso a guia não seja paga em cinco dias, o protocolo é cancelado. A guia sendo paga, o protocolo é confirmado e o processo iniciado.

Neste momento, o diagrama de atividades deixa claro as responsabilidades e a seqüência de tarefas sobre os casos de uso. No momento de elaboração dos diagramas

de classes de programação, toda a seqüência aqui elaborada, será implementada a nível de código em cada classe de programação.

6.4 Diagrama de restrições de acesso em casos de uso para documentos

O diagrama de restrições de acesso em casos de uso auxilia na modelagem das visões que cada ator terá sobre os documentos.

O advogado do autor inicialmente visualiza um documento que lista todos os modelos de petição pré-cadastradas no T.J, pelo caso de uso (Seleciona-Modelo-Petição). O caso de uso Cria-Petição, atualiza os dados contidos no formulário de petição. Anexa-Fundamento-Legal-Petição, permite que além de atualizar dados escritos sobre o formulário, o advogado armazene fotos, documentos, imagens e gráficos. O caso de uso Seleciona-Partes-Para-Petição permite um acesso ao sistema legado existente, para solicitando informações sobre partes já cadastradas. Caso as partes não estejam cadastradas, o mesmo envia as informações de tais partes para serem armazenadas no sistema legado.

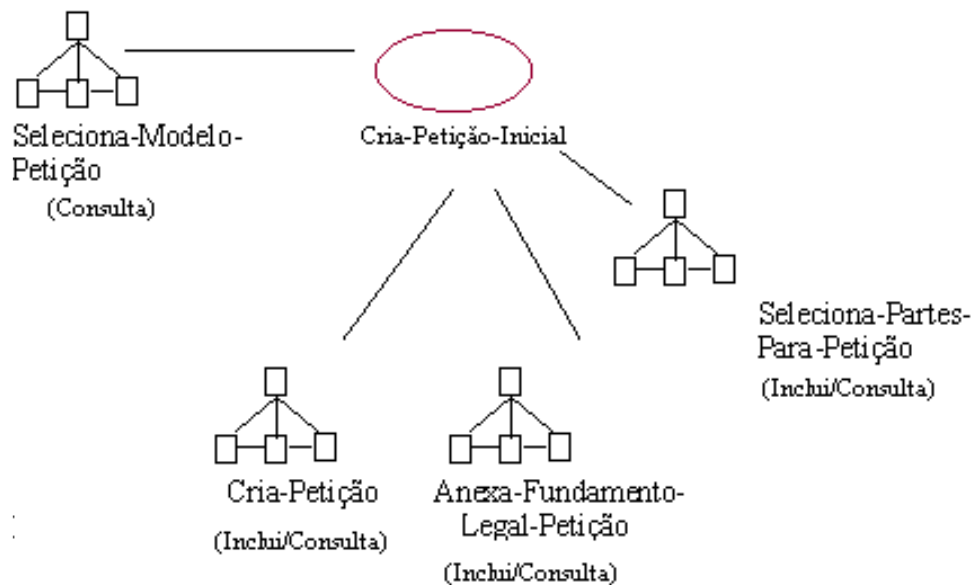


FIGURA 6.5 - Diagrama de restrições de acesso em casos de uso para documentos de um tribunal de justiça

6.5 Diagrama de contextos de navegação

Os diagramas de contextos de navegação permitem que cada restrição de acesso de casos de uso, seja mais detalhado, contendo inclusive sua navegabilidade.

Seguindo o exemplo do estudo de caso, o advogado quando está na restrição de caso de uso Selecciona-Modelo-Petição, pode selecionar no hiperdocumento SEL-PETIÇÃO, o tipo de petição que lhe servirá de modelo. Ao selecionar o modelo, um componente dinâmico (que será definido no diagrama de classes de programação), faz a execução de busca do modelo. Após este componente dinâmico buscar o modelo, apresenta um hiperdocumento MOD-SELECIONADO.

Quando o modelo já estiver selecionado, então o diagrama de restrição de acessos de casos de uso Cria-Petição permite que o advogado preencha o formulário. Este solicita ao advogado atributos importantes para o armazenamento do processo no T.J.. O formulário após preenchido, disponibiliza um botão que aciona o componente dinâmico (VALIDA-FORMULÁRIO), o qual valida os dados contidos neste. ATUALIZA-PETIÇÃO é uma classe que atualiza as informações do formulário no banco de dados do T.J..

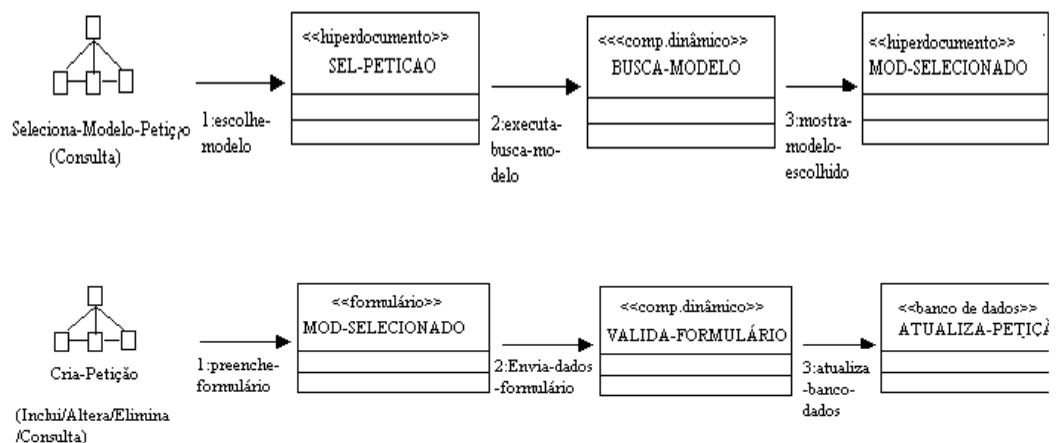


FIGURA 6.6 - Diagrama de contextos de navegação para as restrições de acessos de caso de uso Selecciona-Modelo-Petição e Cria-Petição

6.6 Diagrama de classes persistentes e banco de dados

No estudo de caso em questão, foram desenhados dois diagramas:

- Diagrama de Classes dos Sistemas Legados;
- Diagrama de Classes do Sistema do Tribunal de Justiça.

Estes dois diagramas exemplificam um anexo ao sistema legado de Partes, e a modelagem de diagrama de classes persistentes a parte do banco de dados do Tribunal

de Justiça. O modelo em estudo se tornou muito grande. Então foram selecionados os casos de uso relacionados ao ator advogado para a execução de todos os diagramas desta extensão proposta.

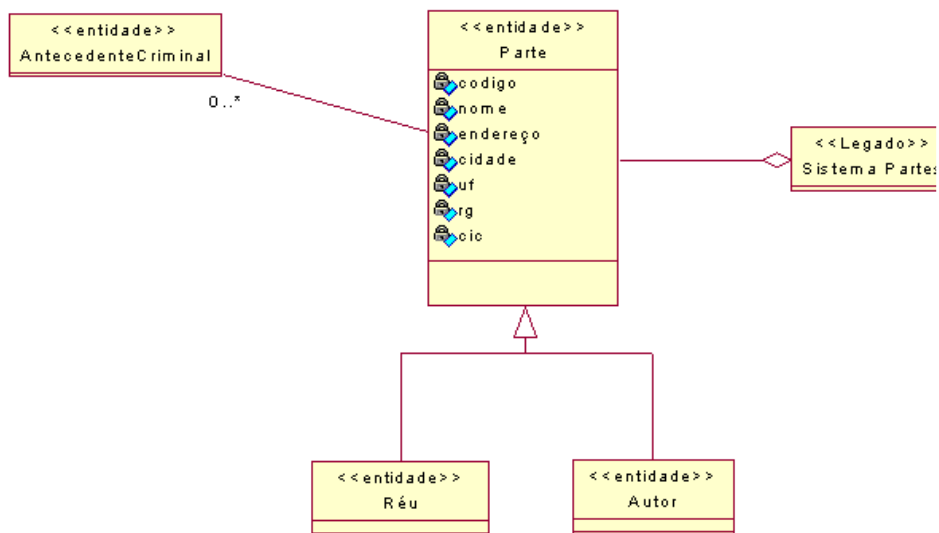


FIGURA 6.7 - Diagrama de Sistemas Legado que gerencia as partes do processo

A figura 6.7 descreve as entidades que estão contidas no sistema legado que possui todas as partes.

Além da classe de estereótipo <<legado>>, o modelo demonstra também as entidades que serão manipuladas pelo mesmo.

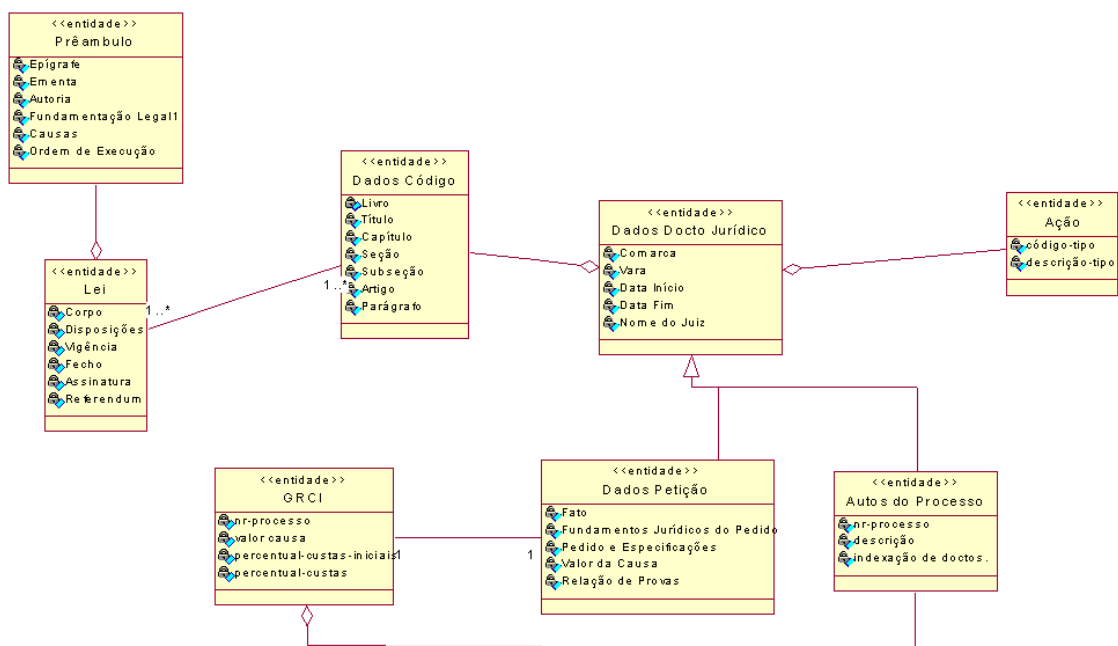


FIGURA 6.8 - Diagrama de classes persistentes do sistema de um tribunal de justiça

O diagrama de classes persistentes, exemplificado na figura 5.10, descreve as principais entidades que serão manipuladas pelo projeto, contextualizado na visão nos casos de uso do advogado do autor.

6.7 Diagrama de classes de programação

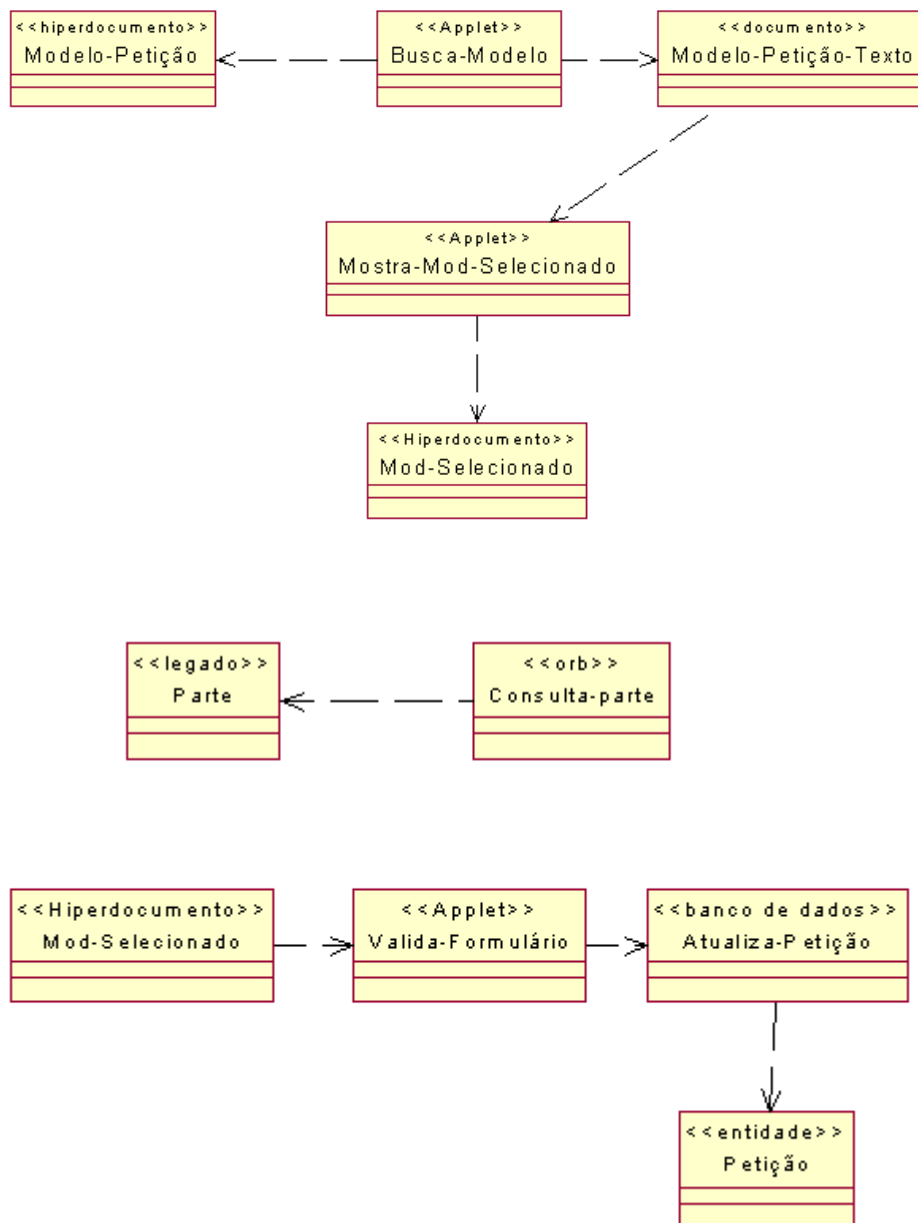


FIGURA 6.9 - Diagrama de classes de programação de um tribunal de justiça

O diagrama de classes, segundo a figura 6.9, demonstra os componentes dinâmicos se relacionando com seus hiperdocumentos (que já fazem parte do diagrama de documentos preliminar).

O hiperdocumento Modelo-Petição, possui a ela associado um componente dinâmico (*Applet*) Busca-Modelo, que irá pesquisar, de acordo com os parâmetros solicitados pelo advogado, dentro de um arquivo texto, o modelo solicitado de petição. Após encontrá-lo, o componente Mostra-Mod-Selecionado exibirá para o advogado, o modelo em formato HTML, através do hiperdocumento Mod-Selecionado.

Ao iniciar a inclusão de dados na Petição-Modelo, o advogado poderá tanto consultar, quanto se não existir, cadastrar as partes envolvidas no processo, no sistema legado do T.J. A consulta ou inclusão de dados está representada pelo componente Consulta-Parte (ORB), que será responsável em atualizar o componente (legado) Partes. Com o formulário a ser preenchido no hiperdocumento Mod-Selecionado, o advogado irá transmitir para criar sua petição no T.J., porém antes disto será disparado outro *applet* que validará o formulário, que evitará alguns tipos de consistências nos campos preenchidos, em seguida esta petição será atualizada no banco de dados do T.J..

6.8 Diagrama de estrutura de documento final

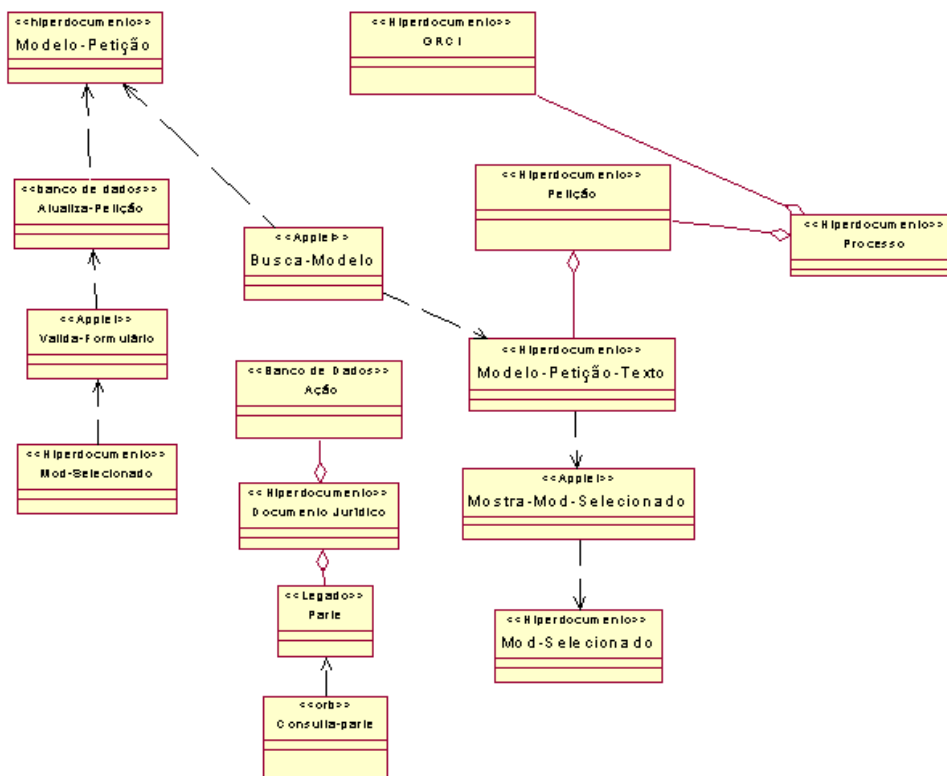


FIGURA 6.10 - Esquema do diagrama de documento final de um tribunal de justiça

O diagrama de estrutura de documento final, apresenta algumas classes do diagrama de classes preliminar, por questões de delimitação do problema para exemplificação.

Neste modelo pode-se observar que os hiperdocumentos e documentos já citados no diagrama de estrutura de dados preliminar, são acompanhados dos componentes dinâmicos que foram elaborados a partir dos diagramas de atividades, de contextos de navegação e de restrições de acesso de casos de uso.

7 Conclusão

A temática comum a todos os conteúdos deste estudo sobre modelagem de *workflow*, modelagem de documentos, *workflow* manipulando documentos e os capítulos que correspondem à própria extensão, denota que a *Internet* mudou a forma de implementação de sistemas de informação, trazendo como consequência, a preocupação de uma modelagem voltada aos conceitos de navegação, visões de documentos, gerenciamento de atividades e estruturação de tais documentos.

A *Internet* pressupõe atividades em sequência, em paralelo e o sincronismo de tais atividades para produzir uma navegabilidade correta. Conforme [BAL 98] e [GIN 2001], os sistemas para *Web* apresentam novas características que devem ser consideradas no desenvolvimento de sistemas. Dentre estas características destacam-se neste trabalho: o gerenciamento de documentos, mecanismos de navegações, visões de documentos e manipulação dos documentos por um sistema de gerência de *workflow*.

Estas peculiaridades inerentes à *Internet* viabilizam uma modelagem de sistemas de informação diferenciada. Pensar em documentos e *workflow* desde o início do levantamento de requisitos de sistemas, propõe um conhecimento maior sobre a implementação final. A implementação não corre o risco de utilizar um “*browser*” apenas como uma forma de *interface*. A *Web* traz inovações na forma de interação do usuário ao sistema, e um sistema de informação na *Internet* deve preocupar-se agregar estas novas características.

Para Flávio Lima em [LIM 2000], o grande benefício de uma modelagem de sistemas integrada às visões de documentos, estruturação de documentos e manipulação dos documentos por um *workflow* encontra-se numa modelagem capaz de combinar às potencialidades organizacionais, a disseminação de princípios gerenciais independentes de seu tamanho e de sua localização geográfica.

Outra característica importante deste trabalho encontra-se na validação da extensibilidade da UML, propondo um novo foco de análise e projeto orientados a objetos.

Todos os setores do mercado convergem para o comércio eletrônico, e quando fala-se em comércio eletrônico deve-se entender que toda a gestão administrativa, financeira, produtiva, logística, e comercial estarão envolvidas na *Internet*. Isto é *business-to-business*, no qual não bastará que uma empresa tenha uma *homepage* na *Internet* divulgando seus produtos ou serviços, mas deverá estreitar seu canal de comunicação, negociação, divulgação, vendas e atendimento a clientes através de sistemas de informação bem projetados para *Web*.

A modelagem de sistemas de informação deve abordar estas novas características de gestão de negócios. A nova modelagem de sistemas de informação deverá atender não somente a uma empresa, mas a uma cadeia de empresas e indivíduos buscando informações, negociando, comprando, vendendo e até mesmo compartilhando informações produtivas entre empresas.

A globalização proporcionou às empresas a implantação de fábricas geograficamente distribuídas. Com isto, a forma atual de trabalho em grupo tem sido exercitada através da *Internet*. [HIL 97]. As pessoas compartilham o que pensam, o que aprendem, projetam, escrevem, criam, analisam e decidem. E todo este compartilhamento pode ser feito através de recursos da *Internet*.

A preocupação da modelagem de sistemas de informação na *Internet* encontra-se em analisar o processo de forma “*top-down*” e modelar tais recursos (*workflow* e documentos) capacitando a automação e acompanhamento de todo fluxo de documentos inerentes à organização.

7.1 Características dos temas abordados no trabalho

O capítulo sobre *workflow* abordou estudos de modelagens de *workflow*, relacionando-se todos os elementos encontrados em várias técnicas, no padrão de interoperabilidade da WfMC, sobre a ferramenta comercial da *Oracle* e sobre o diagrama de atividades da UML. A partir destes elementos foi proposto um metamodelo para viabilizar a análise dos conceitos encontrados para modelagem de sistemas de *workflow*.

No capítulo sobre documentos estruturados e modelagem de documentos, a proposta era o levantamento sobre padrões como o “*Composite*” e todas as definições sobre HTML e XML. Este capítulo foi responsável pelo levantamento de como documentos podem ser estruturados e como podem vir a compor um sistema de informação. Tanto o estudo de *workflow* quanto o estudo de documentos, serviram de base, através de metamodelos, para avaliação de elementos e características importantes que posteriormente foram incorporados à modelagem de sistemas de informação na *Internet*.

Para análise e projeto de sistemas de informação na *Internet*, os dois elementos: documentos e *workflow*, não poderiam ser trabalhados isoladamente. Um *workflow* manipula documentos, e estes documentos podem ser estruturados. Então a idéia de manipulação de documentos pelo sistema de *workflow* tratou de questões conceituais e também trouxe uma visão de como isto poderia ser modelado em UML através do diagrama de atividades da UML.

O alvo do trabalho encontra-se na proposta de extensão da UML para modelar sistemas de informação na *Internet* como sistemas de *workflow* manipulando hiperdocumentos estruturados. Conforme o capítulo 5, a modelagem incorporou características existentes em documentos e *workflow* desde a fase levantamento de requisitos. Já neste momento é feito o levantamento do fluxo de atividades sobre os hiperdocumentos e a modelagem de hiperdocumentos, associadas ao estudo dos cenários de casos de uso.

A notação do diagrama de atividades da UML estendida, tem a finalidade de modelar atividades manipulando os documentos envolvidos no sistema. Além do diagrama de atividades, o diagrama de estrutura de documentos faz com que sejam modelados todos os hiperdocumentos envolvidos na implementação, auxiliado pelos diagramas de restrições de acesso em casos de uso e diagramas de contextos de navegação. Estes dois últimos diagramas citados permitem a modelagem de visões de documentos por ator e pela modelagem da navegação sobre os hiperdocumentos, além de permitir uma visualização mais refinada dos hiperdocumentos envolvidos no processo.

Isto propicia uma implementação planejada, com idéias das características de navegação, visões de documentos, componentes dinâmicos e fluxo de atividades automatizadas.

Para experimentação das extensões da notação do diagrama da UML, foi feito um estudo de caso, que avalia todos os diagramas estendidos com um exemplo sobre um sistema de informação para um Tribunal de Justiça. Relatando toda tramitação de documentos junto aos processos nas várias varas, cartório de distribuição e entre oficiais de justiça. Um exemplo que contém um número grande de atores e documentos sendo manipulados pelo sistema de informação na *Internet*.

7.2 Contribuições do trabalho

As contribuições propostas pelo presente trabalho são:

Aplicação de características de ferramentas de implementação para ambientes *Web*, à modelagem de sistemas, através do desenho de diagramas de estruturação de documentos e modelagem de *workflow*, elaborando um modelo de documentos com aspectos de hiperdocumento: dinâmicos, estáticos e de ligação.

Possibilitar a modelagem de um sistema de informação capaz de abordar a utilização de um sistema de gerência de *workflow*, fazendo com que todos casos de uso sejam mapeados para o diagrama de atividades possibilitando maior produtividade na etapas de projeto e implementação da aplicação *Web*.

Modelagem capaz de integrar e avaliar aspectos de navegação, estruturação de documentos e manipulação de atividades sobre hiperdocumentos.

Diminuição na falha de projetos desenvolvidos para *Web*:

Modelando todos hiperdocumentos necessários à implementação de SII;

Modelando fluxo de atividades existente dentro dos cenários envolvidos na modelagem da aplicação;

Prevedendo a navegabilidade entre os hiperdocumentos;

Prevedendo o comportamento dinâmico da aplicação através do diagrama de atividades e outros diagramas comportamentais da UML;

A modelagem utiliza os conceitos de sistemas de informação na *Internet* como hiperdocumentos, componentes ativos e navegação;

Utilização da UML com poucas extensões;

As extensões utilizadas estão de acordo com os conceitos da UML;

Maior proximidade da especificação da aplicação com as ferramentas de implementação na *Web*;

Modelagem incremental;

Modelo consistido ao longo do processo de modelagem.

7.3 Sugestões para trabalhos futuros

O presente trabalho teve como principal meta, estender à UML, *workflow* manipulando hiperdocumentos. Como sugestão para trabalhos futuros pode-se propor:

Integração da modelagem de sistemas de informações na *Internet* com ferramentas de gerência de *workflow*, como *Oracle Workflow Builder*.

Integração com ferramentas CORBA para acesso aos sistemas legados;

Implementação do trabalho utilizando-se das tecnologias *Web*;

Implementação de um gerador da modelagem.

Os projetos anteriormente relacionados podem ser um pouco complexos, porém representariam uma avançada tecnologia no que se refere à automatização do processo de modelagem de sistemas de informação na *Internet* com enfoque em aspectos dinâmicos sobre sistemas na *Web*.

Bibliografia

- [AMA 97] AMARAL, V. L. **Técnicas de modelagem de workflow**. Porto Alegre: CPGCC da UFRGS, 1997. Trabalho Individual
- [AND 99] ANDERSON, M. **Workflow interoperability – enabling e-commerce**. 1999. Disponível em: < [http:// www.wfmc.org/ standards/ docs/ IneropChallPublic.PDF](http://www.wfmc.org/standards/docs/IneropChallPublic.PDF) >. Acesso em: jan. 2000.
- [BAL 98] BALASUBRAMANIAN, V.; BASHIAN, A. Document mangagement and Web technologies. **Communications of the ACM**, New York, v.41, n.7, p.25-36, July 1998 .
- [BOO 99] BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **The unified modeling language user guide**. Califórnia: Addison Wesley, 1999. 356 p.
- [BOR 2000] BORTOLI, L.A.; PRICE, A.M. de A. Um método de trabalho para auxiliar a definição de requisitos. In: JORNADAS IBEROAMERICANAS DE INGINIERIA DE REQUIQUISITOS Y AMBIENTES DE SOFTWARE, 2000, Cancún. **Memorias**. Cuernavaca: Centro Nacional de Investigación Y Desarrollo Tecnológico, 2000. p.1-12.
- [CAR 95] CARDOSO, O.F. **Introdução ao estudo do direito**. Belo Horizonte: Del Rey, 1995. 438 p.
- [CAS 99] CASATI, F. **Conceptual modeling of workflows**. [Milão]: Politécnico de Milão, 1995. Disponível em: < [Http:// www.elet. polimi.it/ pub/ papers/ Fabio.Casati/OOER95Final](Http://www.elet.polimi.it/pub/papers/Fabio.Casati/OOER95Final), Acesso em: nov. 1999.
- [ELL 80] ELLIS, C.A.; NUTT, G.J. Office Information Systems and Computer Science. **Communications of the ACM**, New York, v.12, n.1, p.27-60, Mar. 1980 .
- [ERI 98] ERIKSON, H.E.; PENKER, M. **UML Toolkit**. New York: Wiley Computer Publishing, 1998. 253 p.
- [FOW 97] FOWLER, M.; KENDALL, S. **UML distilled : applying the standard object modeling language** . 6TH ed. Berkley: Addison Wesley Longman, 1997. 358 p.
- [GIN 2001] GINICE, A.; MURUGESAN, S. Web Engineering : an introduction. **IEEE Transactions on Multimedia**, [S.l.], v.8, n.1, p.14-18, Jan/Mar. 2001 .
- [GAI 97] GAIL, L.R.; MOUE, D.L.; SLEIN, J.A. A case for document management functions on the Web. **Communications of the ACM**, New York, v.40, n.9, p.81-89, Sept. 1997 .
- [GAM 95] GAMA, E.; HELM,R.; JOHN, J. **Design Patterns** : elements of reusable object-oriented software . California: Addison Wesley, 1995. 423 p.

- [GEO 95] GEORGAKOPOULOS, D.; HORNICK, M.; SHETH, A. An overview of workflow management : from process modeling to workflow automation infrastructure. **ACM Distributed and Parallel Databases**, New York, v.1, n.3, p.119-153, Mar. 1995 .
- [HAG 2000] HAGEN, C.; ALONSO, G. Exception handling in workflow management systems. **IEEE Transactions on Software Engineering**, New York, v.26, n.10, p.943-958, Oct. 2000 .
- [HAR 98] HARMON, P. **Understanding UML** : the developer's guide with a Web-based application in Java . San Francisco: Morgan Kaufmann, 1998. 356 p.
- [HIL 97] HILLS, M. **Intranet as groupware** . Canada: John Wiley & Sons, 1997. 323 p.
- [JOO 95] JOOSTEN, S.M.M. A method for analyzing workflows. In: WORKFLOW MANAGEMENT SUPPORT SYSTEMS; ECSCW CONFERENCE, 1995, Stockholm. **Tutorial**. [S.l.: s.n.], 1995.
- [LIM 2000] LIMA, F. WIDe : uma metodologia integrada para desenvolvimento de sistemas de informação na Internet baseados em documentos . Porto Alegre: CPGCC da UFRGS, 2000. Dissertação de Mestrado.
- [MAR 2000] MARJANOVIC, O. Dynamic verification of temporal constraints in production workflows. In: AUSTRALASIAN DATABASE CONFERENCE, ADC, 11.,2000, Cambeira, Australia. **Proceedings...** Los Alamitos, CA: IEEE Computer Society, 2000. p. 74-81.
- [MOR 98] MORGENSTERN, M. Integrating web and database information for collaboration through explicit metadata. In: IEEE INTERNATIONAL WORKSHOPS ON ENABLING TECHNOLOGIES: INFRASTRUCTURE FOR COLLABORATIVE ENTERPRISES, WEIICE, 7., 1998. **Proceedings...** Los Alamitos, CA: IEEE Computer Society, 1998. p. 204-210.
- [NEM 95] NEMETZ, F. **HMT** : Modelagem e projeto de aplicações hipermídia . Porto Alegre: CPGCC da UFRGS, 1995. Dissertação de Mestrado.
- [NIC 98] NICOLAO, M.; VICCARI, R.M.; EDELWEISS, N.; OLIVEIRA, J.P.M. Modelagem conceitual de workflow para cursos na Internet. In : SIMPÓSIO BRASILEIRO DE INFORMÁTICA EDUCATIVA, 9., 1998, Fortaleza. **Anais...** Fortaleza: UFC, 1998.
- [NIO 98] NICOLAO, M. **Modelagem de workflow utilizando um modelo de dados temporal orientado a objetos com papéis** . Porto Alegre: CPGCC da UFRGS, 1998. Dissertação de Mestrado.
- [PAS 94] PASTOR, E. **Metamodelos de Metodologias** . Porto Alegre: CPGCC da UFRGS, 1994. Trabalho Individual.
- [PRE 95] PRESSMAN, R.S. **Engenharia de software** . São Paulo: Makron Books, 1995. 245 p.
- [RUM 99] RUMBAUGH, J.; BOOCH, G.; JACOBSON, I. **The unified modeling language user guide**. Califórnia: Addison Wesley, 1999. 342 p.

- [SCH 2001] SCHWABE, D.; LYARDET, F. Engineering Web applications for reuse. **IEEE Transactions on Multimedia**, [S.l.], v.8, n.1, p.20-31, Mar. 2001 .
- [W3C 2000] W3C WORLD WIDE WEB CONSORTIUM. 2000. Disponível em : < www.w3c.org>. Acesso em: set 2000.
- [WFM 2000] WfMC - WORKFLOW MANAGEMENT COALITION. **Interface 1: Process definition interchange process model**. (Documento Number WfMC TC-1016-P). 1998. Disponível em: < http://www.wfmc.org/standards/docs/TC-1016P_v11_IF1_Process_definition_Interchange.pdf >. Acesso em: fev. 2000.
- [WFM 98] WfMC-O - WORKFLOW MANAGEMENT COALITION. **Interface 1: Process definition interchange organisational model**. (Documento Number WfMC TC-1016-O). 1998. Disponível em: < <http://www.wfmc.org/standards/docs/if19807o.pdf> >. Acesso em: fev. 2000.
- [WFM 2000] WfMC-M - WORKFLOW MANAGEMENT COALITION. **Interface 1: Process definition interchange organisational model**. (Documento Number WfMC TC-1016-M). 1998. Disponível em: < <http://www.wfmc.org/standards/docs/if19807m.pdf>> Acesso em: fev. 2000.
- [WFM 95] WfMC-95 - WORKFLOW MANAGEMENT COALITION. **The workflow reference model**. (Documento Number WfMC TC00-1003). 1995. Disponível em: < <http://www.wfmc.org/standards/docs/tc003v11.pdf>. Acesso em: fev. 2000.